



# Host Access for the Cloud

2.7.1

# Table of contents

---

1. Welcome to Host Access for the Cloud	4
2. Getting Started	5
2.1 How it works	5
2.2 Getting Host Access for the Cloud	7
2.3 Walk through	7
3. Deploying	13
3.1 Deploying Host Access for the Cloud	13
3.2 Minimum System Requirements	14
3.3 About MSS	15
3.4 Planning for Deployment	16
3.5 High Availability Deployment Blueprint	20
3.6 Installing and Upgrading	25
3.7 Ports	29
3.8 Configuring Deployment	30
3.9 Securing Connections	43
3.10 Using Docker	58
4. Managing	66
4.1 Managing	66
4.2 Creating Host Sessions	67
4.3 Connection Settings	68
4.4 Providing Access to Host Sessions	84
4.5 Managing User Preferences	85
4.6 Customizing Host Sessions	86
4.7 Logging	89
5. Using HACloud	91
5.1 Using Host Access for the Cloud	91
5.2 Display Settings	92
5.3 Map Keys	100
5.4 Transfer Files	121
5.5 Specify Edit Options	135
5.6 Working with Sessions	137
5.7 Creating Macros	139
5.8 Macro API Objects	144
5.9 Sample Macros	175
5.10 Run Macro on Event	186
5.11 Printing	187

6. Developing	191
6.1 Developing	191
6.2 Using the Java SDK	192
6.3 Using the Connector for Windows	193
6.4 Using the JavaScript API	194
6.5 Extending the Web Client	195
7. Technical References	197
7.1 Technical References	197
7.2 Monitoring Session Servers using Prometheus and Grafana	198
7.3 Running the Session Server Service as a Dedicated User with Reduced Privileges	200
7.4 Setting the SameSite Attribute	201
7.5 Modifying the Size Limit on Upload File Transfer Operations	202
7.6 Configuring the MSS callback address	203
7.7 Copying Sessions between Management and Security Servers	204
7.8 Changing Ports	205
7.9 Starting and Stopping Services Automatically	206
7.10 Adjusting the URL Path for the Session Server	207
7.11 Configuring User Names when Using Anonymous Access Control	208
7.12 Accessing Host Access for the Cloud using the IIS Reverse Proxy	210
7.13 Using the IIS Reverse Proxy with Host Access for the Cloud	213
7.14 Improving Connection Times on Non-Windows Platforms	214
7.15 Configuring Cross-Origin Resource Sharing (CORS)	215
7.16 Adjusting HTTP Session Timeout	216
7.17 Enabling FIPS Level Security	217
7.18 Using Single Session Mode	218
7.19 Known Issues	219
8. Legal Notice	225

# 1. Welcome to Host Access for the Cloud

---

The Host Access for the Cloud web client provides browser-based HTML5 access to 3270, 5250, VT, UTS, ALC, and T27 host applications. The Host Access for the Cloud product eliminates the need to touch the desktop; no software to deploy, patches to apply, or configurations to make. You can provide platform-independent user access to all your host applications.

The web client operates with complete session protection using TLS to secure communication with your mainframe systems.



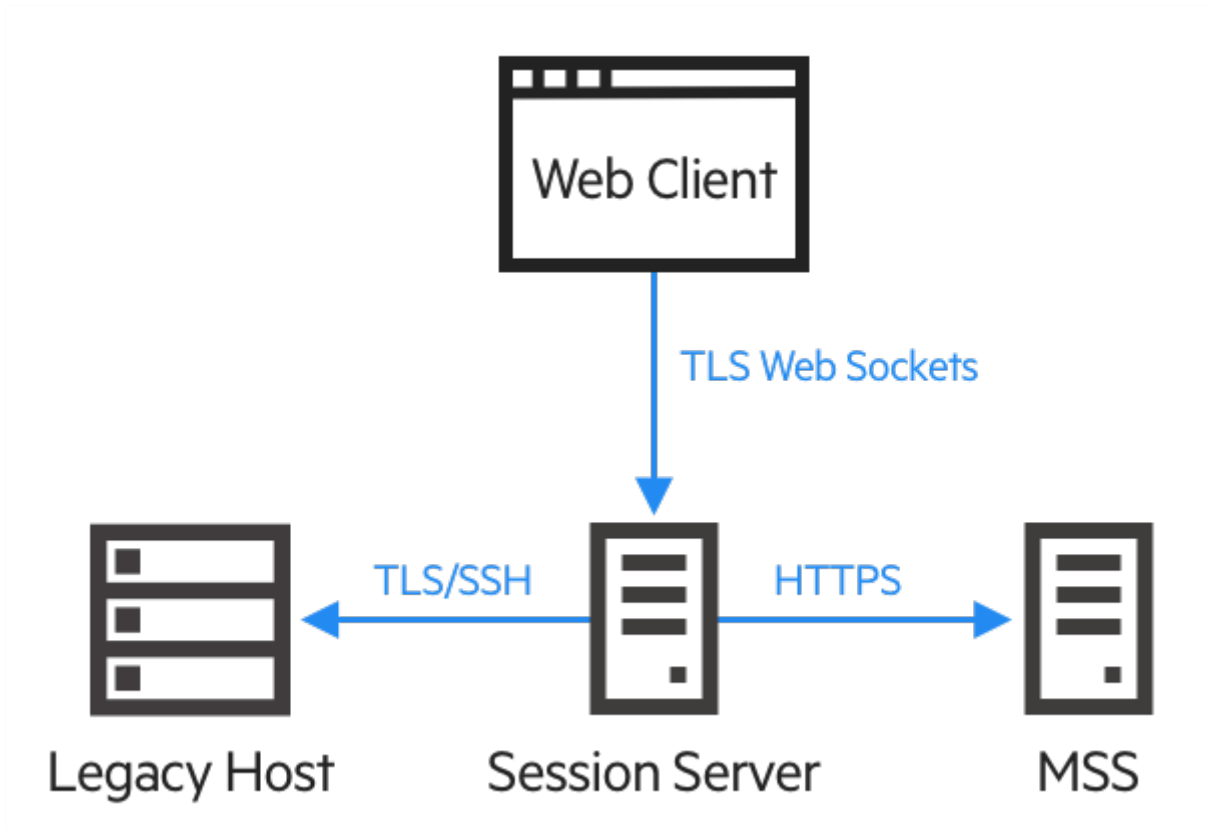
## 2. Getting Started

---

Host Access for the Cloud provides zero-footprint terminal emulation that delivers browser-based HTML5 access to 3270, 5250, VT, UTS, ALC and T27 host applications without the need to touch the desktop or install and manage Java runtime environments. A centralized administrative location reduces IT costs and desktop management time while efficiently providing and delivering host access to end users. Communication is protected using HTTPS, TLS, and SSH security.

### 2.1 How it works

---



## 2.1.1 Components

---

Familiarize yourself with the three components:

- Host Access Management and Security Server

The Host Access Management and Security Server (MSS) provides an Administrative Console, a web-based centralized location where you can add, edit, and delete terminal sessions. MSS is part of the broader Micro Focus story and is compatible with other Micro Focus products.

**MSS**, which you will see throughout the documentation, indicates where additional configuration in the MSS Administrative Console is required.

- Session Server

The session server is the engine that allows end users to securely access your host from the Web Client and other supported connectors. It serves up the Web Client and related connectors on the front end, then manages connections to the host on the back end, ensuring proper authentication and authorization via MSS and secure communication via TLS. Multiple session servers can serve up tens of thousands of host sessions and provide efficient and rapid access to your host data.

- Web Client

The web client is the web-based terminal emulator where your users can easily access authorized sessions from any platform and from any location.

The Web client provides macros, keyboard and color mapping, on-screen keyboard, copy/paste functionality, host-initiated screen updates, and file transfer capabilities.

## 2.1.2 Administrator and end user roles

---

Both administrator and end user roles are represented in the documentation and work flow. The administrator creates sessions, assigns users to those sessions, and sets user preferences. The end user accesses their assigned sessions, interacts with the web client to connect to the host, and accomplishes their tasks.

## 2.1.3 Browser and operating system support

---

Host Access for the Cloud is a 64-bit based product and supports Google Chrome, Mozilla Firefox, and Microsoft Internet Explorer and Edge browsers. The use of Docker containers makes vertical and horizontal scaling possible and supports cloud-based technologies. A complete list of supported platforms and other installation requirements is available in Evaluation system requirements.

## 2.1.4 Security Considerations

---

When you open up your legacy hosts to users outside the corporate firewall - business partners, remote users, mobile sales personnel, and others - you need to shield your information from known security threats. With Host Access for the Cloud, you can provide secure web-to-host access to all your users, whether they're around the corner or around the world. Host Access for the Cloud, along with the MSS, provides HTTPS connections and a variety of authorization and authentication options.

Host Access for the Cloud supports the TLS and SSH protocols to protect mission-critical data. To secure your passwords and other sensitive data, use the HTTPS protocol, which provides TLS encryption.

Host Access for the Cloud can be connected securely to the browser, the host, and the management server. See Secure connections for information on securing those connections.

## 2.2 Getting Host Access for the Cloud

---

### 2.2.1 Evaluation system requirements

---

To successfully install and evaluate Host Access for the Cloud your system needs:

- 8 GB of memory
- A supported browser and operating system.

See [Minimum System Requirements](#) for a comprehensive list of supported environments.

#### Downloading the evaluation software

If you don't have our software yet, visit our site and fill out an evaluation request form. You'll be sent an e-mail message with instructions to download and install an evaluation copy of Host Access for the Cloud good for 120 days. Using this evaluation copy you can open and close host sessions and maintain 25 active host connections at a time. The trial site has all the information you need to take the next step.

The Micro Focus download site contains the compressed files necessary to install all supported platforms, including the Windows connector. Different activation files will enable different editions/platforms of Host Access for the Cloud.

### 2.2.2 Basic install

---

The following instructions provide you with the basic default installation. This means that all components are installed locally and are using default ports. With this installation in place you can follow the walk through and familiarize yourself with Host Access for the Cloud and MSS.

1. From the Micro Focus download site, download your product install package. The package includes support for all supported platforms.
2. Following the install program prompts, install Host Access for the Cloud and Management and Security Server (MSS).

MSS uses activation files (activation.jaw) to enable product functionality. The install program contains the needed activation file and it is activated as part of the install process.



#### Note

During a basic install a self-signed certificate is used to ensure secure connections. When you move into a production environment you can provide your own certificates.

Now you can take the next step; walking through Host Access for the Cloud.

## 2.3 Walk through

---

The following instructions are based on a basic default installation. This means that all components are installed locally and are using default ports. With this installation in place you can follow the steps and familiarize yourself with Host Access for the Cloud and MSS.

See [Deploying](#) for information about installing into production environments and different production scenarios.

### 2.3.1 Steps you will follow

---

1. [Open the MSS Administrative Console](#).
2. [Create a new session](#). This opens a new browser window and the web client Connection panel displays.

3. [Configure settings and connect](#), including connection and user preference options.
4. [Assign users to sessions and configure authentication](#).
5. [Provide end users access to sessions](#).

#### Open the MSS Administrative Console

1. In a Windows environment, from the Start menu, under Micro Focus Host Access for the Cloud, click Administrative Console or open the URL for the administrator login page in your web browser. The URL uses this format: `https://myserver.mycompany.com:443/adminconsole`.
2. If you connect using HTTPS and your server has a self-signed certificate, your browser will warn you about the certificate you created. This is expected behavior; you can accept the self-signed certificate or choose to proceed and the administrator login page will open. After you purchase a CA-signed certificate or import the self-signed certificate into your certificate store, these warnings will stop.
3. The administrative account has a built-in password, `admin`. Log on as an administrator using this password or by entering the password that you specified when you installed MSS.

#### Create a new session



See [Add a Session](#) in the MSS Administrator Guide for complete instructions.



You can add and update session settings from the Manage Sessions panel of the Administrative Console. When you add a session it becomes available in the session list of this panel.

1. From the Manage Sessions panel, click ADD to create a new session.

## Manage Sessions - Add New Session

Configure Session

Product

Host Access for the Cloud ▼

Session name \*

test

Session Server Address \*

https://release-ldap.zfe-ci.attachmate.com

CANCEL

LAUNCH

2. If it is not already selected, select Host Access for the Cloud, enter a session name, and click Launch to open a new browser window and start configuring the session for the server listed at the session server address.
3. On the Create New Session dialog box, select the host type from the drop down list and click Next.

**Create New Session**
×

Name	test2
Type	<div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 5px;">Select session type ▼</div> <div style="border: 1px solid #ccc; padding: 2px; background-color: #007bff; color: white; margin-bottom: 5px;">Select session type</div> <div style="border: 1px solid #ccc; padding: 2px;"> ALC  IBM 3270  IBM 5250  T27  UTS  VT </div>

## Configure settings and connect

You configure different settings and options for the session, as well as connect to the host, in the web client browser window.

1. On the Connection panel, for the session you are creating, enter the needed connection information.


The screenshot shows the 'CONNECTION' panel with the following settings:

Setting	Value
Host	Hostname
Port	23
Connect at startup	Yes
Reconnect when host terminates connection	No
Protocol	TN3270E
Terminal model	Model 2 - 24x80 Extended
Terminal ID	
TLS/SSL security	None
Send keep alive packets	None
Device name	Specify device name
	Device name

At the bottom right, there are 'Cancel' and 'Save' buttons.

2. Connection settings vary depending on the type of host connection. For detailed descriptions of the setting options for each host type, see the web client help. Setting options include mapping keystrokes to selected keys, mapping host colors to match your preferences, and recording session macros.

- Mapping keys

- a. To map keys to selected keys, open Key Mappings.
- b. Press the key or key combination you want to use to trigger the selected action.
- c. From the Action drop down list, select the action you want mapped to the selected keystroke. Click  to complete the key mapping. You can continue adding and mapping keys.
- d. Click Save to complete mapping keys.

- Changing host colors and other options

From the left navigation panel, you can map host colors, set font and keyboard options, and enable hotspots by opening the Display panel. Color choices are specific to each session.

- Setting User Preferences

Open User Preference Rules to extend configuration options to your end users.

3. Click Exit to return to the Administrative Console browser window to authenticate and assign users to sessions.

### Assign users to sessions and configure authentication

Now that sessions are created, you need to grant users access to those sessions. Users are authenticated and assigned to sessions in the MSS Administrative Console. A user can be assigned to multiple sessions.

1. Authentication and authorization validates the identity of a user and the method you want to use to map sessions to individual users or groups of users. From the left navigation panel, select Configure Authentication.
2. Choose an authentication method. Your options change depending on your selection.

## Configure Settings - Authentication & Authorization

Choose Authentication Method

Authentication method

None

LDAP

Single sign-on through IIS

Single sign-on through Windows authentication


X.509

SiteMinder (see help to enable)

Micro Focus Advanced Authentication (not activated, see help to enable)

SAML

REVERT APPLY

3. There are descriptions of the various options in the MSS documentation. Click  .
4. Click Apply to complete the process.
5. Open Assign Access to map sessions to individual users or groups of users.

## Assign Access - Search & Assign

Domain: bhamds.attachmate.com

Search by: Users

SEARCH CLEAR

SELECT ATTRIBUTES

Search Results

\*All users in the selected domain\*

Sessions

Filter

dallas EDIT

Dallas (live) privileged user

dallas with macros

is embedded with authentication... test

Allow access to Administrative Console

Allow user to inherit (\* ) access to sessions

6. Map the sessions to the users you want to access the sessions and click Apply. You can also choose to allow users to inherit access to sessions and to the Administrative Console.

 See [Select a method to authenticate users](#) in the MSS Administrative Guide.

**Provide end users access to sessions**

The final step is to share a URL to your session server with your users. The URL usually looks something like this: `https://myserver.mycompany.com:port`

When they access the session server, your users will be prompted to log in as needed, and then they will be given access to their assigned sessions.

In more complex deployments the URL you provide will be to a load balancer and not the session server itself. These links are often embedded in corporate portals or other proprietary web sites.

## 3. Deploying

---

### 3.1 Deploying Host Access for the Cloud

---

This section goes beyond the basic evaluation setup and assumes you are moving into production. See [Getting Host Access for the Cloud](#) for information on a simple installation.

- [Minimum System Requirements](#)
- [About MSS](#)
- [Planning for Deployment](#)
- [High Availability Blueprint](#)
- [Installing and Upgrading](#)
- [Ports](#)

## 3.2 Minimum System Requirements

---

These platforms *and higher* are supported by Host Access for the Cloud. Requirements do not take into account other applications and resources that may be installed on your system.

### 3.2.1 Supported Web browsers

---

- Google Chrome v75 (recommended)
- Mozilla Firefox v68 (recommended)
- Microsoft Edge 44
- Microsoft Internet Explorer 11 (not recommended)  
See [Browser issues](#) for information on performance issues when using Internet Explorer.
- Apple iOS Safari 13

### 3.2.2 Session Server

---

- **Hardware**
  - CPU - 2 cores (4 cores is recommended)
  - Free memory - 4 GB (6 GB recommended)
- **Operating system (64-bit)**
  - Windows Server 2012
  - SUSE Linux Enterprise Server (SLES) v11 SP4
  - Red Hat Enterprise Linux 7.6
  - Linux on z Systems
    - SUSE Linux Enterprise Server (SLES) v11 SP4
    - Red Hat Enterprise Linux 7.6

### 3.2.3 Additional requirements

---

- See [MSS Installation Guide](#) for system requirements for MSS.
- Load balancers for MSS and Host Access for the Cloud must support sticky sessions and Web Sockets.

## 3.3 About MSS

---

Host Access Management & Security Server (MSS) centrally secures, manages, and monitors your users' access to host connections. Creating sessions, setting up metering, and configuring terminal IDs are all accomplished by working with MSS.

Documentation for MSS:

- [12.8.1 Release Notes](#)
- [Installation Guide](#)
- [Administrative Guide](#)
- [Automated Sign-On for Mainframe - Administrator Guide](#)

## 3.4 Planning for Deployment

---

How many session servers do you need to deploy? How many MSS servers? Are there other considerations? In this section learn how to optimize your session server and MSS server deployment.

### 3.4.1 Scaling and High Availability

---

Determining how many session servers and MSS servers you require to meet your needs is the first task in planning for deployment. Whatever your needs Host Access for the Cloud can be deployed to provide both capacity and high availability.

Your solution depends on your needs, but read High Availability Deployment Blueprint for an illustration of a scalable and highly available deployment.

The main questions that you need to answer are:

- What is the peak number of host sessions that will be used concurrently?
- How many users will be using the system?
- How available does the system need to be in the event of a failure in various areas of the system?

### 3.4.2 Scaling

---

Scalability is a system's ability to handle various amounts of load. To increase capacity, a system can be scaled up (vertically) by running on a more powerful server or scaled out (horizontally) by adding more servers or nodes.

There are trade offs to consider with each case:

- **Scaling up** offers the simplicity of fewer servers, however it also increases the risk of a significant failure if a server goes down.
- **Scaling out** involves more servers but spreads the risk over many servers so when one goes down, a smaller number of users are affected.

Due to the increased resiliency, we recommend scaling out by adding more servers or nodes when increasing capacity.

### 3.4.3 High Availability

---

High availability (HA) is a system's ability to continue providing services when a failure occurs somewhere in the system. HA is achieved by adding redundancy to key components of the system.

 **Note**

This guide discusses providing high availability of the core Host Access for the Cloud services, however true high availability relies on redundancy at many layers in all areas of the systems, which is beyond the scope of this document.



High availability in Host Access for the Cloud is achieved by:

- Deploying enough session servers and MSS servers to provide the needed capacity, with headroom (free capacity) for failures
- Ensuring proper headroom so when a server does fail and the load fails over to the remaining servers, they are not compromised by the extra load
- Using load balancers to distribute the load and direct users to other servers in the event of a failure
- Replication of data between MSS servers, which is handled by MSS clustering

See the [High Availability Deployment Blueprint](#) section for an illustration of how to accomplish these requirements.

### 3.4.4 Session Server Sizing

---

The number of session servers you need is determined by the number of concurrent host sessions you are running. Host sessions generate more load on the session server than users do, therefore we are focusing on the number of required host sessions instead of the number of users.

Number of concurrent host sessions	Number of session servers required
Up to 3000	2 session servers
More than 3000	$(\text{Number of host sessions needed}) / 2000 + 1$ (Minimum of three)

- A single session server supports 2000 concurrent host sessions.
- A session server has built-in headroom to handle 1000 additional host sessions in the event of a failover scenario.
- A minimum of two session servers are required for high availability.

### 3.4.5 MSS Server Sizing

---

Number of concurrent users	Number of MSS servers required
Up to 30,000	3 MSS servers
More than 30,000	$(\text{Number of users needed}) / 10,000 + 1$ (must be an odd number)

- A single MSS server supports 10,000 concurrent users.
- A MSS server has built-in headroom for an additional 5000 users in the event of a failover scenario.
- A minimum of 3 MSS servers are required for high availability
- An odd number of MSS servers are required for high availability due to the need for a database quorum.

### 3.4.6 Using Load Balancers

---

You will need to provide load balancers for both session servers and MSS. There are common settings you should be aware of:

- **Load Balancing Algorithm** - The algorithm determines which server to send new traffic to. We recommend "Least Connections" or something similar. Verifying this setting is properly distributing load is essential for overall system stability. If the load balancer is not configured properly or working effectively, you run the risk of an individual server becoming overloaded.
- **Session Persistence (Affinity/Sticky Sessions)** - This is the ability to send the same user to the same server through multiple requests. Both the session server and MSS are stateful applications and require sticky sessions to be enabled on their load balancers.
- **Health Check Endpoint** - This is the URL on the target service that is used to determine if the instance is healthy and should remain in service. Each server type provides its own health URL.

The [High Availability Deployment Blueprint](#) section provides recommended values for each these settings.

#### TLS Options

There are three typical options for handling TLS on a load balancer. The option you choose depends on your needs.

Your certificate needs to be installed on the load balancer in the first two options. The third option, TLS passthrough, does not require a certificate on the load balancer. The HA Blueprint uses TLS Bridging to provide end-to-end TLS while still allowing cookie-based persistence. The options are:

- **TLS Termination/Offloading** - This option ends the HTTPS connection at the load balancer and continues to the service by means of HTTP.
- **TLS Bridging (Re-encryption)** - This option ends the HTTPS connection at the load balancer and re-establishes a new HTTPS connection between the load balancer and the service. This provides end-to-end TLS while still allowing the load balancer to inject a cookie for session persistence.
- **TLS Passthrough (Required for X.509)** - The load balancer proxies the TLS connection without decrypting it. The downside to this option is that since no cookie can be injected, persistence must be based on source IP or something similar.

#### TLS with X.509 Single Sign-On

When using X.509 authentication, the TLS Passthrough option is required on Host Access for the Cloud and MSS load balancers since client certificates must be presented to the servers in the backend. Because TLS Passthrough is required, you will need a non-cookie based method for session persistence, such as source IP for both the session server and MSS load balancers. This is necessary because with TLS Passthrough there is no chance for the load balancer to decrypt the connection to set or even view a cookie.

### 3.4.7 Terminal ID Manager

---

The Terminal ID Manager does not currently support HA. You can set up a passive server, but the state of the IDS will not be replicated from the active server. When the active server is unavailable, you will still be able to access the passive server, but the IDs will not maintain their current state.

### 3.4.8 Deployment Options

---

You can deploy session servers in one of two ways:

1. Using the traditional method of installing each session server onto a dedicated server
2. Using Docker to run each session server in a container. Docker provides a number of benefits including more flexibility around how many session servers you can run on a single server. See [Using Docker](#) for more information.

### 3.4.9 Understanding Service Interruptions

---

#### Note

When considering deployment options be aware that HACloud is composed of multiple services: the session server, MSS and its add-ons, Terminal ID Manager and Metering. By default, MSS, Terminal ID Manager, and Metering run together in a single process, but they can be configured to run in separate processes. The HACloud session server always runs in its own process.

When the session server is stopped, all host sessions on that server are closed. Users that were logged into that server must re-authenticate when the server restarts or when redirected to a new server.

This table describes the effects on end user's abilities when services are unavailable or have been restarted.

Action	Server down	Server restarted
<b>MSS</b>		
Login	No	Yes
Create new host sessions	No	Yes
Use existing host sessions	Yes	Yes
Reconnect existing host sessions	Yes (excluding SSH sessions)	Yes
Edit sessions	No	Yes (re-login required)
Record/Edit user macros	No	Yes
<b>Terminal ID Manager</b>		
Connect host session that requires an ID	No	Yes
Connect host session that does not require an ID	Yes	Yes
<b>Metering Server</b>		
Create new host sessions	No*	Yes
Use existing host sessions	Yes	Yes

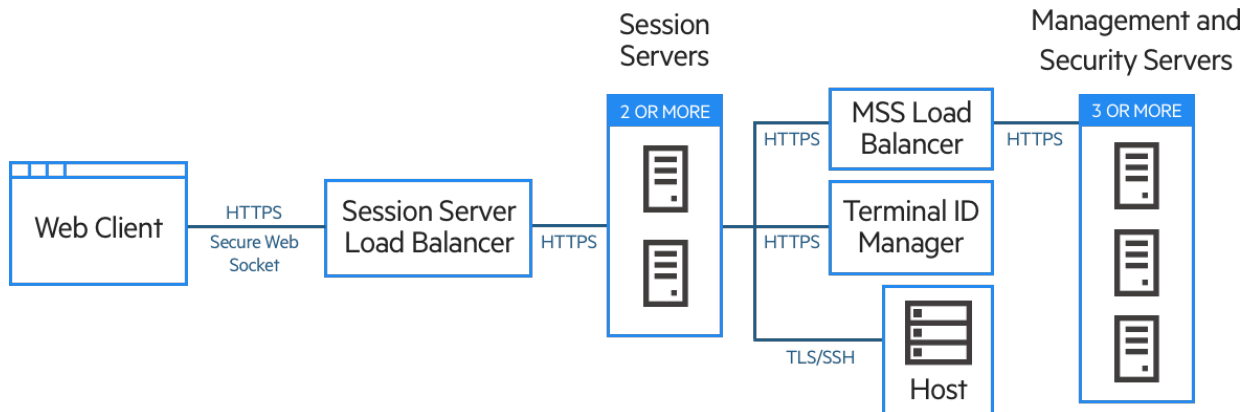
\*If the `metering.host.required` property is set to false, new sessions can be created.

If a session server goes down then all sessions for all users connected through the load balancer to that session server will be lost. Each user will have to login to another session server (through the load balancer) and start new sessions.

## 3.5 High Availability Deployment Blueprint

The following is an example of how to deploy Host Access for the Cloud in a highly available, scalable, and secure manner. While the details of every deployment will differ, for example you may be deploying three or more session servers, the goal here is to provide a known good starting point and to answer common deployment questions.

### 3.5.1 Architecture



This deployment consists of:

- Session server load balancer
- Two or more session servers
- MSS load balancer
- Three or more MSS servers
- Terminal ID Manager
- LDAP or identity management server
- Host/mainframe

### 3.5.2 Deployment benefits

With this example, you will see:

- Capacity for up to three thousand host sessions and the ability to be scaled as needed
- High availability of key services; minimizing single points of failure and distributing load by means of load balancers
- The ability to handle the failure of one session server and one MSS simultaneously without a significant drop in the performance of the web client, due to built-in headroom
- Authentication and authorization options for MSS
- Secure communication through HTTPS

### 3.5.3 Steps when deploying



A number of deployment steps require configuration in the MSS Administrative Console.

We recommend you follow these steps when deploying:

1. Learn about the basics of deployment
2. Provision resources based on system requirements and sizing guidelines
3. Install MSS and create a cluster
4. Configure MSS load balancer
5. Install session servers
6. Configure session server load balancer
7. Configure authentication and authorization
8. Verify deployment

You've learned about the basics of deployment, system requirements and sizing guidelines in previous sections.

### 3.5.4 Installing MSS

---



See the [MSS Installation Guide](#) for complete instructions.

Install three MSS servers and configure each of them for clustering. There is documentation that can step you through the process:

1. Open ports on the firewall. Ports used by MSS and Host Access for the Cloud are listed [here](#).
2. Install MSS and then the Host Access for the Cloud components for MSS by running the Host Access for the Cloud install program on each MSS server.
3. Add each server to a cluster.
4. On each MSS server, configure general settings, security settings, and other settings as required.

#### More information

- [Ports](#)
- [MSS Installation Guide](#)
- [MSS Clustering](#)

### 3.5.5 Configuring an MSS load balancer

---



Use these values when configuring the MSS load balancer:

- **Load balancing algorithm:** Least Connections (or something similar)
- **Session persistence:**

Configure the MSS load balancer to stick on existing cookies and URL parameters in the order specified below:

- Stick on cookie **SESSIONID**
- Stick on URL parameter **sessid** (only necessary if Single sign-on through IIS is configured as the authentication method)
- Stick on cookie **JSESSIONID**
- Stick on URL parameter **jsessionId**
- **Health check endpoint:** `https://<mss-server>/mss/actuator/health`
- **TLS:** Configure TLS and install certificates as needed.

#### More information

- [MSS Using a Load Balancer](#)

### 3.5.6 Installing session servers

---

Install two or more session servers.

For each session server:

1. Open ports on the firewall. [Ports used by MSS and Host Access for the Cloud are listed here.](#)
2. Install the session server. During installation, choose to use a remote MSS server, and enter the MSS load balancer address and port.
3. Import the session server certificate into each MSS's trusted subsystem trust store: `system-trustcerts.bcfks`. This is done automatically for you on the MSS server that the load balancer chose during installation, but must be done manually on the other servers. It is best practice to import or verify its presence on each MSS server.

#### More information

- [Ports](#)
- [Installing and Upgrading](#)
- [Securing Connections](#)

### 3.5.7 Configuring session server load balancer

---

Use these values to configure the load balancer:

- **Load balancing algorithm:** Least Connections (or something similar)
- **Session persistence:** Enabled - use JSESSIONID or a new cookie. Unlike the MSS load balancer you are not required to use the existing JSESSIONID cookie.
- **Health check endpoint:** `https://<session-server:7443>/actuator/health` For the session server in particular, be cautious when you are configuring how to determine a node has failed and what to do once it has failed. If there are users still connected to the instance, those users may lose their host connections. To avoid marking an instance as failed too early, consider increasing timeouts or retries. Some load balancers provide a "drain mode", which allows existing users to stay connected, but will direct new users to other instances.
- **TLS:** Configure TLS and install certificates as needed.

### 3.5.8 Configuring authentication and authorization

---

See [Authentication and Authorization](#) for details on choosing and configuring authentication and authorization

### 3.5.9 Verifying the installation

---

After installing and configuring all components, you'll want to:

- Log into the MSS Administrative Console (through the MSS load balancer).
- Navigate to Manage Sessions > Add a New Session and create a test session.
- Assign the test session to a test user.
- Log into the session server as the test user, through the session server load balancer.
- Verify that the assigned session is available, opens, and can connect.

### 3.5.10 Configuring single sign-on (optional)

---

Following are a few additional considerations when configuring single sign-on in an high availability deployment.

## SAML (Security Assertion Markup Language)



The [MSS Administrator Guide](#) has SAML authentication instructions.

1. Import the MSS load balancer certificate into each MSS `servletcontainer.bcfks` as a trusted certificate.
2. Update the `management.server.url` in each MSS `container.properties` file to use the MSS load balancer address.
3. Set the `management.server.callback.address` property in each MSS `container.properties` file to an address that the session server, for a specific MSS, can reach.
4. Restart the MSS servers.
5. Log on to the Administrative Console of the active MSS server to configure the SAML Authentication. Confirm that the MSS load balancer DNS is used in the Assertion consumer service prefix URL field and add the MSS load balancer and Host Access for the Cloud load balancer DNS to the SAML whitelist
6. Download and edit the service provider metadata to insert each MSS server address as AssertionConsumerService, and import the updated metadata into the SAML identity provider. For example:

```
<md:AssertionConsumerService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST" Location="https://mss-loadbalancer:8443/mss/callback/SAML2Client" index="0"/>
<md:AssertionConsumerService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST" Location="https://mss-server-1:8443/mss/callback/SAML2Client" index="1"/>
<md:AssertionConsumerService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST" Location="https://mss-server-2:8443/mss/callback/SAML2Client" index="2"/>
<md:AssertionConsumerService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST" Location="https://mss-server-3:8443/mss/callback/SAML2Client" index="3"/>
```

7. Adjust the SameSite cookie flag to None. See [Setting the SameSite Attribute](#).

### X.509



The [MSS Administrator Guide](#) has X.509 configuration instructions.

In each case the certificate used must have a Subject Alternative Name (SAN) that contains all MSS server DNS names, along with the load balancer DNS name.

1. Verify that the firewall on the MSS server allows HTTP traffic on the mutual authentication port; 8003 is the default.
2. In each MSS:
  - Replace the certificate of the servlet-engine entry in the `servletcontainer.bcfks` files.
  - Replace the certificate of the system entry in the `system-keystore.bcfks` files.
3. Import the certificate into each session server's `trustcerts.bcfks` file as a trusted certificate.
4. Restart MSS and session servers.
5. Configure both the MSS and HACloud load balancers for TLS passthrough.
6. Configure X.509 authentication as documented here: [How to Configure X.509 Authentication](#).

## 3.5.11 Configuring the Assigned Sessions list

---

You have the option to use the Assigned Sessions list to launch new sessions. To do this additional configuration is needed:

- Configure the MSS load balancer to stick on first the SESSIONID and then the JSESSIONID cookies. It is important to configure the stickiness in that specific order.
- Access to the Assigned Sessions list must be through the same MSS load balancer that the HACloud session server uses to connect to MSS.

**More information:**

- [Providing Access to Host Sessions](#)
- [Using Load Balancers](#)



## 3.6 Installing and Upgrading

---



The MSS Administrator Console help has information on [Product Activation](#).

Keep these things in mind when installing:

- **Activation files**

Activation files (activation.jaw) are used to enable product functionality. For example, the install package contains the needed activation file to enable communication between Host Access for the Cloud and MSS. It is typically activated as part of the install process. Activation files are downloaded from the Micro Focus download site and are specific for the different editions and platforms supported by Host Access for the Cloud. To work in a production environment, activation is required.

If activation was not part of the installation, you need to open the Administrative Console and complete the activation process (Configure Settings > Product Activation). See [Upgrading from previous versions](#) for information on handling activation files when you upgrade.

- **IIS Reverse Proxy with Host Access for the Cloud**

If you plan on using the IIS Reverse Proxy, read [Accessing Host Access for the Cloud using the IIS Reverse Proxy](#) for prerequisites and configuration instructions.

- **Security**

Host Access for the Cloud supports TLS and SSH protocols to protect mission-critical data. To secure your passwords and other sensitive data, you should require browsers to use the HTTPS protocol.

It is advantageous from a security perspective to run services as a dedicated user with a minimal set of privileges. This is known as the principal of least privilege.

- On Linux you should install HACloud and MSS as a dedicated (non-root) user with reduced privileges.
- On Windows you must first install the product, and then adjust the system to run the session server service as a dedicated user with reduced privileges. See [Running the Session Server Service as a Dedicated User with Reduced Privileges](#) for instructions.
- [Instructions for accomplishing this in MSS](#).

### 3.6.1 Installing on different platforms

---

#### Host Access for the Cloud and Java

The session server requires a Java JDK version 8 or higher and MSS requires a Java JRE version 8 or higher. This Java requirement is met during installation, except for systems, such as Linux on System Z that require an IBM JDK. Information on using the nojdk option is available under [Installing on z/Linux](#).

Both Host Access for the Cloud and MSS require that the Java installation support unlimited strength encryption. More information is available on the Java web site.

If necessary, you can use the environment variables specified in the nojdk option and INSTALL4J\_JAVA\_HOME\_OVERRIDE to specify a specific Java installation.

#### Windows

When installing servers on a Windows platform, the installer must be launched by a user who is an Administrator with administrative privileges. A basic Windows installation is described in [Getting Host Access for the Cloud](#).

**UNIX**

- You must either install as “root” or use a user account with root privileges to complete successfully. When the installation has successfully completed, the installed application can be started and managed by “root” or someone running as ‘root’.
- If you are running on Linux platforms, follow these steps to set the session server to start automatically when your system first boots up.
- Elevated privileges are needed to open any application ports lower than 1024. Host Access for the Cloud will not start using a lower port number unless you have system privileges to open low numbered ports.
- You can use the `chmod` command to assign application privileges to users other than root.
- If you are installing on a headless Linux system and there are no fonts installed on the system, you may encounter this font-related error: `java.lang.Error: Probable fatal error: No fonts found`. Ensure that `fontconfig` or at least one font is installed on the system in order to proceed with the installation.

**z/Linux (SUSE E11.x and RHEL 6.x)**

For systems, such as Linux on System Z, that require an IBM JDK, you can use the “nojdk” installer media, which does not include a bundled JDK.

- The installation must be able to locate a Java executable to start. If a Java executable cannot be found by the installer, then you can set the `INSTALL4J_JAVA_HOME` environment variable to refer to a Java installation’s bin directory.
- When started, the installation program will automatically search for version-compatible JDKs on the system. If more than one JDK is found, a list is displayed from which you can choose. If only a JRE is found on the system, you can continue with the installation, but the Host Access for the Cloud server will not run correctly until you have updated the `wrapper.java.command` property located in `sessionserver/conf/container.conf` to refer to a JDK installation.

If necessary, you can use the environment variables named above and `INSTALL4J_JAVA_HOME_OVERRIDE` to specify a specific Java installation.

**Using an unattended installation**

The Host Access for the Cloud installation is based on `install4j` technology, which supports unattended mode. Unattended installation enables you to install the product the same way on a series of computers.

To use unattended installation:

1. Install the session server on a machine using the automated installer. You can use the graphical interface or console mode (`-c`) to install the product.

The installation process creates a text file, `response.varfile`, that contains the selected installation options. The file is located in `[sessionserver Install]\install4j\response.varfile`

2. Copy `response.varfile` to another machine where you would like to install the session server.
3. Locate the appropriate executable to install the product. Launch the installation program using the `-q` argument and a `-varfile` argument that specifies the location of `response.varfile`.

For example, to install the session server on a 64-bit Linux platform with a `response.varfile` located in the same directory, use this command, where `<2.4.x.nnnn>` is the product version and build number:

```
hacloud-<2.4.x.nnnnn>-linuxx64.sh -q -varfile response.varfile
```

You could also add the `-c` option to install in console mode, which would provide feedback such as “Extracting Files” and “Finishing Installation.”

### 3.6.2 Configuring an incomplete installation

---

When the session server is either unable to retrieve a certificate from MSS or is unable to complete the registration process, an incomplete installation can occur. Follow the steps for [adding additional session servers](#) to complete the install.

### 3.6.3 Upgrading from previous versions

---

#### Caution

If you are upgrading, it is important that you remove any activation files from MSS associated with prior versions of Host Access for the Cloud. Leaving obsolete activation files in place may result in limited access to sessions.

1. Before proceeding, backup any changes you made to `hacloud\sessionserver\conf\container.properties` or `hacloud\sessionserver\conf\container.conf`.
2. Install Host Access for the Cloud.
3. Restore the files you backed up in Step One and restart the session server.
4. If not handled during the installation process, install the new activation file or files into MSS using Administrative Console > Configure Settings > Product Activation.

#### Additional configuration

To continue using server side events created in Reflection ZFE versions 2.3.2 or earlier, copy your server-side event JAR files located in `/webapps/zfe/WEB-INF/lib` to `/microservices/sessionserver/extensions/server`, and re-enable extensions.

### 3.6.4 Troubleshooting the installation

---

To complete a successful installation, make sure that you have taken care of these common issues:



#### Are the activation files installed and activated in the Administrative Console?

MSS uses activation files to enable product functionality. With your installation you received an activation file associated with the type of host you are connecting to. For example, if you are licensed for the Unisys Edition, if not handled as part of the install process, you will need to open the Administrative Console, go to Configure Settings > Product Activation and verify that the Host Access for the Cloud Unisys activation file is in place.



#### Is MSS configured for HTTPS?

Connect to the system where the Administrative Server is installed and log in to the Administrative Server. In the Administrative Console, open the Security Setup section and note the protocol selection.



#### Verify that both MSS and Host Access for the Cloud are using trusted certificates

MSS imports certificates and private keys to `C:\ProgramData\Micro Focus\MSS\MSSData\certificates`. See [Securing connections](#). If you are not using trusted certificates, have you configured Host Access for the Cloud to run using HTTP?



#### Are your connection properties configured properly?

In the unlikely event that you have to verify connection information, the `container.properties` file for both the management component and the session server contains the connection properties needed to make the session server to MSS connection as well as the browser to session server connection.

You can find the file in the Host Access for the Cloud installation at

```
<install-dir>/sessionserver/conf/container.properties .
```

 **Install does not complete on UNIX or Linux platforms**

As a result of updated encryption libraries, customers who use headless server-based installations may experience system delays if system entropy is too low. Insufficient entropy may lead to the installation process hanging or degraded server performance. Some platforms already install and enable an entropy service by default, and the issue will not be noticed. If needed, a hardware or software solution can remedy the issue. See the Knowledge Base article, [Ensuring Sufficient Entropy to Avoid System Delays](#).

 **Is the server on which you're installing secured to prevent access to the temp directory?**

See Installation fails due to server preventing access to TEMP directory under [Known Issues](#) for information on this issue.

 **Tip**

For other known issues and troubleshooting information see [Technical References](#).

## 3.7 Ports

---

The following ports are used in Host Access for the Cloud and MSS.

### 3.7.1 Host Access for the Cloud Session Server

---

Port	Component	Remote access required
7443	HTTPS - Web Client and related connectors	Yes
32000-32001	Service wrapper	No

### 3.7.2 Management and Security Server

---

Port	Component	Remote access required
443	HTTPS - Administrative Console, Terminal ID Management, Metering, Metering Management	Yes
7001	Cassandra TLS internode communication	Yes [1]
7199	Cassandra JMX monitoring	Yes [1]
8000	X.509 authentication to MSS via Centralized Management	[4]
8001, 8002	Tomcat AJP used for IIS Integration	Yes [3]
8003	X.509 Trusted subsystem	Yes [1]
8089	Metering server	No [2]
8761	Service Registry	Yes [1]
9042	Cassandra client port	No
9043	Cassandra sidecar	No
32000-32001	Service wrapper monitoring	No
44000	MSS JMX Monitoring	No
Random	Random port required by MSS JMX RMI	No
Random	Random port required by Cassandra JMX RMI	No

[1] Used exclusively for backend communication between MSS And HACloud Session Servers

[2] The metering service is typically accessed via port 443

[3] Only used with IIS Integration

[4] Not used by HACloud Session Servers

Both the Host Access for the Cloud and the MSS Administrative Server ports can be changed depending on your network needs. To modify the session server ports, see [Changing Ports](#).

## 3.8 Configuring Deployment

---

### 3.8.1 Configuring Your Deployment

---

When you begin to configure a deployment of Host Access for the Cloud there are a number of post-installation options to consider, as well as security concerns.

## 3.8.2 Authentication and Authorization

---

### Authentication and Authorization

In HACloud, authentication and authorization are provided by the Host Access Management and Security Server (MSS) and are configured using the Administrative Console.

Authentication validates a user's identity based on some credentials, such as a username/password combination or a client certificate. Authorization is then used to determine which sessions each user can access.

HACloud supports the following authentication methods: None, LDAP, Single Sign-on through IIS, Windows Authentication via Kerberos, Windows Authentication via NTLMv2 (Deprecated), X.509 Client Certificates, SiteMinder, and SAML.

For general information on choosing and configuring authentication and authorization methods, see [Authentication and Authorization](#) in the MSS documentation.

The [High Availability Deployment Blueprint](#) contains additional important details concerning some authentication methods when deploying in an HA environment.

 **Note**

Some authentication methods require HACloud specific configuration. See the topics under this heading in the table of contents for more information.

### Single Sign-on through IIS

**MSS** See [Single Sign-on through IIS](#) in the MSS Administrative Console documentation if you need further information.

This option uses Microsoft IIS web server.

To enable Host Access for the Cloud to work with this authentication method, add the following property in the `<install dir>/sessionserver/conf/container.properties` file :

```
management.server.iis.url= <url>
```

The value of this property is the IIS web server address and port along with the / MSS path. For example: `http://server/mss` . If authentication fails, you may need to remove the domain name in order for the domain credentials to be passed to IIS: `http://server/mss` .



## Windows Authentication - Kerberos

Kerberos is an authentication protocol that uses cryptographic tickets to avoid transmitting plain text passwords. Clients obtain ticket-granting tickets from the Kerberos Key Distribution Center (KDC) and present those tickets as their network credentials to gain access to services.

In Host Access for the Cloud, Kerberos allows end users to access their host sessions on the session server without being prompted for credentials.

### Note

Kerberos authentication to AS/400 hosts is also supported, however that functionality is not yet integrated with Kerberos for authenticating end users accessing the session server.

You enable and configure Kerberos authentication in MSS then enable it on each session server in your deployment. Please review the [Kerberos documentation in MSS](#) to learn about requirements and for more information on configuring and using Kerberos.

Below are the high level steps involved in using Kerberos authentication in Host Access for the Cloud.

#### STEPS TO ENABLE AND CONFIGURE KERBEROS

1. [Enable and configure Kerberos in MSS](#)
2. Configure each HACloud session server for Kerberos
3. Configure browser for Kerberos
4. Launch sessions

#### Configure Kerberos on Session Server

To configure a session server to run Kerberos, edit `service.yml` and add the oauth profile:

1. Open `<install directory>/sessionserver/microservices/sessionserver/service.yml`.
2. Add oauth to the set of active profiles:

```
- name: SPRING_PROFILES_ACTIVE
  value: tls, oauth
```

3. Restart the session server.

Optionally, if configuring a [high availability deployment](#) with load balancers, the oauth profile (previous step) and the following properties must be configured in service.yml on each session server.

1. Configure the URL of the MSS load balancer. The session server will redirect users to this URL for authentication.

```
- name: AUTHSVC_HOST
  value: {HTTPS url of MSS load balancer}
```

2. Configure the domain name of the session server load balancer. MSS will redirect back to this server after a user is authenticated.

```
- name: PROXY_DOMAIN
  value: {FQDN of the session server load balancer}
```

3. Configure the port that is used when accessing the session server via the session server load balancer.

```
- name: PROXY_PORT
  value: {Port number on the session server load balancer}
```

4. Restart the session server.

#### Example

```
- name: SPRING_PROFILES_ACTIVE
  value: tls, oauth
  (If using load balancers...)
- name: AUTHSVC_HOST
  value: https://mss-load-balancer.mydomain.com
- name: PROXY_DOMAIN
  value: sessionserver-load-balancer.mydomain.com
- name: PROXY_PORT
  value: 7443
```

#### Configure your browser for Kerberos

In order to sign in using Kerberos, your browser must be configured correctly for Windows Authentication via Kerberos and your machine must be a member of the proper domain (Kerberos realm). Please consult the help for your specific browser for instructions on how to enable Kerberos.

#### Launch sessions

##### Note

Kerberos authentication is not currently supported by the Assigned Sessions List. It is only available when entering through the session server.

HACloud sessions need no additional configuration to launch and authenticate using Kerberos, as long as you your browser has been configured correctly for Windows Authentication / Kerberos. Just navigate to <https://session-server-lb.mydomain.com:7443> and you'll be automatically logged into the HACloud session server.

## SAML

SAML (Security Assertion Markup Language) is an XML-based open standard format that exchanges authentication and authorization data between an identity provider, (the server that issues SAML assertions and performs authentication), and a service provider, (the web server from which you access information or services). MSS acts as the service provider.

**MSS** See [SAML Configuration Steps](#) in the MSS Administrative Console documentation if you need further information.



### Note

The SameSite flag must be adjusted when using SAML behind a load balancer. See [Setting the SameSite Attribute](#).

## TROUBLESHOOTING THE CONFIGURATION

If you have problems authenticating or see session timeout errors, see [Troubleshooting SAML Setup](#) in the MSS Administrator Guide.

## X.509 Authentication

X.509 client authentication allows clients to authenticate to servers with certificates rather than with a user name and password by leveraging the X.509 public key infrastructure (PKI) standard.

**MSS** MSS has additional information on [X.509 configuration](#).

### ENABLING X.509 CLIENT AUTHENTICATION

- When the user accesses the web client using TLS the browser sends a certificate to the session server identifying the end user and completing the TLS handshake.
- The session server refers to its truststore to check the client's certificate and verify its trust.
- Once the TLS negotiation is complete (the session server trusts the end user), the session server sends the end user's public certificate to MSS for further validation.
- MSS also verifies that it trusts the end users certificate using its trust store.
- When MSS finishes the validation, the end user will have successfully authenticated.

The client's full certificate chain needs to be present in the session server and MSS truststores or alternatively signed by a Certificate Authority that is present in the truststores.

The browser determines what client certificate to send using a browser or smart card specific configuration.

#### Basic steps

1. Trust certificates in the session server and MSS if they have not already been trusted.
2. Restart the servers.
3. Configure X.509 in the MSS Administrative Console.

#### Trust the certificate in MSS and the session server

- Trust the certificate in MSS  
MSS' trusted store may already contain your signing authority certificate. This is often the case with well-known certificate signing authorities, and if so, then you can skip this step.  
To check:
- Open the Administrative Console, click Configure Settings, and open the Trusted Certificates tab. Open Trusted Root Certificate Authorities to see a list of available certificates.
- If your certificate is not listed you need to install your signing root CA into MSS following the prompts and documentation in the Administrative Console.
- Trust the certificate in the session server

To install the certificate into the session server:

In `<install_directory>\sessionserver\etc` import the certificate:

```
keytool -importcert -file <cert-file> -alias <alias-to-store-cert-under>
-keystore trustcerts.bcfks -storetype bcfks -providername BCFIPS
-providerclass org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider
-providerpath ../lib/bc-fips-*.jar
-storepass changeit
```

#### Restart all the servers

For the configuration to take effect, you must restart all servers.

#### Configuring X.509 with LDAP fail over in the MSS Administrative Console

Once the certificates are in place, you can enable X.509 with the Fallback to LDAP authentication option in Management and Security Server Administrative Console | Configure Settings |

Authentication & Authorization. See the Administrative Console online help for descriptions of the configuration options.

#### ENABLING X.509 THROUGH A LOAD BALANCER CONFIGURED FOR TLS TERMINATION

In this configuration, the load balancer provides end user authentication by validating their client certificate. However, the client certificate still needs to be sent to all the MSS systems in order to identify the incoming user.

If the load balancer is configured to terminate the TLS connection, the user's certificate can be added to an HTTP header, extracted by the session server, and then passed to MSS for authorization. In order to pass the certificate in a header, first set the header name in the HACloud Session Server's `container.properties` file:

To pass the certificate in a header

1. Set the header name in the HACloud session server's `container.properties` file:

```
x509.header.client.cert=X-SSL-Client-Cert
```

2. Set the header value to the user's certificate in the load balancer configuration. For example, using a BigIP iRule:

```
HTTP::header insert X-SSL-Client-Cert [URI::encode $client_cert]
```

This assumes that `$client_cert` has been set to the user's certificate in PEM format. If the user's certificate is in DER format, use Base64 encoding:

```
HTTP::header insert X-SSL-Client-Cert [b64encode $client_cert]
```

Encoding the certificate ensures that the header value is one line of ASCII text. This is required for the HACloud session server to read the value.

#### Note

The client certificate authentication must still take place between the load balancer and the session server. The load balancer needs to be configured to send its certificate to the session server and the load balancer's CA must be present in the session server's trust store.

3. After configuring the load balancer to send its certificate to the HACloud session server and configuring the user's certificate to be passed in the header, restart the session server.

Connecting with a certificate or smart card through the load balancer will result in successful authentication and authorization as the user represented by the certificate. To verify proper operation, set the session server's log level to DEBUG and examine the `sessionserver.log` file for entries like these:

```
Attempting to extract certificate from X-SSL-Client-Cert header.
User <DN value> has been preauthenticated from <IP address>
```

#### Additional configuration

By default, the HACloud session server's trust store contains the Java CA certificates. As a result, the HACloud session server will accept any client certificate signed by well-known CAs. In order to ensure that only the desired load balancers connect to the session server, you must remove the Java CA certificates from the trust store and ensure only the necessary certificates are installed in the trust store.

To filter the allowed client certificates by issuer DN, set the following properties in the HACloud session server's `container.properties` file:

```
X509.client.cert.issuer=<DN value>
X509.client.cert.subject=<Subject DN value>
X509.client.cert.serial=<Serial number>
X509.client.cert.sha1=<SHA1 fingerprint>
X509.client.cert.sha256=<SHA256 fingerprint>
```

DN values must match the load balancer's certificate issuer or subject DN exactly. The serial number value should be a decimal (base 10) value. SHA1 and SHA256 fingerprint values should be entered in hexadecimal. When any of these properties are set, the incoming certificate's attributes will be checked to ensure they match the specified property values. Authorization will fail if any values do not match.

### 3.8.3 Set Up Metering

**MSS** MSS provides metering capabilities to monitor host sessions. See [Metering](#).

The Management and Security Server provides metering capabilities to monitor host sessions.

Before configuring metering for Host Access for the Cloud, verify that you have metering enabled for MSS. There are complete instructions in the [Installation Guide](#).

In Host Access for the Cloud, metering is set globally for all emulation sessions created by the session server. Settings are configured in the `sessionserver/conf/container.properties` file.

Property	Description
<code>metering.enabled</code>	Turns metering on or off, with a value of "true" or "false". Any value other than "true" turns metering off.
<code>metering.host.required</code>	Determines whether the session can connect to the host even if the metering server cannot be contacted. "True" means that session connections will fail if the metering host is unavailable. "False" means that session connections will still work even if the metering host is unavailable.
<code>metering.server.url</code>	Specifies the name or address of the metering server, the port, the protocol, and the webapp context. The syntax is <code>host:port protocol context</code> . This syntax is the same as that used by the MSS server in the <code>MssData/serverconfig.props</code> file to register metering servers. The <code>host:port</code> section of the URL must escape the ":" character. For example, <code>10.10.11.55\443</code> .

Example additions to `sessionserver/conf/container.properties`

```
metering.enabled=true
metering.host.required=false
metering.server.url=10.10.11.55\443|https|meter
```



#### Note

In the event that all licenses are in use and you attempt to make a connection, the session will be disconnected. To determine whether the host has disconnected or the metering service has stopped the connection, see the `<install_dir>/sessionserver/logs/sessionserver.log` file.

## 3.8.4 Set Up the Terminal ID Manager

---

**MSS** Setting up the Terminal ID Manager requires that this feature is enabled in MSS.

The Management and Security Server provides a Terminal ID Manager to pool terminal IDs, track ID usage, and manage inactivity timeout values for specific users, thus conserving terminal ID resources and significantly reducing operating expenses.

The Terminal ID Manager Add-On requires a separate license.

Before you configure the Terminal ID Manager for Host Access for the Cloud, verify that you have this option enabled for MSS. There are complete instructions in the MSS Installation Guide.

### Tip

If MSS and Host Access for the Cloud are installed on the same machine, no additional configuration is needed.

### Configuring Terminal ID Manager for Host Access for the Cloud

To configure the Terminal ID Manager for Host Access for the Cloud, you must provide the correct address to the Terminal ID Manager.

1. Open the `sessionserver/conf/container.properties` file.
2. Update `id.manager.server.url=https://localhost:443/tidm` to reflect the address of the Terminal ID Manager configured in Management and Security Server.
3. Restart the session server.



### 3.8.5 Set Up Automated Single Sign-On for Mainframe

---

**MSS** [Automated Sign-On for Mainframe - Administrator Guide](#) has additional information on configuring this option.

Automated Sign-On for Mainframe is an add-on product to Management and Security Server that enables an end user to authenticate to a terminal emulation client and be automatically logged on to a host application on the z/OS mainframe.

1. Install and configure the Automated Sign-On for Mainframe add-on for Management and Security Server. You can find complete instructions in the [Automated Sign-On for Mainframe - Administrator Guide](#).
2. After the Management and Security Server setup is complete, open the Administrative Console to add sessions and map users to those sessions. During that process, you can complete the additional configuration needed to implement automated sign-on.
3. A Host Access for the Cloud macro sends the user's mainframe username and pass ticket to the host application. The user is then automatically logged in. To help create the macro:
  - The Macro API contains the [AutoSignon](#) object that provides the methods needed to create a Host Access for the Cloud login to use with the Automated Sign-On for Mainframe feature.
  - You can also reference the [sample macro Automatic Sign-On Macro for Mainframes](#) that uses the AutoSignon object to create a macro that uses the credentials associated with a user to obtain a pass ticket from the Digital Certificate Access Server (DCAS).

### 3.8.6 Set Up Kerberos for AS/400 Single Sign-on

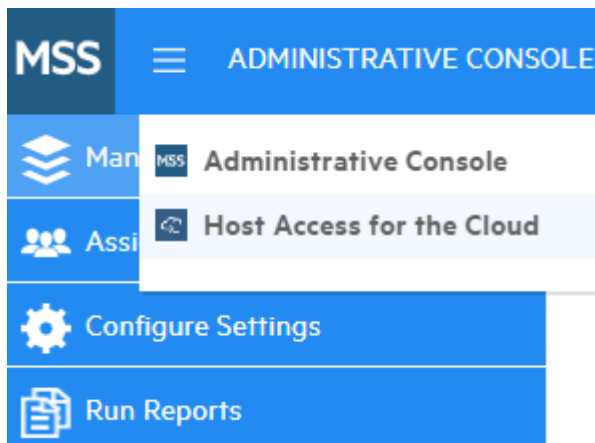
Kerberos is an authentication protocol that uses cryptographic tickets to avoid transmitting plain text passwords. Client services obtain ticket-granting tickets from the Kerberos Key Distribution Center (KDC) and present those tickets as their network credentials to gain access to services.

#### Note

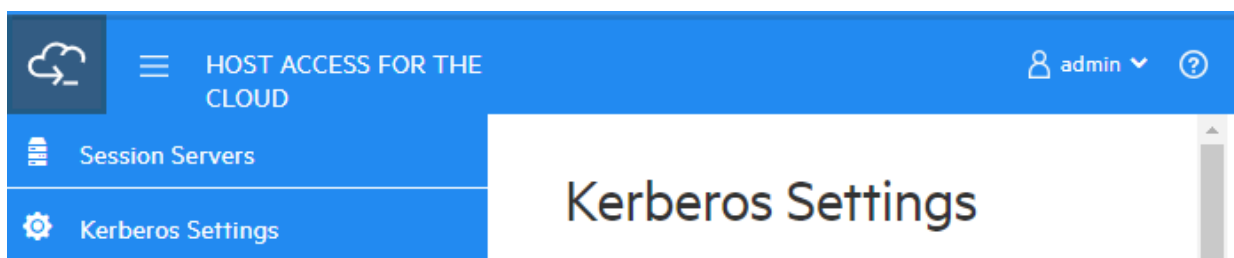
Kerberos authentication for end users accessing the session server is also supported, however that functionality is not yet integrated with Kerberos for AS/400 authentication. This feature allows for automated sign on from the session server to an AS/400 host. MSS must be configured with an authentication method that results in a user principle that is resolvable in the Kerberos Active Directory domain, for example LDAP, SAML, or Siteminder. A Windows Active Directory Server is required.

By using Kerberos, after an initial domain sign-on, users do not have to enter their credentials when accessing AS/400 sessions in Host Access for the Cloud.

An overview of enabling and using this feature can be found in the MSS Administrative Console > Host Access for the Cloud panel documentation.



Choose Host Access for the Cloud from the drop down list, then select Kerberos Settings, and click the Help button:



## 3.9 Securing Connections

---

### 3.9.1 Secure Connections

---

Host Access for the Cloud uses Transport Layer Security (TLS) to cryptographically secure communication between client web browsers, session server, MSS and backend hosts.

#### Overview

##### Public Key Infrastructure (PKI)

TLS uses Public Key Infrastructure (PKI) to implement security. PKI uses keys, both public and private, to secure client and server communication. Public and private keys are mathematically related, but they are not the same. This means that a message encrypted with a public key can only be decrypted using the private key. Together, these keys are known as a key pair.

##### Certificates

Digital certificates are credentials that verify the identities of individuals, computers, and networks. They provide the link between a public key and a business that has been verified (signed) by a trusted third party, known as a certificate authority (CA). Digital certificates provide a convenient way to distribute trusted public encryption keys.

##### Keystores

Certificates and private keys are stored in Java keystores. Keystore entries are identified using an unique identifier, known as an alias. Often private keys and certificates, with their corresponding public key, are stored separately from those certificates received from other parties that you are using for trust purposes. This separate keystore is referred to as a truststore. A truststore contains certificates from parties that you expect to communicate with or from Certificate Authorities that you trust to identify other parties.

#### Your default secure installation

During the installation of HACloud and MSS, self-signed certificates are generated, exchanged, and then used to secure all communication between the session server, web browsers and MSS. Self-signed certificates are identity certificates that are signed by the same entity whose identity they certify.

Both session servers and MSS servers use their generated self-signed certificates to identify themselves to remote clients such as web browsers and other session servers and MSS servers. These self-signed certificates and their private keys are stored in their respective keystores.

To complete secure communication between clients (web browsers, session servers and MSS servers), the clients must trust the generated self-signed certificate. The session server trusts MSS' certificate during installation and stores it in its truststore. Likewise, during installation MSS retrieves and trust the session server's certificate and stores it in its truststore.

See [Stores used by the session server](#)



The MSS Administrative Console help has detailed information on [General Security and Certificates](#)

## 3.9.2 Stores Used by the Session Server

---

### Identity certificates

As a convenience, the identity certificates for each type of server are available at the following locations, outside of their respective keystores mentioned below.

- HACloud session server certificate - `HACloud/sessionserver/etc/<computer-name>.cer`
- MSS certificate - `MSS/server/etc/<computer-name>.cer`

### Session server keystore and truststore

The keystore and truststore used by the session server are described in the table below.

- Location: `HACloud/sessionserver/etc/`
- Type: `bcfks` (Bouncy Castle FIPS keystore)
- Default password: `changeit`

Keystore	Function
<code>keystore.bcfks</code>	<ul style="list-style-type: none"> <li>• Credential store for incoming TLS connections</li> <li>• Contains the certificate served up by the session server</li> <li>• Used for embedded web server (Jetty)</li> <li>• Created at start up</li> </ul>
<code>trustcerts.bcfks</code>	<ul style="list-style-type: none"> <li>• Trust store for outgoing TLS connections</li> <li>• Used to verify the servers the session server connects to, such as MSS</li> <li>• Trust store for verifying incoming load balancer connections when using X.509 authentication through a load balancer</li> <li>• Created at start up</li> </ul>

#### Note

Trust for host emulation connections is managed by MSS. See [Make a secure emulation connection to a trusted host](#)

### To change a keystore or truststore password

In `HACloud/sessionserver/conf/container.properties`, update these settings:

- `server.ssl.key-store-password`
- `server.ssl.trust-store-password`

For security reasons it is best to use an obfuscated password. To generate one, run the following command from the `HACloud/sessionserver` directory:

```
../java/jre/bin/java -cp ../lib/jetty-util-<version>.jar org.eclipse.jetty.util.security.Password passwordToObfuscate
```

### 3.9.3 Tools

---

- KeyStore Explorer - You can take advantage of the KeyStore Explorer utility to provide a simple user interface to create signing requests (CSR) and import CA-signed certificates into Host Access for the Cloud.
- To launch KeyStore Explorer on Windows - run `\HACloud\utilities\keystore-explorer.bat` as an administrator or with administrative rights.
- To launch KeyStore Explorer on UNIX - run `hacloud\utilities\keystore-explorer.sh` as an administrator or with administrative rights.

The utility has an online Help system available to walk you through the user interface.

- Java Keytool - The Java Key and Certificate Management Tool manages a keystore of cryptographic keys, X.509 certificate chains, and trusted certificates. It uses a command line interface. The Java Key and Certificate Management Tool documentation is available for both Unix and Windows platforms:
  - [Unix](#)
  - [Windows](#)
- Entropy and encryption - Entropy is the randomness collected by an operating system for use in cryptography. This randomness is often collected from hardware sources, like mouse movements. A lack of entropy can have a negative impact on performance and security and will be more apparent in headless server-based installations.

Solutions to improve entropy generation:

- Keytool parameter - to change how entropy is generated, add an additional parameter to the linux keytool command line. `-J-Djava.security.egd=file:/dev/urandom`
- Haveged tool - this is a tool to help remedy low-entropy conditions in the Linux random device, which can occur under some workloads, and especially on headless servers. See <https://wiki.archlinux.org/index.php/Haveged> for information on this tool.

### 3.9.4 How do I...?

---

#### Apply for a Digital Identity Certificate (Certificate Signing Request)

##### Terms used:

- private key - a secret key known only to the owner, used with an algorithm to encrypt/decrypt data
- key pair - private key and its associated certificate chain
- distinguished name - the identifying information in a certificate. A certificate contains DN information for both the owner / requester of the certificate (called the Subject distinguished name) and the CA that issued the certificate (called the Issuer distinguished name)
- X.509 certificate - a digital certificate that uses the widely accepted international X.509 public key infrastructure (PKI) standard to verify that a public key belongs to the user

Before creating a Certificate Signing Request (CSR), the applicant first generates a key pair, keeping the private key secret. The CSR contains information that identifies the applicant (such as a distinguished name in the case of an X.509 certificate) which must be signed using the applicant's private key. The CSR also contains the applicant's chosen public key.

##### HOW TO CREATE A CSR USING KEYSTORE EXPLORER

To create a CSR you will create a key pair and then generate a certificate request. If you do not need to update certificate information, you can skip creating the key pair and proceed to generating the certificate request.

- Create a new key pair
  - a. From the Tools menu, select Generate Key Pair.
  - b. On the Generate Key Pair dialog box, enter the algorithm information and certificate details. Click OK.
  - c. Specify the relevant alias (*servlet-engine*) and default password (*changeit*).
- Generate a certificate request
  - a. Select the key pair you just created.
  - b. From the right-click menu, select Generate CSR.
  - c. Browse to the file location where you want to generate the CSR and enter the file name. Click OK.

##### HOW TO CREATE A CSR USING JAVA KEYTOOL

- Create Key Pair (replace the `dname` parameter with your own) in the `sessionserver/etc` folder:

```
..\..\java\bin\keytool.exe -genkeypair -dname "CN=hacloud-1.microfocus.com, O=Micro Focus, C=US" -alias servlet-engine -keyalg RSA -
keysize 2048 -keystore keystore.bcfks -validity 1095 -storetype bcfks -storepass changeit -keypass changeit -providername BCFIPS -
providerpath ../lib/bc-fips-*.jar -providerclass org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider
```

- Generate Certificate Request:

```
..\..\java\bin\keytool -certreq -alias servlet-engine -keystore keystore.bcfks -file cert_request.csr -ext
ExtendedKeyUsage=serverAuth -storetype bcfks -storepass changeit -providername BCFIPS -providerpath ../lib/bc-fips-*.jar -
providerclass org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider
```

After you receive the certificate from the CA, you will import the certificate into Host Access for the Cloud.

## Replace the Session Server Certificate

Your installation is secured using self-signed certificates. Self-signed certificates, while just as secure as commercial certificates, aren't automatically trusted. This makes them hard to manage. Commercial certificates are necessary when you need widespread support for your certificate and fortunately, most web browsers and operating systems already support many commercial certificate authorities.

### Information you must know:

- keystore location - `/etc/keystore.bcfks`
- keystore format - `bcfks` (Bouncy Castle FIPS)
- default password - `changeit`
- key pair alias - `servlet-engine`

How to replace the self-signed certificate varies depending on whether you are replacing the self-signed certificate with one obtained through a CSR into the default keystore or whether you are replacing it with your own non-default keystore and certificate.

### More information

- [Replace the self-signed certificate with certificate reply from Certificate Authority \(CA\)](#)
- [Replace the certificate with your non-default keystore](#)

## Replace Self-signed Certificate with Certificate Reply from Certificate Authority

1. Create a Certificate Signing Request (CSR) for the session server and send it to the certificate authority of your choice (CA) of your choice. When you receive the signed certificate from the CA, then:
2. Import the CA-signed certificate/chain into the session server's keystore.

You can accomplish this task using either KeyStore Explorer or the Java Keytool command line instructions. Whatever tool you decide to use, if the CA Reply contains separate root and intermediate certificate files, import the root certificate into the keystore first, followed by the intermediate certificate.

### Using Keystore Explorer

- a. Open `keystore.bcfks` in KeyStore Explorer. Use the password `changeit`.
- b. If separate root and intermediate certificate files are available, from the tool bar, select Import Trusted Certificate to import certificates.
- c. Select the servlet-engine key pair. Right-click and select Import CA Reply to import the file into the key pair.
- d. If prompted, enter the password, `changeit`.
- e. Browse to the location where the CA Reply file is stored, select the file, and click Import.

### Using Java Keytool

These examples use keytool command at the `sessionserver\etc` directory.

For Windows:

Import Root CA and intermediate certificates

```
..\..\java\bin\keytool.exe -importcert -alias rootca -trustcacerts -file <RootCA.cer> -keystore keystore.bcfks -storetype bcfks -storepass changeit
..\..\java\bin\keytool.exe -importcert -alias intermediateca -trustcacerts -file <IntermediateCA.cer> -keystore keystore.bcfks -storetype bcfks -storepass changeit -providername BCFIPS -providerpath ../lib/bc-fips-*.jar -providerclass org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider
```

Import CA Reply

```
java
..\..\java\bin\keytool.exe -importcert -alias servlet-engine -trustcacerts -file <CertChainFromCA.p7b> -keystore keystore.bcfks -storetype bcfks -storepass changeit -providername BCFIPS -providerpath ../lib/bc-fips-*.jar -providerclass org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider
```

For Unix:

Import Root CA and intermediate certificates

```
../..../java/bin/keytool -importcert -alias rootca -trustcacerts -file <RootCA.cer> -keystore keystore.bcfks -storetype bcfks -storepass changeit -providername BCFIPS -providerpath ../lib/bc-fips-*.jar -providerclass org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider
../..../java/bin/keytool -importcert -alias intermediateca -trustcacerts -file <IntermediateCA.cer> -keystore keystore.bcfks -storetype bcfks -storepass changeit -providername BCFIPS -providerpath ../lib/bc-fips-*.jar -providerclass org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider
```

Import CA Reply

```
../..../java/bin/keytool -importcert -alias servlet-engine -trustcacerts -file <CertChainFromCA.p7b> -keystore keystore.bcfks -storetype bcfks -storepass changeit -providername BCFIPS -providerpath ../lib/bc-fips-*.jar -providerclass org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider
```



3. Trust the new certificate in MSS.

- As the administrator, log into MSS.
- From the left panel, click Configure Settings > Trusted Certificates.
- Select Trusted Sub-System. The list contains the certificates that are trusted by MSS.
- Click IMPORT to add the session server certificate to the list.
- There is no need to repeat the procedure for each MSS. Changes are automatically replicated to other MSS in the cluster.

There is detailed help available in the Administrative Console help -[Trusted Certificates](#).

## Replace the Certificate with Your Non-default Keystore

You can use a keystore other than the default ( `sessionserver/etc/keystore.bcfks` ) to store your CA-signed certificates.

Specify the following properties in `sessionserver/conf/container.properties` :

```
server.ssl.key-store
server.ssl.key-store-password
```

Where the keystore path is set to the non-default keystore file name and the keystore password is set to the obfuscated value generated by the following command from the `sessionserver` directory:

```
../java/jre/bin/java -cp ./lib/jetty-util-<version>.jar org.eclipse.jetty.util.security.Password passwordToObfuscate
```

For example:

```
server.ssl.key-store=${server.home}/etc/custom.bcfks
server.ssl.key-store-password=0BF:1vn2lugu1saj1v9i1v941sarlugw1vo0
```

### Tip

To avoid confusion, delete the default keystore.

To prevent the default keystore from being generated when the server starts up, open `/conf/product-core-ctx.xml` in a text editor and either remove or comment out the `ServletEngineKeystoreGenerator` section. Restart the session server.

## Replace the MSS Certificate

**MSS** Review how to replace the MSS certificate in the [General Security and Certificates](#) topic.

During installation, to establish secure communication, the session server trusted the existing MSS certificate. If the MSS certificate is updated, it must be re-trusted by all the HACloud session servers.

### To replace the MSS certificate

- To trust the new MSS certificate, import it into the session server's truststore using the alias mss. See [Import a certificate into the session server's truststore](#).
- You must import the new MSS certificate into each session server.

### Make a Secure Emulation Connection to a Trusted Host

Follow these steps to configure a TLS connection between the Host Access for the Cloud session server and a host that supports TLS:

1. Configure the trusted keystore in MSS.
2. Configure the terminal session.

#### HOW TO CONFIGURE THE KEYSTORE IN MSS

**MSS** Open the MSS Administrative Console > Configure Settings > Trusted Certificates and choose Terminal Emulator Clients. You can access the documentation for the Administrative Console by clicking the Help icon in the upper right of the page.

For a session to trust the TLS host it connects to, the public certificate of the host must be added to a trusted keystore using the Management and Security Server (MSS). The Host Access for the Cloud session retrieves this certificate the first time a session connects.

When the certificate is successfully added to the MSS server's trusted keystore, you are returned to the list of certificates and you should see the new host.

## HOW TO CONFIGURE A HACLOUD TERMINAL SESSION

Depending on your host type, you can configure a terminal session using different security protocols.

Type	Procedure
Using TLS	<p>To connect to the new trusted host using TLS, configure a terminal session as usual, and in the Settings dialog box, specify TLS as the security protocol. Make sure to specify the correct TLS port for the connection.</p>
Using Secure Shell (SSH) with VT host types	<p>Secure shell provides encrypted communications between the client and a VT host. MSS has a known hosts list that contains the public keys of hosts that you can connect to using SSH. SSH connections can be made only to hosts already trusted by an administrator. The first time an SSH connection is made from a session to a host, the known hosts file is downloaded from MSS to the session server. When you attempt to create or edit a session using SSH in the session management panel, you will be notified if the key is not recognized as trusted and asked if you want to trust the key and continue.</p> <ul style="list-style-type: none"> <li>• If you enter yes, the host will be trusted and added to the known host list, and you will be prompted for the SSH host password.</li> <li>• If you do not answer yes, then the host will remain untrusted and the session will be disconnected.</li> </ul> <p>You can also configure the SSH known hosts file manually by establishing an SSH connection from a session to the host, and adding the remote host's key fingerprint to the known hosts list in MSS.</p>
Configure the known hosts file for SSH connections in MSS	<ol style="list-style-type: none"> <li>1. Connect to the system where MSS is installed and navigate to the server's certificates folder: <code>C:\ProgramData\Micro Focus\Mss\MssData\certificates</code> (Windows) or <code>/var/opt/microfocus/mss/Mssdata/certificates</code> (UNIX).</li> <li>2. Copy the public certificate file of the new SSH host into the <code>MssData/certificates</code> (Windows) or <code>/etc/ssh/ssh_host_rsa_key.pub</code> (UNIX) folder described above. Only <code>ssh-rsa</code> and <code>ssh-dss</code> are valid as public key types for MSS known_hosts entries. The host's public key format can be OpenSSH, Base64-encode,.DER, or.PFX. The file should follow this format: hostname, IP-address key-type key. For example, a public key entry might look like this: <code>alpsuse132, 10.117.16.232 ssh-rsa AAAAB3NzaC1yc2EAAAADAQAB.....</code></li> <li>3. Log in to MSS (for example, <code>http://mycompany.com/adminconsole</code>).</li> <li>4. Open the Administrative Console.</li> <li>5. Click Configure Settings &gt; Secure Shell. After the public key is imported into the known hosts file, you will return to the Secure Shell Known Hosts page and the new host will appear in the list.</li> <li>6. Follow the directions in MSS to import a known host. After the public key is imported into the known hosts file, you will return to the Secure Shell Known Hosts page and the new host will appear in the list.</li> </ol>

### Set up Server Side Events to Make Outbound TLS Calls from the Session Server

When writing Java code that runs in your server side events, you may want to make outbound calls to remote servers using TLS. If the remote server is well known it may already be trusted by the session server and there isn't anything more to setup. However, often times the remote server isn't well known and you will need to trust it by importing its certificate into the session server's truststore.

#### To trust the remote server

Import its public certificate into the session server's truststore following these instructions, [Import a certificate into the session server's truststore](#).

### Add More MSS Servers to My Installation

During installation both the MSS and HACloud servers have exchanged and trusted their certificates. When you add additional MSS servers, their certificates also need to be trusted.

**MSS** Configuration is required in the MSS Administrative Console > Configure Settings > Trusted Certificates > Trusted Sub-System.

#### TO SETUP TRUST BETWEEN MSS AND THE SESSION SERVERS

- Trust the new MSS server by importing the MSS certificate into the session server's truststore. See [Import a certificate into the session server's truststore](#).
- The new MSS needs to trust each of the session servers.
  - a. As the administrator, log into MSS.
  - b. From the left panel, click Configure Settings > Trusted Certificates.
  - c. Select Trusted Sub-System. The list contains the certificates that are trusted by MSS.
  - d. Click Import to add the session server certificate to the list.
  - e. Repeat this process for each session server.

### Add Additional Session Servers to My multi-MSS Installation

During the installation the session server and MSS have already exchanged and trusted their certificates, all MSS servers already trust all existing session servers. However, when you add more session servers, a trust relationship needs to be established between the new session servers and the existing MSS servers.

#### TO ADD MORE SESSION SERVERS

1. Import the MSS server certificate into the session server truststore.
2. Import the session server certificate into the MSS server truststore. See [General Security and Certificates](#) in the MSS documentation.
3. Retrieve the `service.registry.password` from the MSS server `container.properties` file.
4. Set the `service.registry.password` in the session server `container.properties` file.

#### More information

- [Import a certificate into the session server's truststore](#)
- [General Security and Certificates](#) in the MSS documentation



### Import a Certificate into the Session Server's Truststore

When the session server attempts to make outbound secure connections to remote servers it verifies the identity of the remote server using the certificates in its truststore. Any certificate imported into this truststore will be trusted.

#### Information you must know:

- keystore location - `/etc/trustcerts.bcfks`
- keystore format - `bcfks` (Bouncy Castle FIPS)
- default password - `changeit`

#### USING KEYSTORE EXPLORER

1. Open `trustcerts.bcfks` using the password `changeit`.
2. From the toolbar, select Import Trusted Certificate.

#### USING JAVA KEYTOOL

From the `sessionserver/etc` directory:

```
../../../../java/bin/keytool -importcert -alias <import-cert> -trustcacerts -file <import-cert.cer> -keystore trustcerts.bcfks -storetype bcfks
-storepass changeit -providername BCFIPS -providerpath ../lib/bc-fips-*.jar -providerclass
org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider
```

## 3.10 Using Docker

---

The Docker open platform has [excellent documentation](#) which you should read and understand.

### 3.10.1 Why Docker?

---

Docker is a container-based platform that enables you to develop, deploy, and run applications within a container. Your application, plus any dependencies your application requires, such as binaries and libraries, and configuration information are held within the container. You can deploy multiple containers, all of which run in Docker and on top of the operating system.

Using Docker you can scale your applications vertically, meaning multiple instances of the session server can exist on one server and each instance will perform exactly as it did when you created and tested it.

### 3.10.2 What are the benefits?

---

Containerization delivers multiple benefits:

- Performance

Virtual machines are an alternative to containers, however containers do not contain an operating system (unlike VMs). This means containers are faster to create, quicker to start, and have a much smaller footprint.

- Agility

Because containers are more portable and have better performance, you can take advantage of more agile and responsive development practices.

- Isolation

Docker containers are independent of one another. This is important because a Docker container containing one application, including the required versions of any supporting software, will not interfere with another container of the same application which requires different supporting software. You can have total confidence that at each stage of development and deployment the image you create will perform exactly as expected.

- Scalability

Creating new containers is simple and quick. The [Docker documentation](#) has information on how to manage multiple containers.

### 3.10.3 Terminology

---

There are basic terms you need to be familiar with when working with Docker. For more information see the Docker Documentation site.

#### **Container**

A run-time instance of an image. A container is usually completely isolated from the host environment, only able to access host files and ports if it has been configured to do so. To run an image in a container you use the `docker run` command.

#### **Docker Hub**

A cloud-based community resource for working with Docker. Docker Hub is typically used for hosting images, but can be used for user authentication and automating the building of images. Anyone can publish images to Docker Hub.

#### **Docker Compose**

Compose is a tool that uses YAML files to configure your application services and then define and run multi-container Docker applications. To learn more about Compose visit the Docker Compose documentation.

#### Dockerfile

A text document containing the commands to build a Docker image. You can specify complex commands (such as specifying an existing image to use as a base) or simple ones (such as copying files from one directory to another). To build an image from a Dockerfile you use the `docker build` command.

#### Image

A standalone, executable package that runs in a container. A Docker image is a binary that includes everything needed to run a single Docker container, including its metadata. You can build your own images (using a Dockerfile) or use images that have been built by others and then made available in a registry (such as Docker Hub). To build an image from a Dockerfile you use the `docker build` command. To run an image in a container you use the `docker run` command.

### 3.10.4 Getting started with Docker and Host Access for the Cloud

---

When you install HACloud, if you choose to use Docker, the install package contains an initial Dockerfile and accompanying application jar file to get you started using the session server in containers. These files are available prior to your installation.

#### Note

Make sure you are running the latest version of Docker and Docker Compose.

There are examples located in the `docker/samples` folder. See [Examples](#) for instructions.

There are four steps involved in creating the base image:

1. Install Docker. Follow the [instructions](#) on the Docker web site.
2. Extract the download package file and locate `Dockerfile`, `entrypoint.sh` and `sessionserver.jar` in the Docker folder. The `entrypoint.sh` file must have the executable bit set.
3. Build the Docker image.
4. Run the Docker image.

#### Build the Docker image

Assuming you have followed steps one and two; installed Docker and extracted and located `Dockerfile` and `sessionserver.jar`, the next step is to build the base Docker image of the session server.

1. Run this command from the folder containing the Dockerfile:

```
docker build -t hacloud/sessionserver:<version> .
```

Replace `<version>` with the version of the session server. If a version is not available, the default tag `(-t)` is `latest`.

2. Verify that the image was successfully created. Run:

```
docker images
```

The output should contain information about the image you just built.

### **Run the image**

Before you can run the session server image in a Docker container, you must complete the following steps:

- **Specify the address of the MSS server**

To specify the location of your MSS server, pass in an environment variable to the session server through Docker. For example, `--env MSS_SERVER=mss.server.com`

- **Specify the service registry password**

To specify the service registry password, pass in an environment variable to the session server through Docker. For example, `--env SERVICE_REGISTRY_PASSWORD=<your_password>`

You can retrieve the password from the `service.registry.password` property located in `./mss/server/conf/container.properties` on the MSS server. Use the entire `service.registry.password` property.

- **Tell MSS to trust the session server's identity certificate**

You accomplish this step using the Administrative Console > Configure Settings > Trusted Certificates. See the MSS Administrative Console documentation, [Trusted Certificates](#). The session server's certificate is available in the `sessionserver/etc` directory.

- **Provide the keystore containing the session server's identity certificate**

The session server identifies itself using a certificate. The certificate is expected to be present in the Java keystore `/sessionserver/etc/keystore.bcfks` located in the container. The servlet-engine key pair entry must contain the full certificate chain.

- **Provide the truststore containing the MSS certificate**

When the session server makes outbound TLS connections, it verifies the trust of the remote servers (such as MSS) using certificates in its truststore. The certificates present in the Java keystore `/sessionserver/etc/trustcerts.bcfks` located in the container will be trusted.

- **Mapping your keystore and truststore to the ones in the container**

You have two options to provide these keystores into the container:

- Using a volume mount
- or
- Extending an existing Docker image

#### Using a volume mount

A volume mount mounts a file or directory on the host machine into a container. The file or directory is referenced by its full or relative path on the host machine.

This volume mounts the keystore and truststore files on the host to the Docker container.

```
docker run -d \
  --env MSS_SERVER=<mss_server_name> \
  --env SERVICE_REGISTRY_PASSWORD=<enter password here> \
  --env MANAGEMENT_SERVER_URL=https://<mss_server_name>:<port>/mss \
  --env HOST_NAME=<docker server DNS name> \
  --volume ~/demo_keystore.bcfks:/sessionserver/etc/keystore.bcfks \
  --volume ~/demo_truststore.bcfks:/sessionserver/etc/trustcerts.bcfks \
  --publish 7443:7443 \
  sessionserver
```

#### Extending an existing Docker image

With this method you create a new Dockerfile to copy the files you need into the Docker image. This makes your Docker image more portable.

- First, create a Dockerfile that will extend from the `hacloud/sessionserver` Docker image.
- FROM `hacloud/sessionserver: <for example, hacloud/sessionserver:latest or hacloud/sessionserver:version>`
- COPY `<your-path>/keystore.bcfks /sessionserver/etc/keystore.bcfks`
- COPY `<your-path>/truststore.bcfks /sessionserver/etc/trustcerts.bcfks`
- Next, create the extended Docker image, naming it `demo`.

```
docker build -t demo .
```

- Finally, run the `demo` image.

```
docker run -d \
--env MSS_SERVER=<mss_server_name> \
--env SERVICE_REGISTRY_PASSWORD=<enter password here> \
--env MANAGEMENT_SERVER_URL=https://<mss_server_name>:<port>/mss \
--env HOST_NAME=<docker server DNS name> \
--publish 7443:7443 \
demo
```

- **Specify Docker host name and port**

The session server needs to broadcast its host name for MSS to find it. Because Docker generates a random unique name that isn't reachable outside the container, you need to specify your Docker host's name for MSS. It is also necessary to tell the session server which port you are publishing on your Docker host. Clients accessing the session server will end up hitting `<docker_host_name>:<docker_published_port>`.

```
--env HOST_NAME=docker_host_name
--env SERVER_PORT=docker_published_port
```

### 3.10.5 Examples

---

The examples, located in the `docker/samples` folder, walk you through four scenarios using Docker Compose. Compose is a tool that uses a YAML file to configure and run your applications with a single command.

#### Prerequisites

To run the examples:

- Install Docker Compose. Review the Docker documentation on Docker Compose before proceeding.
- A running MSS server
- A keystore file to secure TLS connections to the session server that is trusted by MSS.
- A truststore file that has the MSS server certificate in place
- To Build the session server Docker image

The examples include:

- **Basic** - A basic example providing a demo keystore and truststore files in which you can import a MSS server certificate.
- **Hybrid** - An hybrid example which assumes a local Host Access for the Cloud installation and mounts existing, on disk, keystore and truststore files to the Docker container.
- **Extensions** - An extension example showing how to update, modify, and customize the web client.
- **Load Balance** - A load balancer example illustrating how to balance between linked containers.

#### Basic

This basic example illustrates how to run the session server Docker image in Docker Compose. For this example you will need to import your MSS server's certificate into the provided sample `./certs/demo_truststore.bcfks` using something like KeyStore Explorer. Your MSS certificate, by default, is located at `/mss/server/etc/<computer-name>.cer`. See [Import a certificate](#) into the session server's truststore.

Before running the example, update the `MSS_SERVER`, `HOST_NAME`, and `SERVICE_REGISTRY_PASSWORD` values in `docker-compose.yml`.

- To start the session server service:

```
docker-compose up
```

- To run the service in a daemon (detached mode):

```
docker-compose up -d
```

- To look at running containers:

```
docker ps
```

## Hybrid

In this example a local installation of Host Access for the Cloud, with existing keystore and truststore files on disk is present. These files will be mounted (copied) to the Docker container.

Before running the example, update the `MSS_SERVER`, `HOST_NAME`, `SERVER_PORT`, and `SERVICE_REGISTRY_PASSWORD` values in the `.env` file.

To start the session server service:

- Copy `.env` and `docker-compose.yml` to `sessionserver/microservices/sessionserver/`.
- From this directory, run: `docker-compose up -d`

## Extensions

Using extensions and your own HTML, CSS, or JavaScript, you can update, modify, and customize the presentation of the web client from within the browser. See [Extending the Web Client](#) for more information.

This example sets `SPRING_PROFILES_ACTIVE` to `extensions-enabled` and maps the location of the extensions in `docker-compose.yml`.

Before running the example, update the `MSS_SERVER`, `HOST_NAME`, and `SERVICE_REGISTRY_PASSWORD` values in the `.env` file.

To start the session server service:

```
docker-compose up -d
```

You could also choose to extend the base `hacloud/sessionserver` Docker image and copy the extension files into the Docker container:

1. Create the Dockerfile that extends from the `hacloud/sessionserver` Docker image.

```
FROM hacloud/sessionserver

COPY ./certs/keystore.bcfks /sessionserver/etc/keystore.bcfks
COPY ./certs/trustcerts.bcfks /sessionserver/etc/trustcerts.bcfks
COPY ./extensions /sessionserver/extensions/
```

2. Create the extended Docker image and name it `extensions`.

```
docker build -t extensions .
```

3. Update `docker-compose.yml` to use the new `extensions` image

```
version: '3'
services:
  sessionserver:
    image: extensions
    environment:
      - LOGGING_FILE_NAME=./logs/sessionserver.log
      - LOGGING_FILE_MAXSIZE=10MB
      - LOGGING_FILE_MAXHISTORY=10
      - MSS_SERVER=${MSS_SERVER}
      - SERVICE_REGISTRY_PASSWORD=${SERVICE_REGISTRY_PASSWORD}
      - SPRING_PROFILES_ACTIVE=extensions-enabled
    ports:
      - ${SERVER_PORT}:7443
```

## Load Balance

HAProxy is a load balancer. Learn more about HAProxy on their web site.

In this example, an haproxy service is included in the `docker-compose.yml` file. The example uses an haproxy image to balance between linked containers. This example uses SSL Bridging to link the containers.

To provide secure communication between the clients and the load balancer, you must update the `LOAD_BALANCER_CERT` property in the `.env` file with the location of the load balancer certificate.

To help you test, you can generate a self-signed certificate:

1. Generate a unique private key (KEY):

```
sudo openssl genrsa -out mydomain.key 2048
```

2. Generate a Certificate Signing Request (CSR):

```
sudo openssl req -new -key mydomain.key -out mydomain.csr
```

3. Create a Self-Signed Certificate (CRT):

```
sudo openssl x509 -req -days 365 -in mydomain.csr -signkey mydomain.key -out mydomain.crt
```

4. Append KEY and CERT to `loadbalancer.pem`:

```
sudo cat mydomain.key mydomain.crt >./certs/loadbalancer.pem
```

To start the session server and haproxy services:

```
docker-compose up -d
```

-OR-

```
docker-compose up --scale sessionserver=n -d
```

Where `n` is the number of session server instances.

You can change the number of session server instances after the services start:

```
docker-compose scale sessionserver=n
```



To access the session server and HAProxy stats page:

- <https://server:7443>
- <http://server:1936/haproxy?stats>

Using:

- user: *admin*
- password: *password*

## 4. Managing

---

### 4.1 Managing

---

Creating and configuring sessions and making sure everything runs smoothly and securely means that your users will be successful. The following should help you administer and manage your sessions and host connections.

- [Creating Host Sessions](#)
- [Providing Access to Host Sessions](#)
- [Managing User Preferences](#)
- [Customizing Host Sessions](#)
- [Logging](#)

## 4.2 Creating Host Sessions

---

Host Access for the Cloud supports IBM 3270, 5250 and VT hosts as well as UTS, T27 and ALC host types.

Your users gain access to the host through sessions that you create and configure. Sessions are created by an administrator in the MSS Administrative Console. When you launch a session from the Administrative Console, the web client Connection panel opens in a separate browser window. You configure connection options from this panel. Options vary depending on your host type.

### To create a session:

1. Create sessions in the MSS Administrative Console. See [Add a Session](#) in the MSS documentation.
2. In the HACloud web client, on the **Create New Session** dialog box, select the type of host you are connecting to from the drop down list.
3. On the Connection panel, identify the name of the host to which you want to connect. You can use a full host name or its IP address.
4. Type the number of the port you want to use.
5. Complete the information needed for the host connection.
6. Save your connection settings.
7. After you complete configuring and testing your session, click Exit to return to the MSS Administrative Console.
8. Using the [Assign Access](#) view in the MSS Administrative Console, assign the session you created to your users.
9. Once sessions have been assigned to your users, you can let them know [how to access their sessions](#).

---

Check the settings for your session type.

- [Common connection settings](#)
- [3270 and 5250 connection settings](#)
- [VT connection setting](#)
- [UTS connection settings](#)
- [T27 connection settings](#)
- [ALC connection settings](#)

## 4.3 Connection Settings

---

### 4.3.1 Connection Settings

---

Host Access for the Cloud supports IBM 3270, 5250 and VT hosts as well as UTS, T27 and ALC host types.

Your users gain access to the host through sessions that you create and configure. You configure connection options from the Settings panel. There are a number of settings that are common to all host types, otherwise options vary depending on your host type.

**More information**

- [Common connection settings](#)
- [3270 and 5250](#)
- [VT](#)
- [UTS](#)
- [T27](#)
- [ALC](#)

## 4.3.2 Common Connection Settings

---

These options are common to all supported host types.

- **Connect at startup**

By default, sessions are configured to connect to the host automatically when you create or open a session. However, you can set up a session so that it doesn't automatically connect to the host. Choose NO to manually connect to the host.

- **Reconnect when host terminates connection**

When set to Yes, Host Access for the Cloud attempts to reconnect as soon as the host connection terminates.

- **Protocol**

From the drop down list, select the protocol you want to use to communicate with the host. To establish a host connection, both the web client and the host computer must use the same network protocol. The available values are dependent on the host to which you are connecting. They are:

Protocol	Description
TN3270	TN3270 is a form of the Telnet protocol, which is a set of specifications for general communication between desktop and host systems. It uses TCP/IP as the transport between desktop computers and IBM mainframes.
TN3270E	TN3270E or Telnet Extended is for users of TCP/IP who connect to their IBM mainframe through a Telnet gateway that implements RFC 1647. The TN3270E protocol allows you to specify the connection device name (also known as LU name), and provides support for the ATTN key, the SYSREQ key, and SNA response handling. If you try to use Telnet Extended to connect to a gateway that doesn't support this protocol, standard TN3270 will be used instead.
TN5250	TN5250 is a form of the Telnet protocol, which is a set of specifications for general communication between desktop and host systems. It uses TCP/IP as the transport between desktop computers and AS/400 computers.
Secure Shell (VT)	<p>You can configure SSH connections when you need secure, encrypted communications between a trusted VT host and your computer over an insecure network. SSH connections ensure that both the client user and the host computer are authenticated; and that all data is encrypted. Two authentication options are available:</p> <ul style="list-style-type: none"> <li>• <b>Keyboard Interactive</b> - You can use this authentication method to implement different types of authentication mechanisms. Any currently supported authentication method that requires only the user's input can be performed with Keyboard Interactive.</li> <li>• <b>Password</b> - This option prompts the client for a password to the host after a host connection is made. The password is sent to the host through the encrypted channel.</li> </ul>
Telnet (VT)	Telnet is a protocol in the TCP/IP suite of open protocols. As a character stream protocol, Telnet transmits user input from character mode applications over the network to the host one character at a time, where it is processed and echoed back over the network.
INT1 (UTS)	Provides access to Unisys 1100/1200 hosts using the TCP/IP network protocol.
TCPA (T27)	Use this protocol to connect to Unisys ClearPath NX/LX series or A Series hosts. TCPA Authentication is the process of verifying user login information. When properly configured, you can request a security credential from your application's credential server and send the credential back to the server. If the credential is valid, your application will be logged in; you do not have to enter a user ID or password. If the credential is not valid however, you will be prompted for a user ID and a password.
MATIP (ALC)	Mapping of Airline Traffic Over Internet Protocol (MATIP) uses TCP/IP for airline reservation, ticketing, and messaging traffic.

## • TLS Security

TLS protocols allow a client and server to establish a secure, encrypted connection over a public network. When you connect using TLS, Host Access for the Cloud authenticates the server before opening a session, and all data passed between and the host is encrypted using the selected encryption level.

### Tip

When TLS security is set to TLS 1.3 or TLS 1.2, you have the option to verify the host name against the name on the server certificate. It is highly recommended that you enable host name verification for all sessions.

The following options are available:

Security Options	Description
None	No secure connection is required.
TLS 1.3	Connect using TLS 1.3. When <b>Verify server identity</b> is set to <b>Yes</b> , the client checks the server or host name against the name on the server certificate. It is highly recommended that you enable host name verification for all sessions.
TLS 1.2	Connect using TLS 1.2. When <b>Verify server identity</b> is set to <b>Yes</b> , the client checks the server or host name against the name on the server certificate. It is highly recommended that you enable host name verification for all sessions.

### Note

See the section on [Secure connections](#) for information on adding trusted certificates, key stores, using SSH, and other advanced security information.

## • Enable emulation tracing

You can choose to generate host traces for a session. No is the default. Select Yes to create a new emulation host trace each time the session is launched. The trace file is stored in `<install directory>/sessionserver/logs/hosttraces/<date(yyyymmdd)>/<trace-file>`.

## Using Terminal ID Manager

**MSS** To use Terminal ID Manager, you must have a Terminal ID Manager server configured. See [Set Up the Terminal ID Manager](#).

If you selected **Terminal ID Manager**, you can use it to provide IDs to client applications at runtime and to manage pooled IDs for different host types. An ID is connection data that is unique for an individual host session.

If you decide to use Terminal ID Manager and have configured the Terminal ID Manager server, then you can select from the options below to configure the criteria for acquiring an ID. All criteria must be met in order for an ID to be returned.

 **Note**

Keep in mind that by specifying a criterion, you are indicating that the ID should be allocated only when an ID with that specific value is found. The set of criteria selected here must be an exact match of the set of criteria specified on at least one Pool of IDs in Terminal ID Manager before the ID request can succeed.

**Terminal ID Manager Criteria**

Criterion	Description
Pool name	Include this attribute and enter the name of the pool to limit the ID search to a specified pool.
Client IP address	The IP address of the client machine will be included as part of the request for an ID.
Host address	The address of the host configured for this session will be included as part of the request for an ID.
Host port	The port for the host configured for this session will be included as part of the request for an ID.
Session name	When selected, requires that the ID is configured to be used by this session exclusively.
Session type	The session type (for example, IBM 3270, IBM 5250, UTS, ALC or T27) is always included as part of any request for an ID.
User name	<p>Use this criterion to ensure that only IDs created for exclusive use by specific users will be allocated. The current user's name, which must be found on an ID before it can be allocated, is the name of the user that the session is allocated to at runtime.</p> <p>To configure a session based on user names, a default place holder user name is available: tidm-setup.</p> <p>For the administrator to configure sessions using tidm-setup, the Terminal ID Manager must have IDs provisioned for tidm-setup. You can override the default name with one of your own by modifying the <code>&lt;install-dir&gt;/sessionserver/conf/container.properties</code> file as follows:</p> <pre>id.manager.user.name=custom-username</pre> <p>where <code>custom-username</code> is replaced by the name you want to use.</p>
Application name (UTS)	The name of the host application will be used as part of the request for an ID.



To determine the connection attempt behavior if Terminal ID Manager does not successfully allocate an ID to this session, use **If ID is not allocated**:

- **Fail connection attempt** - When selected, the session does not attempt to connect when an ID is not allocated.
- **Allow connection attempt** - When selected, the session attempts to connect when an ID is not allocated. The attempt may be rejected by the host. There are some host types that permit a user to connect without an ID.

To confirm that Terminal ID Manager can provide an ID using the criterion and value selections you have made, click [Test Terminal ID Manager Criteria](#).

- **Send keep alive packets** - Use this setting to provide a constant check between your session and the host so that you become aware of connection problems as they occur. Choose from the following types of keep alive packets:

This option	Does this ...
None	The default. No packets are sent.
System	The TCP/IP stack keeps track of the host connection and sends keep alive packets infrequently. This option uses fewer system resources than the Send NOP Packets or Send Timing Mark Packets options.
Send NOP packets	Periodically, a No Operation (NOP) command is sent to the host. The host is not required to respond to these commands, but the TCP/IP stack can detect if there is a problem delivering the packet.
Send timing mark packets	Periodically, a Timing Mark Command is sent to the host to determine if the connection is still active. The host should respond to these commands. If a response is not received or there is an error sending the packet, the connection shuts down.

- **Keep alive timeout (seconds)** - If you choose to use either the Send NOP packets or the Send timing mark packets option, select the interval between the keep alive requests set. The values range from 1 to 36000 seconds (1 hour); the default is 600 seconds.

### Test Terminal ID Manager Criteria

The Terminal ID Manager provides IDs to client applications at runtime. To confirm that Terminal ID Manager can provide an ID using the criteria and value selections you selected use this test option.

Criteria for the current session are specified on the Connection panel after selecting **Use Terminal ID Manager** from either the Device Name (3270, 5250 host types), the Terminal ID (UTS) field, or the Station ID (T27) field. By default, the selected criteria for the current session are displayed.

Click **Test** to confirm that Terminal ID Manager can provide an ID matching the configured criterion and value selections. The test returns the name of an available ID that satisfies the selected attribute values.

#### Testing for other criteria and values

You can also use this panel to test criteria different from those associated with the current session.

1. Select any of the session types from the Session type list, and select the criteria you want to test. You can test alternate values that you want to use in a sample Terminal ID Manager request.
2. Click **Test** to confirm that Terminal ID Manager can provide an ID matching the criterion and value selections. The test returns the name of an available ID that satisfies the selected values.

### 4.3.3 3270 and 5250 connection settings

---

In addition to the [Common connection settings](#), 3270 and 5250 host types require these specific settings.

- **Terminal model**

Specify the terminal model (also known as a display station) you want Host Access for the Cloud to emulate. There are different terminal models available depending on the host type.

If you choose **Custom Model**, you can set the number of columns and rows to customize the terminal model.

- **Use Kerberos automatic sign-on (5250 only)** 

When set to **Yes** the user does not have to enter sign-on credentials. Kerberos automatic sign-on is configured in the MSS Administrative Console > Host Access for the Cloud. In configuring HACloud to use the Kerberos authentication protocol there are terms that you should understand and prerequisites to adhere to in advance of configuring this option. These options are explained in detail in the MSS Administrative Console > Host Access for the Cloud panel documentation, available from the Help button. See [Set Up Kerberos for AS/400 Single Sign-on](#) for more information.

- **Terminal ID (3270 only)**

When Host Access for the Cloud connects to a Telnet host, the Telnet protocol and the host negotiate a terminal ID to use during the initial Telnet connection. In general, this negotiation will result in the use of the correct terminal ID, and so you should leave this box empty.

- **TLS Security**

TLS protocols allow a client and server to establish a secure, encrypted connection over a public network. When you connect using TLS, Host Access for the Cloud authenticates the server before opening a session, and all data passed between and the host is encrypted using the selected encryption level. See [Common Connection Settings](#) for detailed information on this common setting.

- **Device name**

If you selected TN3270, TN3270E, or TN5250 as the protocol, specify the device name to use when the session connects to the host. The device name is also known as the host LU or pool. You can also choose to:

- **Generate a unique device name** - Automatically generates a unique device name.
- **Use Terminal ID Manager** - Displays additional settings to complete. See [Using Terminal ID Manager](#)
- **Always Prompt the User for ID** - The end user is prompted for the device ID each time a connection is attempted.
- **Prompt the User if ID not Specified** - The end user is prompted the first time a connection is attempted after which the value is saved. The saved value will continue to be used without additional prompting.

If you do not specify a device name for the session, the host dynamically assigns one to the session. A device name that is set within a macro will override this setting.

## 4.3.4 VT connection settings

---

In addition to the [Common connection settings](#), VT hosts require additional settings. These settings vary depending on the protocol you are using; Telnet or SSH. The settings are applicable to both protocols unless noted.

## VT session configuration options

VT Settings	Description
Terminal ID	This setting determines the response that Host Access for the Cloud sends to the host after a primary device attributes (DA) request. This response lets the host know what terminal functions it can perform. The Host Access for the Cloud response for each Terminal ID is exactly the same as the VT terminal's response; some applications may require a specific DA response. This terminal ID setting is independent of the Terminal type setting. The options are: VT220, VT420, VT100, DEC-VT100, and VT52.
Allow unknown hosts (SSH)	This setting provides the administrator with the ability to decide whether the web client will allow unknown hosts. Options are: <ul style="list-style-type: none"> <li>• Yes - Unknown hosts and all SSH connections are permitted. Web client users are not prompted about whether hosts should be trusted.</li> <li>• Ask - The web client user is prompted whether the host should be trusted when they connect to an unknown host they haven't encountered before. If they choose to trust the host, then its public key is stored in their user preferences and subsequent connections will not elicit a prompt unless the host key changes.</li> <li>• No - No unknown hosts are permitted. Only those hosts the administrator chooses to trust when configuring the session are permitted. End users are never prompted and the session will either connect or not connect depending on the administrator's choices.</li> </ul>
Suppress banner messages (SSH)	When enabled, the SSH banner is not displayed. This option is useful when recording SSH login macros.
Local Echo (Telnet)	Automatic (default). How Host Access for the Cloud responds to remote echo from a Telnet host: Automatic attempts to negotiate remote echo, but does what the host commands. Yes means Host Access for the Cloud negotiates local echo with the host, but always echoes, while No means Host Access for the Cloud negotiates remote echo with the host, but does not echo.
Renegotiate Echo (Telnet)	No (default). When set to Yes, passwords are not displayed on the local screen, but all other typed text is visible. Host Access for the Cloud supports the Telnet Suppress Local Echo (SLE) option when connected to a host in half-duplex mode. This means that Host Access for the Cloud will suppress character echo to the host computer, and with SLE support Host Access for the Cloud can be instructed to suppress echo locally.
Set Host Window Size	Yes (default). This setting sends the number of rows and columns to the Telnet host whenever they change. This enables the Telnet host to properly control the cursor if the window size is changed.
Request Binary (Telnet)	No (default). Telnet defines a 7-bit data path between the host and the terminal. This type of data path is not compatible with certain national character sets. Fortunately, many hosts allow for 8-bit data without zeroing the 8th bit, which resolves this problem. In some cases, however, it may be necessary to force the host to use an 8-bit data path by selecting this check box.
Send LF after CR (Telnet)	No (default). A "true" Telnet host expects to see a CrNu (carriage return/ null) character sequence to indicate the end of a line sent from a terminal. There are some hosts on the Internet that are not true Telnet hosts, and they expect to see a Lf (linefeed) character following the Cr at the end of a line. If you're connecting to this type of Telnet host, select Yes.
Ctrl-break sends (Telnet)	Choose what sequence Ctrl-break sends to the host when pressed. Options are: Telnet break sequence (the default), Interrupt process, or Nothing.

Host Character Set	The default value for the Host character set depends on the type of terminal you are emulating. This setting reflects the current terminal state of VT Host Character Set, which can be changed by the host. The associated default setting, saved with the model is DEC Supplemental.
Auto Answerback	No (default). This setting specifies whether the answerback message (set with the Answerback property) is automatically sent to the host after a communications line connection.
Answerback String	<p>This setting allows you to enter an answerback message if the host expects an answer in response to an ENQ character.</p> <p>The answerback string supports characters with codes less than or equal to 0xFFFF via Unicode escape sequences. The escape sequence begins with \u followed by exactly four hexadecimal digits. You can embed Unicode escape sequences in any string. For example, this embedded \u0045 will be interpreted as this embedded E, since 45 is the hexadecimal code for the character E.</p> <p>To pass Unicode escape sequences to the host, escape the sequence with a leading backslash. For example, to send the string literal \u001C to the host, map a key to \\u001C. Host Access for the Cloud will convert this to the string \u001C when that key is pressed and send the 6 characters of the resulting string to the host.</p>

#### More information

- [TLS descriptions](#)

### 4.3.5 UTS connection settings

---

In addition to the [Common connection settings](#), UTS hosts require these additional settings:

## UTS INT1 session configuration options

UTS INT1 options	Description
Application	The name of the host application or host operating mode to be accessed. This is the word or phrase that the local machine sends to the host when you first establish communication with the host. If you were using a host terminal, this would be the \$\$OPEN name of the application. The application name is typically the same as the environment name. However, they can be different. For example, the environment name might be MAPPER, and the application might be UDSSRC. During a terminal emulation session, you would type \$\$OPEN MAPPER at the prompt, and INT1 would send UDSSRC to the host once the connection is established.
TSAP	The desired Transport Service Access Point (TSAP), up to 32 characters (such as TIPCSU for TIP connections, RSDCSU for Demand connections). A TSAP is required only if you are connecting to a Host LAN Controller (HLC) or to a Distributed Communications Processor (DCP) in IP router mode. If you're not sure which value to use, contact your host administrator.
Initial transaction	The character, word, or phrase that the local machine will send to the host when communication with the host is first established (up to 15 characters). This parameter is optional and is primarily used with TIP. For example, you might type ^ to run MAPPER. This parameter can also be used to transmit passwords.
Start transaction	When you configure an initial transaction, by default, the data is sent as soon as the session connection is established. You can decide when to send an initial transaction by using a particular string to trigger the initial transaction. For example, to wait for a successful login before sending initial transaction data, type in a string to be used to identify a successful login. You can use this setting in combination with <b>Send initial transaction</b> .
Send initial transaction	You can determine when the initial transaction is sent: <ul style="list-style-type: none"> <li>• <b>Immediately</b> - Default.</li> <li>• <b>When start of entry (SOE) character is received</b> - Useful when multi-line transactions must be completed before sending the string.</li> <li>• <b>After specified milliseconds</b></li> </ul>
Terminal ID	Choose options to specify a terminal ID or to use the Terminal ID Manager. To specify a terminal ID, type it in the <b>Specify Terminal ID</b> field. <ul style="list-style-type: none"> <li>• <b>Specify Terminal ID</b> The Terminal ID, a terminal identifier (typically up to 8 alphanumeric characters) to use for the communication session associated with this path. Also known as a TID or PID, each terminal ID should be unique to the host.</li> <li>• <b>Prompt the User if ID not Specified</b> The end user will be prompted the first time a connection is attempted after which the value is saved. The saved value will continue to be used without additional prompting.</li> <li>• <b>Always prompt the user for ID</b> When you select this option the end user will be prompted for the terminal ID each time a connection is attempted.</li> <li>• <b>Use Terminal ID Manager</b> If you choose <b>Use Terminal ID Manager</b>, you are prompted to select the Terminal ID attributes you want to use to obtain an ID. See Terminal ID Manager Attributes.</li> </ul> <p>To test the attributes, click <b>Test</b>.</p>



**More information**

- [Terminal ID Manager Attributes](#)
- [TLS Descriptions](#)

## 4.3.6 T27 connection settings

---

Along with the [Common connection settings](#), you can configure these additional T27 connection options:

### T27 Connection Settings

T27 options	Description
Terminal type	Select the type of terminal to emulate during the session. T27 emulation supports Unisys TD830, TD830 ASCII, TD830 INTL, and TD830 NDL terminal types.
Request binary	You must enable the Request binary option when you require pass through printing. The default is No. TCPA defines a 7-bit data path between the host and the terminal emulator. This type of data path is not compatible with certain national character sets. However, many hosts allow for 8-bit data without zeroing the 8th bit, which resolves this problem. However, it may be necessary to force the host to use an 8-bit data path; you can do so by selecting this option.
Line width	Select the number of characters the host will send to the client. The default is 80 characters.
TLS security	See <a href="#">TLS Descriptions</a> for a description of the various options.
Station ID	Choose an option to specify a station ID or use the Terminal ID Manager. To specify a station ID, choose <a href="#">Specify Station ID</a> and type the name in the Station ID field. Each station id should be unique to the host and typically consists of up to eight alphanumeric characters. <ul style="list-style-type: none"> <li>• Prompt the user if ID not specified The end user will be prompted the first time a connection is attempted after which the value is saved. The saved value will continue to be used without additional prompting.</li> <li>• Always prompt the user for ID When you select this option the end user will be prompted for the station ID each time a connection is attempted.</li> <li>• Use Terminal ID Manager If you select Use Terminal ID Manager, you will see a number of Terminal ID criteria to configure. See <a href="#">Terminal ID Manager Criteria</a> for descriptions of the various options.</li> </ul> If you do not specify a station id for the session, the host dynamically assigns one to the session.

### More information

- [TLS Descriptions](#)
- [Terminal ID Manager Criteria](#)

## 4.3.7 ALC connection settings

---

In addition to the [Common connection settings](#), ALC hosts require these additional settings:

### ALC Connection Settings

ALC options	Descriptions
TLS security	See <a href="#">TLS Descriptions</a> for a description of the various options.
Character encoding	Choose ASCII, EBCDIC, or IPARS (default) as the code set.
Configuration file	Enter the configuration (CNF) file that associates configuration information appropriate for a specific host type.
Terminal address	<p>Select whether you want to specify the terminal address or use the Terminal ID Manager.</p> <ul style="list-style-type: none"> <li>• <b>Terminal address</b> - Specify whether to use 2-byte or 4-byte addressing mode. Although a unique 5-byte address is required when you specify the terminal ID instead of using ID Manager, this option specifies how many bytes of the 5-byte terminal ID address are sent with each message for the purposes of multiplexing. If you specify 2-byte addressing mode, only the last 2 bytes of the ASCU (Agent Set Control Unit) cluster address (A1, A2) are sent. If you specify 4-byte addressing mode, the full ASCU cluster address (H1, H2, A1, A2) is sent. Specify the unique 5-byte terminal address for this session. The terminal address is made up of five 2-hex-digit values in this order: H1, H2, A1, A2, and TA (terminal address). This unique address is usually assigned by the network administrator.</li> <li>• <b>Terminal ID Manager</b>- Provides IDs to client applications at runtime. If you choose this option, there are additional configuration options to complete. See <a href="#">Terminal ID Manager Criteria</a> for descriptions of those options.</li> </ul>

### More information

- [TLS Descriptions](#)
- [Terminal ID Manager Criteria](#)

## 4.4 Providing Access to Host Sessions

---

Your end users access sessions either through a session server or through the Assigned Sessions list portal. With both options, once authenticated, your users are presented with a list of sessions that they can access and successfully launch.

### Tip

Using a load balancer for high availability and scalability is highly recommended. See [Planning for Deployment](#) for more information.

### 4.4.1 Session Servers

---

Users most commonly access their sessions by going to the session servers, typically through a load balancer.

End user access on a session server is available at `https://<sessionserver>:7443/`

### 4.4.2 Assigned Sessions List

---

Using the Assigned Sessions list, users can launch all of their sessions from a consolidated HTML-based portal. After a user authenticates, they see their list of assigned sessions.

The Assigned Sessions list is available at `https://<mssserver>/sessions/`

See [Configuring the Assigned Sessions list](#) for information on how to configure the Assigned Sessions list.

## 4.5 Managing User Preferences

---

As an administrator you can choose what options users can configure for their sessions. These options are set on a per session basis and all users who have access to a particular session can configure their own session instance.

1. From the left navigation panel, choose **User Preference Rules**.
2. Select which options you want to allow your users to configure.
3. Click Save.

Each user's configurations are specific to their instance of the session and will not conflict with those of other users.

There is a **Restore Defaults** option available on the various settings and display panels. As an administrator, this option restores the web client back to its default settings. For end users this option will restore the values set by the administrator when the session was created.

### **Warning**

When the authentication method is set to None, be aware that all users share the same settings. During session configuration, it is best to not allow users to modify their session settings (User Preference Rules), because they can overwrite each other's choices. To workaroud this constraint, it is possible to [provide user identities in other ways](#).

### More information

- [Display Settings](#)
- [Specify Edit Options](#)
- [Transfer Files](#)
- [Creating Macros](#)

## 4.6 Customizing Host Sessions

---

### 4.6.1 Customizing Host Sessions

---

You can use these features to customize sessions for your end users:

- **Plus** - Enable custom controls to provide a more efficient work flow and a more modern and friendly interface. See [Use Plus to customize screens](#).

Using this option, you can add tool tips to fields, replace old-style numbered lists with more modern drop-down lists, add buttons to the host interface and program them to start macros or perform other actions, and replace manual date entry with a graphical calendar date-picker.

- **Server-side Events** - Supply procedural Java code that extends and improves the presentation of host data.

Using server side events, you can define specific events and suspend the host application, replacing or interrupting it with code that you have supplied to the session, as well as extend error handling options. For example, you can add an event that recognizes when an error occurs and then implements the code to intercept the error, take control, and correct the error. See [Server-side events](#).

- **Advanced** - Only use as directed by Micro Focus Technical Support.

These options are configured on the Customization panel.

1. Click Settings on the toolbar to open the left navigation panel.
2. Click Customization.

---

#### More information

- [Use Plus to customize screens](#)
- [Use server-side events](#)

## 4.6.2 Use Plus to customize screens

---

### Note

The Plus feature requires archive files ( `.rdar` ) produced by Micro Focus Screen Designer version 9.5 or higher. The Screen Designer is available in Micro Focus Rumba Desktop 9.5. Reflection Desktop 16.1 includes a limited version of the Screen Designer. To get access to more controls and full use of Plus and the Screen Designer, you can purchase and install the Micro Focus Reflection Desktop Plus add-on.


1. On the **Customization** panel, click **Enable Plus**.
2. Select the Plus archive file you want to use from the drop-down list or upload a file from a different location. Plus archive files are identified by a `.rdar` file extension.

Archive files are the output of a Screen Designer project and are used to provide the custom control criteria.

If you are updating the Plus archive ( `.rdar` ) file associated with your Plus enabled session, you must first delete the folder containing the old `.rdar` file from the session server. After you delete the folder, you can open your Plus enabled session and the new `rdar` file will be downloaded to the session server.

3. Verify that the number of milliseconds for the host settle delay time is accurate. This is the time that the server waits for a synchronous connection before deciding that the host has finished sending data.

4.

When you return to your session, Plus is available. Click  on the toolbar to turn off the custom controls.

When you enable Plus for a session, all end users of that session see the Plus icon on the toolbar and any controls made available through the Screen Designer customization file.

---

### More information

- [Customizing Host Sessions](#)

### 4.6.3 Use server-side events

---

Using server-side events, you can supply procedural Java code that can extend and improve the presentation of host data.

The Customization panel tells the web client where to find the event after you configure it. See [Using the Java SDK](#) for instructions on using the SDK and the samples available to you.

1. Open the **Customization** panel.
2. Under **Server Side Events**, type the fully-qualified class name to the event.
3. Launch the session and test the event.

Access [API documentation and event samples](#).

---

#### More information

- [Customizing Host Sessions](#)
- [Using the Java SDK](#)
- [Developing](#)



## 4.7 Logging

---

### 4.7.1 Locating log files

---

There are two log files available:

- `<install_dir>/sessionserver/sessionserver.log` - the log file for the session server application.
- `<install_dir>/sessionserver/container.log` - the log file for the container that hosts the Host Access for the Cloud application.

### 4.7.2 Configuring log rotation

---

You can configure log rotation by editing these values in

`<install_dir>\sessionserver\microservices\sessionserver\service.yml` :

```
LOGGING_FILE_MAXSIZE
LOGGING_FILE_MAXHISTORY
```

### 4.7.3 Setting logging levels

---

There are various types of logging levels you can use to produce different types of information. You can configure logging levels in `<install_dir>\sessionserver\microservices\sessionserver\service.yml`.

#### Note

Lines in `service.yml` must be indented using spaces.

Use the following format to set logging levels:

```
- name: logging.level.<logger>
  value: "<log level>"
```

Where `<logger>` is the name of the logger to adjust and `<log level>` is one of the following:

- **Trace** - designates finer-grained informational events than Debug
- **Debug** - designates fine-grained informational events that are most useful to debug an application.
- **Info** - designates informational messages that highlight the progress of the application at coarse-grained level.
- **Warn** - designates potentially harmful situations.
- **Error** - designates error events that might still allow the application to continue running.
- **Fatal** - designates very severe error events that will presumably lead the application to terminate.

#### Note

You must restart the session server after any changes to `service.yml`.

### 4.7.4 Enabling Web client-to-session server logging

---

While the browser provides a basic mechanism for logging to its JavaScript console, the Web Client extends this capability and, with some configuration, you can log events to the session server for viewing by an administrator.

By default, nothing is logged to the session server. **You must set the log level**, following the instructions below, in order to enable this feature.

The available log levels are: `debug`, `info`, `warn`, `error`, or `off`. The default log level is `off`.

### Adjusting the logging level for all Web client users

To adjust the logging level for all Web clients, add the following entry to

```
<install_dir>\sessionserver\microservices\sessionserver\service.yml
```

```
-name: <logger>
value: "<log level>"
```

Where `<logger>` is:

```
logging.level.com.microfocus.zfe.webclient.core.handler.ClientLoggingHandler-webclient
```

#### Note

Use caution when increasing the logging level for all Web Client users in a production environment due to a potential increase in network traffic.

### Adjusting the logging level for an individual user

There are two options for adjusting the logging level for individual users:

- To temporarily adjust the logging level for a particular user's Web client instance without requiring a session server restart, instruct the user to add the following URL parameter when loading the Web client in their browser:
  - `https://mysessionserver.com:7443/?log=<log level>` -
- To adjust the logging level for an individual user without requiring them to make changes, add the following entry to `service.yml`:

```
-name: <logger>
value: "<log level>"
```

Where `<logger>` is:

```
logging.level.com.microfocus.zfe.webclient.core.handler.ClientLoggingHandler-webclient-<username>
```

and where `<username>` is the user name of the person whose logging levels you are adjusting.

#### Note

Logging based on a username requires an authentication mode that involves usernames.

## 5. Using HACloud

---

### 5.1 Using Host Access for the Cloud

---

There are multiple session and display options available so you can personalize your session and make sure you are working efficiently.

- [Display Settings](#)
- [Map Keys](#)
- [Configure User Macros](#)
- [Transfer Files](#)
- [Specify Edit Options](#)
- [Working with Sessions](#)
- [Creating Macros](#)
- [Printing](#)

## 5.2 Display Settings

---

Display settings vary depending on the host type and are specific to the session you are configuring.

### 5.2.1 Color Mapping

---

You can customize the color of your screen and the appearance of different host attributes in the terminal window. For each item, you can select a color for the foreground and the background colors for all supported host connections. Colors are specified using the color grid or by entering the Hex code format.

There are many web sites that list available Hex colors, for an example see [w3schools.com HTML Color Picker](https://www.w3schools.com/html/html_color_picker.asp).

You may see different options depending on the type of host connection.

#### Options specific to UTS hosts

- **Use color information from the host** - To use the colors specified here rather than any colors specified by the host, clear this option.
- **Enable blink** - To disable blinking, clear this option.
- **Select attribute to edit** - In UTS emulation, colors are set directly by the host. You can specify colors for text associated with specific screen display options. Including the following and available combinations:  
Plain, Underline(UND), Strikethru (STK), Left Column Separator (LCS), Control Page, and Status Line (OIA).
- **Video intensity** - The video intensities, Blink, Dim, Protected, and Reverse are combined with the attributes to create additional combinations. For example, you could map foreground or background colors to all cells with Dim + Blink + Underline or Reverse + Protected + Strikethru + Underline.

When you select a video intensity (or combination of intensities), those intensities are combined with the value of the attribute drop down list to form a single color mapping.

#### Options specific to VT and T27 hosts

- **Enable blink** - To disable blinking, clear this option.
- **Enable bold** - Displays text set with bold attributes as bold text in the terminal window. To display bold characters as plain text, clear this option.
- **Enable underline** - Displays text with underline.
- **Inverse video (VT-only)** - This option reverses the foreground and background colors when the VT host sends an inverse video escape sequence. If this option is not enabled, the inverse video sequences sent from the host are ignored.

#### To customize colors for all host types

1. From the left navigation panel, click Display.
2. Under Color Mappings, click the background color field to open the color grid. From the color grid, select the color you want to use as the host background color. Alternatively, type the Hex color number for the color you want to use.

3. From the drop down list, select the default host color you want to change. For example, if you select host pink from the drop down list and then change the foreground color to red, whenever you encounter pink text it will appear as red.
4. Open the color grid for the Foreground to choose a color to map the new color for the text or type the Hex code you want to use. Select Background to map the new color to the background field.
5. Click Save to close the Display panel and resume configuring your host connection.

Restore defaults clears any changes you made and resets the colors to the default host settings.

## 5.2.2 Configure Hotspots

---

Hotspots are buttons that appear over common host commands in terminal sessions. When you use hotspots you can control the terminal session using a mouse or a finger-tap instead of the keyboard. The hotspot transmits a terminal key or command to the host. By default, hotspots are configured for the most common 3270, 5250, and VT commands.

Hotspots are enabled and visible by default, however you can disable hotspots for a particular session or choose to hide them.

- Enable hotspots  
Choose No to disable hotspots for the session you are connecting to.
- Show hotspots  
Choose No to hide hotspots on the screen. Hotspots remain functional.

### Hotspots for 3270 hosts

Hotspot	Description
PF1...PF24	Transmits a PF1...PF24 to the host
PA1, PA2, or PA3	Transmits a PA1, PA2, or PA3 to the host
enter	Transmits an Enter key to the host
more	Transmits a Clear key to the host

### Hotspots for 5250 hosts

Hotspot	Description
enter	Transmits an Enter key to the host
more	Transmits a Roll Up key to the host (scrolls down one page)
PF1...PF24	Transmits a PF1...PF24 to the host

### Hotspots for VT hosts

Hotspot	Description
F1...F20	Transmits a F1...F20 to the host

## 5.2.3 Configure screen dimensions for VT, UTS and T27 hosts

---

As an administrator you can select the number of columns and rows for VT, UTS and T27 sessions.

1. Open the Display panel.
2. Under Dimensions, specify the number of columns and rows you want each screen to possess. The default values are 80 columns by 24 rows.

There are some host-specific settings available:

- Pages - If you are connecting to a T27 host screen, you can set the number of pages to display. The default is 2.
- Clear on host change - If you are connecting to a VT host screen, select this option to clear the terminal window and move the contents to the scrollbar buffer when the column size changes.

3. Click Save.

## 5.2.4 Set Cursor Options

---

Use the cursor options to configure the appearance and behavior of the cursor and ruler.

<b>This option</b>	<b>Does this...</b>
Cursor type	<ul style="list-style-type: none"> <li>• Underline displays the text cursor as an underline.</li> <li>• Vertical bar displays the cursor as a vertical line.</li> <li>• Block displays the text cursor as an inverse video block.</li> </ul>
Ruler type	<ul style="list-style-type: none"> <li>• Vertical displays a vertical ruler at the cursor position.</li> <li>• Horizontal displays a horizontal ruler at the cursor position.</li> <li>• Crosshair displays both a horizontal and vertical ruler at the cursor position.</li> </ul>
Cursor color	Click the color field to open the color grid. From the color grid, select the color you want to use as the color of both the cursor and ruler. Alternatively, type the Hex color number for the color you want to use.
Cursor blinks	By default, the cursor (whether in block or underline mode) blinks. Clear this option to display a visible non-blinking cursor.

## 5.2.5 Set Font Options

---

Use these font options to make sure that your terminal characters display with your preferred font size and style.

<b>This option</b>	<b>Does this...</b>
Font size	<ul style="list-style-type: none"> <li>• Auto (default) - The font scales automatically according to the size of the window. With this option selected, you can choose to Preserve the aspect ratio which means that the font size will be adjusted dynamically but the terminal display will not be stretched or scaled to fill the available space.</li> <li>• Fixed Specify the size, in pixels, for the terminal window display.</li> </ul>
Zero character	<p>To differentiate the default zero character from the letter O, select one of the following options:</p> <ul style="list-style-type: none"> <li>• Default</li> <li>• Zero with a slash</li> <li>• Zero with a dot</li> </ul>

## 5.2.6 Set VT Scrollback Buffer Options

---

The VT scrollback buffer contains the data that has scrolled off the display and is no longer accessible by the host computer. When a scrollback buffer exists, you can view it by using the vertical scroll bar.

The scrollback buffer is enabled by default. When enabled, the session maintains a buffer of lines that have scrolled off the terminal screen. This option is available to all users when they are granted permission to modify Terminal Display Settings by the administrator.

This option	Does this...
Scrollback row limit	Limits the number of rows held in the scrollback buffer. The default setting is 500 rows.
Save display before clearing	When selected (the default), the data on the terminal display moves into the scrollback buffer when you, or the host, clear the terminal display. If you prefer not to have the terminal display saved to the scrollback buffer, clear this option; when the terminal display is cleared, the data is discarded.
Save from scrolling regions	When top and bottom screen margins are set (for example, by a text editor such as EDT or TPU, or with the DECSTBM function) the area within the margins is called the scrolling region. When this option is cleared, scrolling text within this region isn't saved to the scrollback buffer. Select this option to save information within scrolling regions to the scrollback buffer. Note: This can cause display memory to fill quickly.
Save before clearing from any row	This setting specifies whether data that has been cleared from a portion of the terminal window is saved in display memory.
Compress blank rows	Select this option to save room in display memory by compressing multiple blank rows into a single blank row.

## 5.2.7 Set Keyboard Options

---

You can set the following keyboard options:

- [3270 options](#)
- [5250 options](#)
- [VT options](#)
- [T27 options](#)

### 3270 options

- **Typeahead**  
When this option is selected, Host Access for the Cloud buffers the characters that you type in the terminal window. Typeahead allows you to keep typing after you send data to the host. Without typeahead, characters you type are ignored until the host is ready for more data.
- **Word wrap**  
When this option is selected, word wrap functionality is enabled within a multi-line, unprotected field. In word wrap mode, some of the blank spaces between words are replaced by line breaks so that each line is visible in the terminal window and can be read without horizontal scrolling.
- **Attention key sends**  
Specifies what is sent when the ATTN key is pressed. The options are Telnet break, Abort output, and Interrupt process.

**5250 options**

- Typeahead

When this option is selected, Host Access for the Cloud buffers the characters that you type in the terminal window. Typeahead allows you to keep typing after you send data to the host. Without typeahead, characters you type are ignored until the host is ready for more data.

- Error auto reset

When selected, the next key pressed after a keyboard error clears the error, restores the previous error line data, and attempts to execute the keystroke as follows:

If the cursor is in a valid input field and the key is a data key, the data is entered there if it is valid data for that field (for example, a numeric character in an input field that only accepts numbers).

- If the cursor is in a valid input field and the key is a function key, the key operation is executed.
- If the current cursor position is not in a valid input field and the key is a data key, the cursor is moved to the next valid input field and the data is entered there if it is valid data for that field.
- If the current cursor position is not in a valid input field and the key is a function key, the cursor is moved to the next valid input field and the key is ignored.
- If the current screen contains no valid input fields, you'll see an error message with each keystroke you press, and no keystrokes are executed

When cleared, you must press Reset to clear the error message from the error line before you can resume data entry.

By default, this option is not selected.

- Waive field checks for PF key

Select this option to allow PF keys to be sent to the host from restricted fields. This option is cleared by default.



**VT options**

- Backspace sends

Configures the function that the Backspace key sends. On the VT terminal keyboard the back arrow key (<x) is configurable: it can send either a delete (ASCII 127) or a backspace (ASCII 8) character.

- Local echo (VT)

This option causes each character typed at the keyboard to be displayed on the screen. This option is cleared by default, because most hosts echo back received characters.

- Cursor keys

Controls the characters that the four arrow keys (on both the numeric and editing keypad) transmit. This setting is typically set by the host. In general, you should keep this set to Normal.

If the arrow keys aren't working properly, it may mean that this option remained incorrectly set to Application when a host program terminated abnormally. Changing this setting back to Normal should fix the problem with the arrow keys.

- Keypad

Controls the characters that the numeric keypad keys transmit. This setting is typically set by the host. In general, you should keep this set to Numeric.

If the number or PF keys aren't working properly, it may mean that this option was incorrectly left set to Application when a host program terminated abnormally. Changing this setting back to Numeric should fix the numeric keypad.

**T27 options**

- Enable lower case (T27)

Enables lower case, as well as upper case letters to be displayed on the screen. Default. If this is option is disabled only upper case letters will display.

**5.2.8 Terminal Settings**

---

Terminal settings vary depending on your host type.

**3270 and 5250 terminal settings**

- Host character set

Select the 3270 or 5250 host character set you want to use. This setting chooses a conversion table to convert host characters (EBCDIC) into PC characters (ANSI). This setting should match the national character set used by your host system. If it doesn't match, then some characters, such as accents, may not display correctly. See your host documentation for definitions of the characters in each set. The default value is US English (037).

- Country extended graphics code (3270 only)

With this option selected (the default), additional characters are available in the configured National character set. See your host documentation for details.

**VT terminal settings**

- Terminal type (VT)

Specifies which terminal should be emulated. These choices affect the codes generated by the numeric keypad, the interpretation of control functions, and the response to terminal identification requests.

- Terminal ID (VT)

Specifies the response that Host Access for the Cloud sends to the host after a primary device attributes (DA) request. This response lets the host know what terminal functions it can perform. This setting is independent of the terminal type setting. When set to the default value of Reflection, Host Access for the Cloud responds to a primary DA request with the set of features it supports. If your host requires a more specific terminal ID, select another value from the list.

- New line (VT)

Select this option to send both a carriage return and linefeed when you press Enter. When Host Access for the Cloud receives a linefeed, form feed, or vertical tab, it moves the cursor to the first column of the next line. When this option is cleared (default), the Enter key sends only a carriage return. A linefeed, form feed, or vertical tab received from the host moves the cursor down one line in the current column. If lines on the display keep getting overwritten (that is, the host is not sending a linefeed along with a carriage return), select this option. If the New line option is selected but the host does not expect to receive a linefeed with each carriage return, lines will be double-spaced on the display.

### **T27 terminal settings**

- Host character set (T27)

Using this option you can specify host to screen translation. Select the language used to convert characters received from the host before they are displayed on the local machine. The default is No translation.

## 5.2.9 Set Other Display Options

---

This option	Does this...
Column separator style (5250)	Use this option to specify which character (if any) should be used to render column separators in 5250 terminal sessions. The options are: <ul style="list-style-type: none"> <li>• Dots- Dots are used to separate columns. The default.</li> <li>• Vertical bars - Vertical lines are used to separate columns.</li> <li>• None - No characters are used to separate columns</li> </ul>
Input field underlining (3270, 5250)	You can determine how the underlining of host input fields is handled: <ul style="list-style-type: none"> <li>• Host controls underlining (Default)</li> <li>• Always underline input fields</li> <li>• Never underline input fields</li> </ul>
Status line (VT)	To enable a status line at the bottom of the display. <ul style="list-style-type: none"> <li>• None to disable the status line. (Default)</li> <li>• Indicator to display the page, cursor position, and printer status.</li> <li>• Host Writable to have the host application display information in the status line.</li> </ul>
Preserve aspect ratio	Select this option to maintain the host screen aspect ratio regardless of the size of the browser window. Aspect ratio describes the proportional relationship between the width of an image and its height.
Display OIA (3270, 5250)	Select this option to display the operation and status messages in the Operator Information Area (OIA) at the bottom of the terminal window . By default, OIA display is enabled.
Display status line (ALC)	Turns on a status line at the bottom of the display.
Ignore mouse click on window activation	When a mouse click activates the terminal window, this option specifies whether actions such as updating the terminal cursor position, clearing a selection, or executing a hotspot are also performed. By default, these actions are not performed.
Auto wrap (VT)	When selected, characters automatically wrap at the right margin and continue on the next line. When cleared, characters do not wrap when they reach the right margin of the display. New characters overwrite the character at the right margin until a carriage return is entered.

## 5.3 Map Keys

---

### 5.3.1 Map Keys

---

You can create keyboard shortcuts that perform any assignable action during a session. The Key Mappings settings page provides a view of the default keyboard map for each host type and the mapped custom keys, indicated in boldface type, for that session.

See [Host Keyboard Mapping](#) for the different host keyboard mappings.



#### **Mapping keys as an administrator and as an end-user**

There are a few differences in behavior between the administrator and the end-user when mapping keys.

- End users can only add or modify key mappings if they are granted permission by the administrator using the User Preference Rules panel.
- Any changes the administrator makes show up to the end user as indistinguishable from default host key mappings. Once granted permission, the person can modify, add or delete any mappings regardless of any administrator changes. However restoring key mappings only restores them back to the modified state created by the administrator for the current session.

### Adding or modifying mapped keys

1. From the toolbar, click Settings.
2. From the left navigation pane, open the Key Mappings panel. The mapped keys for the host type you are connecting to are visible.
3. To add a new key mapping:

- Click . You can choose to type the key sequence you want to use or use the keyboard by toggling  between the two options.
- From the Action drop down list, select the action you want to associate with the key selection. If you select Send text, enter the string you want sent to the host in the Value field. Likewise, if you select Run Macro, choose the macro you want triggered by the keyboard shortcut. You must create the macro before you can map it to the Run Macro action.

The Send text action supports mapping characters with codes less than or equal to 0xFFFF via Unicode escape sequences. The escape sequence begins with `\u` followed by exactly four hexadecimal digits. You can embed Unicode escape sequences in any string. For example, this embedded `\u0045` will be interpreted as this embedded E, since 45 is the hexadecimal code for the character E.

To pass Unicode escape sequences to the host, escape the sequence with a leading backslash. For example, to send the string literal `\u001C` to the host, map a key to `\\u001C`. Host Access for the Cloud will convert this to the string `\u001C` when that key is pressed and send the 6 characters of the resulting string to the host.



The Disable action renders the key inoperable. When pressed the key will not initiate any action. This differs from the Unmap action which removes the key mapping but preserves a browser short-cut if it is defined.

- Click the blue check mark to accept the mapping and add the key map to the session.

4. To modify an existing mapping:

Select the row containing the key you want to modify.



Follow the steps for adding a new key mapping, clicking  to save the new mapping. Alternatively, you can click away from the modified row and the change will be saved. All new and modified key maps are indicated in boldface type. You can restore the original key mapping at any time by clicking .

### Filtering the list

The Filter field makes it easy to see just those mappings you are interested in. The filter is based on keywords and affects each column of the table. For example, if you enter Send text in the Filter field, only keys mapped to the Send text action are displayed.

Using the Show only modified mappings option lets you see only those mappings that have been previously modified.

**Some things to remember:**

- Mapping right and left modifier keys to individual actions

You can map the right and left modifier keys to individual actions. However when they are combined with other keys, there is no differentiation between the right and left keys. For example, Left-Alt can be mapped to Action-A while Right-Alt is mapped to Action-B, but Left-Alt + H will be stored as Alt+H and both Left-Alt+H and Right-Alt+H will be associated with a single mapped action.

- Key stroke combinations and copy/paste operations

Different key stroke combinations are also used for copy/paste operations. For example, on a VT host screen, Ctrl+ Shift + A initiates a Select All action. See Editing the Screen for a list of copy/paste key actions.

- Keyboard shortcuts and browsers

Browsers use keyboard shortcuts to save both time and mouse clicks. When mapping keystrokes it is important to keep this in mind. Handy Keyboard Shortcuts gives a brief overview of the keyboard shortcuts used by different browsers. In most cases Host Access for the Cloud key mappings take precedence over browser key shortcuts. Occasionally, where this is not the behavior you want for a particular key combination, you can choose Unmap in the action list to unmap the short-cut. This lets the key event to pass through to the browser.

## 5.3.2 Host Keyboard Mapping

---

The following tables provide the default keys, key name, and key description for the different host keyboard mappings.

- [IBM 3270 Keyboard Mapping](#)
- [IBM 5250 Keyboard Mapping](#)
- [VT Keyboard Mapping](#)
- [UTS Keyboard Mapping](#)
- [T27 Keyboard Mapping](#)
- [ALC Keyboard Mapping](#)

**IBM 3270 keyboard mapping**



<b>Key</b>	<b>Maps to</b>	<b>Description</b>
Ctrl + F1	Attention	Sends the ATTENTION key to the host
Shift + Tab	Backtab	Moves the cursor to the previous unprotected field
Ctrl + F2	Clear	Clears the screen and sends the CLEAR key to the host
Alt + ArrowLeft	Cursor left double	Moves the cursor two positions to the left
Alt + ArrowRight	Cursor right double	Moves the cursor two positions to the right
Ctrl + F3	Cursor select	Simulates a lightpen select in the current field
Alt + Delete	Delete word	Deletes three characters from the current field
Ctrl + 5	Duplicate	Inserts the DUP character at the cursor location
Enter	Enter	Sends the ENTER key to the host
End	Erase end of field	Erases all data from the cursor location to the end of the current field
Alt + F5	Erase input	Erases all data in all unprotected fields of the current screen
Ctrl + Alt + F	Field delimiter	Toggles whether field delimiters are displayed on screen
Ctrl + 6	Field mark	Inserts the Field Mark character at the cursor location
Home	Home	Moves the cursor to the first unprotected field on the screen
Insert	Insert	Toggles Insert mode
Shift + Enter	New line	Moves to the next unprotected field
Ctrl + 1	PA1	Sends the PA1 key to the host
Pageup	PA1	Sends the PA1 key to the host
Ctrl + 2	PA2	Sends the PA2 key to the host
Pagedown	PA2	Sends the PA2 key to the host
Ctrl + 3	PA3	Sends the PA3 key to the host
F1 - F10	PF1 - PF10	Sends the PF1, PF2...PF10 key to the host
Alt + 1 or F11	PF11	Sends the PF11 key to the host
Alt + 2 or F12	PF12	Sends the PF12 key to the host
Shift + F1	PF13	Sends the PF13 key to the host
Shift + F2	PF14	Sends the PF14 key to the host
Shift + F3	PF15	Sends the PF15 key to the host
Shift + F4	PF16	Sends the PF16 key to the host
Shift + F5	PF17	Sends the PF17 key to the host
Shift + F6	PF18	Sends the PF18 key to the host
Shift + F7	PF19	Sends the PF19 key to the host
Shift + F8	PF20	Sends the PF20 key to the host
Shift + F9	PF21	Sends the PF21 key to the host

<b>Key</b>	<b>Maps to</b>	<b>Description</b>
Shift + F10	PF22	Sends the PF22 key to the host
Alt3	PF23	Sends the PF23 key to the host
Shift + F11	PF23	Sends the PF23 key to the host
Alt4	PF24	Sends the PF24 key to the host
Shift + F12	PF24	Sends the PF24 key to the host
Ctrl +P	Print	Prints the contents of the screen to the printer
Escape	Reset	Resets keyboard error conditions
Ctrl + S	System request	Sends the SYSTEM REQUEST key to the host

**IBM 5250 keyboard mapping**

<b>Key</b>	<b>Maps to</b>	<b>Description</b>
Escape	Attention	Sends the ATTENTION key to the host
Ctrl + F2	Clear	Clears the screen and send the CLEAR key to the host
Ctrl + F3	Cursor select	Simulates a lightpen select in the current field
Ctrl + Backspace	Destructive backspace	Moves the cursor one position to the left
Ctrl + 5	Duplicate	Inserts the DUP character at the cursor location
Ctrl + End	End of field	Moves the cursor to the end of the field
End	Erase end of field	Erases all data from the cursor location to the end of the current field
Alt + End	Erase input	Erases all data in all unprotected fields of the current screen
Alt + F5	Erase input	Erases all data in all unprotected fields of the current screen
Ctrl + Enter	Field exit	Moves the cursor out of an input field
KP + Subtract	Field exit minus	Moves the cursor out of a signed-numeric or numeric-only field, inserting a minus sign in the last position of a signed-numeric field, or changing the last position in a numeric-only field to an alphabetic character that tells the system that this field has a negative value.
Ctrl + Subtract	Field exit minus	Moves the cursor out of a signed-numeric or numeric-only field, inserting a minus sign in the last position of a signed-numeric field, or changing the last position in a numeric-only field to an alphabetic character that tells the system that this field has a negative value.
KP + Add	Field exit plus	In a signed-numeric field, this function moves the cursor to the next field, removing a minus sign if there is one in the last position. In a numeric-only field, this function moves the cursor to the next field, changing the last position to an alphabetic character that tells the system that this field has a positive value.
Ctrl + Add	Field exit plus	In a signed-numeric field, this function moves the cursor to the next field, removing a minus sign if there is one in the last position. In a numeric-only field, this function moves the cursor to the next field, changing the last position to an alphabetic character that tells the system that this field has a positive value.
Ctrl + 6	Field mark	Inserts the field mark character at the cursor location
Ctrl + H	Help	Sends the Help key to the host
Alt + F7	Hex mode	Places the terminal in Hex mode
Home	Home	Moves the cursor to the first unprotected field on the screen
Insert	Insert	Toggles Insert mode
Shift + Enter	New line	Moves to the next unprotected field
Ctrl + 1	PA1	Sends the PA1 key to the host
Ctrl + 2	PA2	Sends the PA2 key to the host

<b>Key</b>	<b>Maps to</b>	<b>Description</b>
Ctrl + 3	PA3	Sends the PA3 key to the host
F1 - F11	PF1 - PF11	Sends the PF1, PF2...PF11 key to the host
Alt + 1	PF11	Sends the PF11 key to the host
Alt + 2	PF12	Sends the PF12 key to the host
F12	PF12	Sends the PF12 key to the host
Shift + 1	PF13	Sends the PF13 key to the host
Shift + F2...F10	PF14...PF22	Sends the PF14...PF22 key to the host
Alt + 3	PF23	Sends the PF23 key to the host
Shift + F11	PF23	Sends the PF23 key to the host
Alt + 4	PF24	Sends the PF24 key to the host
Shift + F12	PF24	Sends the PF24 key to the host
Ctrl + P	Print	Prints the contents of the screen to the printer
Control	Reset	Resets the keyboard error conditions
Pageup	RollDown	Sends the RollDown key to the host
Pagedown	RollUp	Sends the RollUp key to the host
Ctrl + Home	Start of field	Moves the cursor to the start of the field
Ctrl + S	System request	Sends the SYSTEM REQUEST key to the host

**VT keyboard mapping**

<b>Key</b>	<b>Maps to</b>	<b>Description</b>
Ctrl + Cancel	Break	Sends the Break key to the host
Ctrl + Enter	Enter	Sends the Enter key to the host
Alt + F1	F1	Sends the F1 key to the host
Ctrl + F1...F10	F11...F20	Sends the F11...F20 key to the host
Home	Find	Sends the Find key to the host
F1	Hold	Sends the Hold Screen to the host
Pause	Hold	Sends the Hold Screen to the host
Insert	Insert	Sends the Insert key to the host
Ctrl + Insert	Keypad 0	Sends the numeric keypad 0 key to the host
Ctrl + End	Keypad 1	Sends the numeric keypad 1 key to the host
Ctrl + ArrowDown	Keypad 2	Sends the numeric keypad 2 key to the host
Ctrl + Pagedown	Keypad 3	Sends the numeric keypad 3 key to the host
Ctrl + ArrowLeft	Keypad 4	Sends the numeric keypad 4 key to the host
Ctrl + Clear	Keypad 5	Sends the numeric keypad 5 key to the host
Ctrl + ArrowRight	Keypad 6	Sends the numeric keypad 6 key to the host
Ctrl + Home	Keypad 7	Sends the numeric keypad 7 key to the host
Ctrl + ArrowUp	Keypad 8	Sends the numeric keypad 8 key to the host
Ctrl + Pageup	Keypad 9	Sends the numeric keypad 9 key to the host
Ctrl + Alt-add	Keypad comma	Sends the numeric keypad Comma key to the host
Ctrl + add	Keypad minus	Sends the numeric keypad Minus key to the host
Ctrl + decimal	Keypad period	Sends the numeric keypad Period key to the host
Ctrl + Delete	Keypad period	Sends the numeric keypad Period key to the host
Ctrl + Alt + ArrowUp	Row up	In the scrollbar buffer moves up a row
Ctrl + Alt + ArrowDown	Row down	In the scrollbar buffer moves down a row
Pagedown	Next	Sends the Next Screen key to the host
Ctrl + Pause	PF1	Sends the PF1 key to the host
Ctrl + Divide	PF2	Sends the PF2 key to the host
Ctrl + Multiply	PF3	Sends the PF3 key to the host
Ctrl + Subtract	PF4	Sends the PF4 key to the host
Pageup	Previous	Sends the Prev Screen key to the host
Delete	Remove	Sends the Remove key to the host
End	Select	Sends the Select key to the host
Shift + F6...F10	UDK 6...10	Sends the User Defined Key 6...10 to the host
Shift + Ctrl + F1...F10	UDK11...20	Sends the User Defined Key 11...20 to the host

## UTS keyboard mapping

<b>Key</b>	<b>Maps to</b>	<b>Description</b>
F4	Clear Change Bit	Sends the CLEARCHANGEBIT key to the host
Keypad+Enter	Carriage Return	Sends a carriage return to the host
Ctrl + PageDown	Clear End of Display	Clears text from the cursor location to the end of the display
Ctrl+PageUp	Clear End of Display FCC	Clears all data (including FCC information) from the cursor to the end of the display
Ctrl+End	Clear End of Field	Clears text from the cursor location to the end of the field
Ctrl+Shift+end	Clear End of Line	Clears text from the cursor location to the end of the row
F7	Clear FCC	Clears the field control character
Ctrl+Home	Clear Home	Sends the CLEAR_HOME key to the host
Ctrl+H	Column Separator Right	Sends the COLUMN_SEP_RIGHT key to the host
Ctrl+F1	Control Page	Sends the CONTROL_PAGE key to the host
Keypad+2	Cursor Down	Moves the cursor one row down
Keypad+4	Cursor Left	Moves the cursor one column to the left
Keypad+6	Cursor Right	Moves the cursor one column to the right
Keypad+8	Cursor Up	Moves the cursor one row up
Delete	Delete in Line	Sends the DELETE_IN_LINE key to the host
Ctrl+Delete	Delete in Page	Sends the DELETE_IN_PAGE key to the host
Ctrl+Shift+Delete	Delete Line	Deletes the row at the cursor location
Ctrl+ArrowDown	Duplicate Line	Duplicates the row at the cursor location
F8	Enable FCC	Enables the field control character
Keypad+-	End of Display and Transmit	Sends the EOD_AND_TRANSMIT key to the host
Shift+End	End of Field	Moves the cursor to the end of the field
End	End of Line	Moves the cursor to the end of the row
Ctrl+ArrowRight	End of Page	Moves the cursor to the end of the page
Shift+Space	Erase Character	Erases the character at the cursor location
Ctrl+Shift+E	Euro Character	Sends the Euro character to the host
Ctrl+1...Ctrl+9	F1...F9	Sends the F1...F9 key to the host
Ctrl+0	F10	Sends the F10 key to the host
Ctrl+-	F11	Sends the F11 key to the host
Ctrl+=	F12	Sends the F12 key to the host
Ctrl+Q	F13	Sends the F13 key to the host
Ctrl+W	F14	Sends the F14 key to the host
Ctrl+E	F15	Sends the F15 key to the host
Ctrl+R	F16	Sends the F16 key to the host



Key	Maps to	Description
Ctrl+T	F17	Sends the F17 key to the host
Ctrl+Y	F18	Sends the F18 key to the host
Ctrl+U	F19	Sends the F19 key to the host
Ctrl+l	F20	Sends the F20 key to the host
Ctrl+0	F21	Sends the F21 key to the host
Ctrl+P	F22	Sends the F22 key to the host
Shift+F3	FF	Sends a formfeed to the host
F9	Generate FCC	Generates a field control character
Home	Home	Moves the cursor to the first field in the display
Ctrl+Shift+Space	Insert in Line	Sends the INSERT_IN_LINE key to the host
ICtrl+Space	Insert in Page	Sends the INSERT_IN_PAGE key to the host
Ctrl+Shift+Insert	Insert Line	Inserts a new row into display memory
Insert	Insert Mode	Toggles insert character mode
F5	Locate FCC	Disables the field control characters and moves to the first character of the next field to the right of the cursor
F3	Message Wait	Sends the MESSAGE_WAIT key to the host
Shift+F2	New Line	Moves the cursor to a new row
Keypad+Shift+2	Next Field	Moves the cursor to the next field
Keypad+Shift+4	Next Field	Moves the cursor to the next field
PageDown	Page Down	Sends the Page Down key to the host
PageUp	Page Up	Sends the Page Up key to the host
Keypad+Shift+6	Previous Field	Moves the cursor to the previous field
Keypad+Shift+8	Previous Field	Moves the cursor to the previous field
Clear	SOE Character	Sends the SOE character to the host
F12	SOE Character	Sends the SOE character to the host
Ctrl+Clear	Set Tab	Sends the SET_TAB key to the host
Ctrl+Tab	Set Tab	Sends the SET_TAB key to the host
Shift+Home	Start of Field	Moves the cursor to the start of the field
Ctrl+ArrowLeft	Start of Line	Moves the cursor to the start of the row
Ctrl+[	System Mode	Sends the SYSTEM_MODE key to the host
Ctrl+J	Toggle Column Separator	Toggles the column separator
Ctrl+F12	Toggle Message Wait Beep	Sends the TOGGLEMSGWAITBEEP key to the host
Ctrl+L	Toggle Strike Thru	Toggles strike thru mode
Ctrl+K	Toggle Underline	Toggles underline mode

<b>Key</b>	<b>Maps to</b>	<b>Description</b>
Ctrl+Enter	Transmit	Transmits the contents of the display to the host
ScrollLock	Transmit	Transmits the contents of the display to the host
Keypad++	Transmit	Transmits the contents of the display to the host
Keypad+Ctrl+	Transmit	Transmits the contents of the display to the host
Escape	Unlock	Sends the UNLOCK key to the host
Ctrl+] ]	Workstation Mode	Sends the WORKSTATION_MODE key to the host

## T27 keyboard mapping

<b>Key</b>	<b>Maps to</b>	<b>Description</b>
Backspace	Backspace	Moves the cursor one column to the left
Shift+tab	back tab	Moves the cursor to the previous field
Ctrl+Delete	Clear End of Line	Clears text from the cursor location to the end of the row
Shift+Home	Clear Page Home	Clears the page and homes the cursor
Left Ctrl	Control Page	Puts the session in control mode
Down arrow	Cursor Down	Moves the cursor one row down
Left arrow	Cursor Left	Moves the cursor one column to the left
Right arrow	Cursor Right	Moves the cursor one column to the right
Up arrow	Cursor Up	Moves the cursor one row up
Ctrl+left arrow	Cursor Word Left	Moves the cursor to the previous word
Ctrl+right arrow	Cursor Word Right	Moves the cursor to the next word
Ctrl+D	Delete Line	Deletes the row at the cursor location
Ctrl+End	End of Line	Moves the cursor to the end of the row
End	End of Page	Moves the cursor to the last field on the page
Shift+Ctrl+E	Euro Character	Sends a Euro character to the host
Home	Home	Moves the cursor to the first field in the display
Insert	Insert Mode	Puts the session in insert mode
Ctrl+l	Insert Line	Inserts a new row into display memory
Ctrl+1	PF1	Sends the PF1 key to the host
Ctrl+10	PF10	Sends the PF10 key to the host
Ctrl+2	PF2	Sends the PF2 key to the host
Ctrl+3	PF3	Sends the PF3 key to the host
Ctrl+4	PF4	Sends the PF4 key to the host
Ctrl+5	PF5	Sends the PF5 key to the host
Ctrl+6	PF6	Sends the PF6 key to the host
Ctrl+7	PF7	Sends the PF7 key to the host
Ctrl+8	PF8	Sends the PF8 key to the host
Ctrl+9	PF9	Sends the PF9 key to the host
PageDown	Page Down	Displays the next page
PageUp	Page Up	Displays the previous page
Ctrl+E	Put ETX	Inserts an end-of-text character and homes the cursor
Keypad /	Put Local	Puts the session in local mode
Keypad *	Put Receive	Puts the session into receive mode

<b>Key</b>	<b>Maps to</b>	<b>Description</b>
Enter	Return	Sends the return key to the host
Keypad Enter	Return	Sends the return key to the host
Ctrl+A	Select All	Selects all text
Shift+down arrow	Select Down	Selects text down
Shift+left arrow	Select Left	Selects text left
Shift+right arrow	Select Right Selects text right	
Shift+up arrow	Select Up	Selects text up
Shift+Ctrl+1	Shift F1	Sends the Shift F1 key to the host
Shift+Ctrl+0	Shift F10	Sends the Shift F10 key to the host
Shift+Ctrl+2	Shift F2	Sends the Shift F2 key to the host
Shift+Ctrl+3	Shift F3	Sends the Shift F3 key to the host
Shift+Ctrl+4	Shift F4	Sends the Shift F4 key to the host
Shift+Ctrl+5	Shift F5	Sends the Shift F5 key to the host
Shift+Ctrl+6	Shift F6	Sends the Shift F6 key to the host
Shift+Ctrl+7	Shift F7	Sends the Shift F7 key to the host
Shift+Ctrl+8	Shift F8	Sends the Shift F8 key to the host
Shift+Ctrl+9	Shift F9	Sends the Shift F9 key to the host
F5	Specify	Transmits the cursor location to the host
Tab	Tab	Moves the cursor to the next field
F2	Transmit	Transmits the page to the host
Keypad +	Transmit	Transmits the page to the host
Ctrl+F2	Transmit Line	Transmits the current row to the host
Keypad -	Transmit Line	Transmits the current row to the host

**ALC keyboard mapping**

Key	Maps to	Description
Ctrl+M	Auto Move Down	Toggles the session ability to receive multiple pages
Backspace	Backspace	Moves the cursor one column to the left
Shift+tab	back tab	Moves the cursor to the previous field
Ctrl+Home	Clear	Clears the screen and sends the CLEAR key to the host
Ctrl+B	Clear Broadcast	Clears the SITA broadcast message
:	Colon	Inserts a colon character at the cursor position
Ctrl+L	Cross of Lorraine	Inserts the Cross of Lorraine character at the cursor position
↓	Cursor Down	Moves the cursor down a row
Keypad ↓	Cursor Down	Moves the cursor down a row
←	Cursor Left	Moves the cursor to the previous word
Keypad ←	Cursor Left	Moves the cursor to the previous word
→	Cursor Right	Moves the cursor to the next word
Keypad →	Cursor Right	Moves the cursor to the next word
↑	Cursor Up	Moves the cursor up a row
Keypad ↑	Cursor Up	Moves the cursor up a row
Delete	Delete character	Deletes the character at the cursor location
Ctrl+Delete	Delete Line	Deletes the line at the cursor position
=	Display	Inserts the display character at the cursor position
Ctrl+N	Display New Line	Inserts the display character at a new line
]	Dollar	Inserts the U.S. dollar sign character at the cursor position
.	End Item	Inserts the end item character at the cursor position
End	End of Line	Moves the cursor to the end of the line
Ctrl+T	End Transaction	Closes the PNR
Ctrl+E	Erase End of Display	Erases all data from the cursor position to the end of display
Ctrl+End	Erase End of Line	Erases all data from the cursor position to the end of line
Home	Home	Moves the cursor to the first unprotected field on the screen
Ctrl+I	Ignore	Cancel any changes made to the current PNR
Ctrl+Insert	Insert Line	Inserts a new line into display memory
Insert	Insert Space	Inserts a space into display memory
\	New Line	Inserts the newline character at the cursor position
[	Pillow	Inserts the pillow character at the cursor position
Ctrl+G	Pound	Inserts a British pound mark at the cursor position
Ctrl+Enter	Print Enter	Sends the response to the printer

<b>Key</b>	<b>Maps to</b>	<b>Description</b>
Ctrl+P	Protected Reset	Moves the cursor to the first unprotected field
Ctrl+↑	Recall Next Input	Recalls the next input or entry
Ctrl+↓	Recall Previous Input	Recalls the previous input or entry
Ctrl+Z	Reenter	Resends the previously sent message to the host
Ctrl+R	Repeat	Redisplays the last message sent by the host
Escape	Reset	Resets keyboard error conditions
Shift+Ctrl+↓	Scroll Line Down	Scrolls the display down one line
Shift+Ctrl+↑	Scroll Line Up	Scrolls the display up one line
PageDown	Scroll Page Down	Scrolls the display down one page
PageUp	Scroll Page Up	Scrolls the display up one page
Ctrl+A	Select All	Selects all text
Shift+↓	Select Down	Selects all text down
Shift+↑	Select Up	Selects all text up
Shift+←	Select Left	Selects all text left
Shift+→	Select Right	Selects all text right
'	Start of Message	Inserts a start-of-message character at the cursor position
F12	Statistics	Displays communication statistics
Tab	Tab	Moves the cursor to the next unprotected field
Ctrl+F	Toggle CODACOM	Toggles CODACOM mode
Enter	Transmit	Transmits page to the host
Keypad Enter	Transmit	Transmits page to the host
Shift+Enter	Transmit	Transmits page to the host
Shift+Escape	Unlock Keyboard	Unlocks the keyboard
Ctrl+U	Unsolicited Message	Retrieves an unsolicited message from the host



## 5.4 Transfer Files

---

Host Access for the Cloud supports three different file transfer protocols:

- IND\$FILE for 3270 host transfers
- AS/400 for 5250 host transfers
- File Transfer Protocol (FTP), which allows a local computer to act as an FTP client.

Once connected, you can view files on the server and use FTP to transfer files between your local computer (or any networked drive) and the FTP server.

Batch file transfer is available for FTP transfers. Using this option you can download and upload multiple files in one operation.

Before you can transfer or send files, the administrator must enable the transfer, send options for the current session, and make the necessary configurations. This is done on the File Transfer settings panel.

Depending on the host file system and transfer method you want to use, you will see different configuration options. Once configured, the file transfer dialog box is available from the tool bar.

- [IND\\$FILE](#)
- [AS/400](#)
- [FTP](#)
- [Batch transfers](#)

### 5.4.1 IND\$FILE

---

IND\$FILE is a file transfer program from IBM which you can use to transfer information between your computer and a 3270 host computer.

From the Host file system drop down list, select which IBM 3270 operating environment the host is running. Host Access for the Cloud supports TSO (Time Sharing Option), CMS (Conversational Monitor System) and CICS. The default selection is None.

There is support for ASCII or binary transfers and, if you connected to a TSO host, you can navigate directly to a particular TSO dataset.

#### **General options for CICS, CMS, and TSO host file types**

Automatically show host files - By default, the host file list contains all the host files that are available to transfer. To retrieve host files only when you request them, disable this option. On the Transfer dialog box, click Show host files to retrieve the host files.

## Transfer options for CICS, CMS, and TSO host file types

Option	Description
Transfer method	<ul style="list-style-type: none"> <li>• Binary - Use for program files and other types of files that should not be translated, such as files that have already been formatted for a particular type of printer or files with application-specific formatting. Binary files contain non-printable characters; using this method, a file is not converted or translated during the transfer.</li> <li>• ASCII - Use to transfer text files with no special formatting. ASCII files on the PC are translated to the EBCDIC character set on the host and host text files are converted from EBCDIC to ASCII when they are downloaded.</li> </ul>
CR/LF processing	If this option is selected, carriage return - line feed pairs will be stripped from files sent to the host and added to the end of each line on files received from the host.
Startup command	Specifies the host program used to initiate the file transfer. IND\$File, the default, is appropriate for CMS and TSO hosts. For CICS hosts, IND\$File may be appropriate, or you may need to specify your site's CICS transaction (for example, CFTR).
Startup parameters	Use this field for any parameters specific to the IND\$File program on your host system. The contents of this field are appended to the end of the transfer command generated by Host Access for the Cloud. Host Access for the Cloud does not validate the parameters.
Max field size	Select a field size to use with the Write Structured Field protocol. The default value is 4 kilobytes. Typically, the larger the buffer size, the faster the transfer. Most systems support 8K; if you choose a value that is too large for your host, it will disconnect your session when you first attempt to send a file big enough to fill the buffer. The person who installs the host communication software usually supplies this value. For example, IBM's host TCP/IP product gets this value from the DATABUFFERPOOLSIZE parameter, which defaults to 8K buffers. See your system administrator if you don't know what to enter here.
Lead key	You can specify certain actions before transferring or listing files. Your choices are None, Auto Sense, and Clear. If set to None, LISTCAT is issued automatically. If set to Auto Sense, the current screen contents are examined to determine if a LISTCAT or TSO LISTCAT should be sent. If set to Clear, the Clear key is sent before issuing command. For TSO, Clear also means "TSO" will not be prepended to the request files command.
PC code page	The character set to use when reading or writing local files during a file transfer. The value <b>Default</b> uses the code page corresponding to your operating systems locale. If you need a different character set to specify the PC code page, select it from the list.
Host code page	The character set to use when translating EBCDIC characters while transferring files to or from the host. The default, <b>Use NCS setting</b> , uses the national character set specified on the Display panel under Terminal. If you need a different character set to specify the host code page, select it from the list.
Response timeout (seconds)	Specifies how many seconds Host Access for the Cloud should wait for a host response before timing out and returning an error. The default value is 60 seconds.
Startup timeout (seconds)	Specifies the number of seconds Host Access for the Cloud should wait for a host response when attempting to connect to a host. If the specified amount of time elapses with no response from the host, Host Access for the Cloud times out and returns an error. The default value is 25 seconds.

## Send options for CICS, CMS, and TSO host file types

Option	Description	Applies to this host type
Record format	<p>Use this option to specify the record format for files sent to the host.</p> <ul style="list-style-type: none"> <li>• Default - The host determines the record format. This is the default option.</li> <li>• Fixed - Forces the host to create fixed-length records.</li> <li>• Undefined - Forces the host to create files without a specific record format (this value is only relevant for TSO systems).</li> <li>• Variable - Forces the host to create variable-length records and preserves the format of a binary file.</li> </ul>	TSO, CMS
Allocation units	Specifies the disk subdivisions for your primary and secondary space allocations. If you select Default (default), the unit is determined by the host. You can also select Cylinder, Track, or Block. If you select Block, use the Average block box to specify the size for an average block (in bytes).	TSO
Logical record length	The record size (in bytes) for the file being created on the host. If you leave this box blank, the record size is determined by the host. You can set any value between 0 and 32767 to accommodate any range accepted by your host. This option is not available on CICS hosts. For ASCII files, set this value to accommodate the longest line in your file. When you leave this box blank, the host usually accepts lines of up to 80 characters.	TSO, CMS
If host file exists	<p>Specifies how the transfer should operate if a file with the same name already exists.</p> <ul style="list-style-type: none"> <li>• Append - Append the contents of the local file to the existing host file.</li> <li>• Overwrite - Overwrite the contents of the host file</li> </ul> <p>With CICS systems there is no way to tell if a host file already exists, so Overwrite is the only available option for sending files to a CICS system.</p>	TSO, CMS
Block size (bytes)	On TSO hosts, specifies the block size for the file being created on the host. For files with fixed-length records, this value must be a multiple of the Logical record length (because blocks are divided into logical records). You can set any value between 0 and 32767, to accommodate any range accepted by your host.	TSO
Average block (bytes)	The size for an average block. This value is only relevant if you are using blocks as your allocation unit.	TSO
Primary allocation (allocation units)	The size of the primary allocation for the host file being created.	TSO
Secondary allocation (allocation units)	The size of any additional allocations in the event that the primary allocation is not sufficient. Multiple secondary allocations (known as "extents") are allowed, up to a host-specified limit (generally 15).	TSO

 **Note**

When using CICS as the host system you must enter the names of the files you are transferring manually. A list of files to choose from is not available.

### Transferring files (IND\$FILE)

You must be connected and logged into the host to transfer files for the current 3270 session.

1. Verify that the host is in a 'ready' state to accept the IND\$FILE command.

2.



From the tool bar, click the IND\$File icon .

3. The File Transfer dialog box displays a list of host files and directories that are available to transfer. Directories and files are indicated by an icon when you select the file. For CICS hosts, type in the names of the files you want to transfer.

4. Select the transfer method. The options are:

- Binary

Use for program files and other types of files that should not be translated, such as files that have already been formatted for a particular type of printer or files with application-specific formatting. Binary files contain non-printable characters; using this method, a file is not converted or translated during the transfer.

- ASCII

Use to transfer text files with no special formatting. ASCII files on the PC are translated to the EBCDIC character set on the host and host text files are converted from EBCDIC to ASCII when they are downloaded.

5. If you are connected to a TSO host, click Level to type in the new dataset you want to view. Host Access for the Cloud updates the remote file list using the dataset level you specify.

 **Note**

When specifying files using *UploadAs* or *Download*, a fully qualified data set name needs to be enclosed in single quotes. Data set names not enclosed in single quotes will, by default, be prefixed with a high level qualifier specified in the TSO PROFILE.

You can refresh the file list at any time by clicking the Refresh icon in the upper left corner of the File Transfer dialog box.

- 
- [Downloading files \(IND\\$FILE\)](#)
  - [Uploading files \(IND\\$FILE\)](#)
  - [Troubleshooting your file transfers](#)

### Downloading files (IND\$FILE)

1. From the list, click the name of the file to initiate the transfer.

or

Click **Download** and enter the name of the host file you want to transfer. You can download from both TSO and CMS host types. However, TSO and CMS represent host files differently; this means the format of the file name you enter into the message prompt will vary.

- **TSO** - Wrap the name of the host path in single quotes to specify the complete dataset name. For example, 'BVTST03.DATA.TXT'. To specify a file location relative to the dataset level you set above, omit the single quotes. For example, DATA.TXT, which identifies the same dataset but relative to BVTST03.
- **CMS** - A typical CMS input would be BVTST01 DATA A1. Single quotes are not needed.

2. If necessary, you can cancel the transfer from the transfer progress panel.

### Uploading files (IND\$FILE)

#### Note

IBM mainframe computer systems impose certain naming conventions for files. For detailed information on naming requirements, see the [IBM documentation](#).

Choose either method for uploading files:

- From the **File Transfer** dialog box, click **Upload**.  
You can specify a different name for the uploaded file. Click **Upload** as, browse to the file you want to upload, and when prompted type the name you want to use. Remember that when connected to a TSO host, a fully qualified data set named needs to be enclosed in single quotes. See Step 5 under Transferring files.  
or
- Drag the file you want to upload from its location to the **File Transfer** dialog box. Click **Refresh** to verify the file was successfully uploaded.

If you cancel the upload process before a file has been completely transferred, a partial file will be left behind on the host.

### Troubleshooting your file transfers

Occasionally you might encounter errors when attempting a file transfer. These errors may be mainframe issues or may be caused by browser security settings.

- If a transfer completes but the file doesn't contain the data expected, verify that the transfer method is properly set to either Binary or ASCII.
- There is a 50MB file size limit on file transfer upload operations. You can [modify this value](#).
- For host-specific errors, see [IBM File Transfer Error Messages](#)

## 5.4.2 AS/400

---


Using AS/400 file transfer you can transfer data between your computer and an iSeries host.

Generally, AS/400 file transfers are straightforward and not complex. However, since the host data is managed as a DB2 database, you can, using the SQL Editor, create fairly complex queries.

### To configure AS/400 file transfer


1. Create an HACloud 5250 terminal session, enter a host name or address and name the session.
2. On the Settings panel, choose File Transfer.
3. Select Enable AS/400 and proceed with the configuration.
  - **Host**  
The host address that you provided for the terminal session is pre-populated in the host field. You can, if needed, use a different host. To specify a different port, append the port number to the host address. For example, host.mycompany.com:23.
  - **TLS Security**  
From the drop down list, select the TLS security option you want to use.  
*To use this option:* The AS/400 database server's certificate must be added to the list of trusted certificates in MSS. If the certificate has not already been added, then see [Trusted Certificates](#) in the MSS documentation for instructions.
  - **Default transfer method**  
Set your preferred default transfer method; fixed width Text, or Comma Separated Values (CSV). The transfer method can be changed when you perform a transfer.
  - **Include column headers by default**  
Select this option to include column headers by default for all downloaded data. You can modify this setting for each download on the file transfer dialog box.  
  
Column headers do not originate on the host file, but are added when a file is downloaded. They are automatically removed when a file is uploaded.
4. Click Save and connect to the session.

### Transferring files (AS/400)


After the session has been configured to use AS/400 file transfer functionality, click  on the toolbar to open the **File Transfer** dialog box. This dialog contains a list of host files that are available to transfer. If prompted, enter your AS/400 login credentials.

- [Downloading files \(AS/400\)](#)
- [Downloading using SQL](#)
- [Uploading files \(AS/400\)](#)
- [Adding a library](#)

#### DOWNLOADING FILES (AS/400)

The AS/400 file system consists of Libraries, Files and Members. Libraries are identified by this icon: . While you cannot download libraries, you can click on the library to see the files and members contained within.

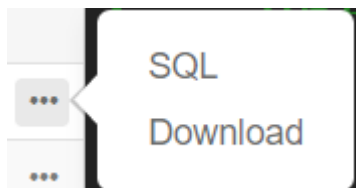
Select **Include column headers** to display column headings for downloaded data.

1. Open the library containing the files (  ).
2. Expand the file that contains the member you want to download.
3. Click on a member to download it.
4. Open the browser's download folder to confirm the file is there. Open the file in a text editor.

#### DOWNLOADING USING SQL

You can create SQL queries to obtain only the data you need from a file member on the host. This lets you select specific fields and ignore others.

1. Open the library and file you want to download.
2. Open the options menu and click **SQL**.



3. The SQL Editor opens and contains the SELECT statement that is used to download the whole member. The file member is referenced as LIBRARYNAME/FILENAME(MEMBERNAME).
4. Click **Run** to download the whole member. or edit the SQL and click **Run** to retrieve a subset of the data.

#### UPLOADING FILES (AS400)

You can only upload data into files as either new or replacement members. The AS/400 file contains a specification describing the data in members and each member in a given file has the same structure. Typically, you cannot (or should not) download a member from one file and upload it to another unless both files have the same data specification.

Since data can only be uploaded as members, you need to open a file and display its members in the file list dialog before the Upload button is enabled.

1. Open the file that you want to upload to. The Upload button is now available.
2. Either:
  - Click the **Upload** button and select a file from your local file system to upload.
  - or
  - Click the down arrow on the Upload button and select **Upload as...** Then select the file, give it a new name, and click OK.

#### ADDING A LIBRARY

Typically, as an AS/400 user, you will have access to a certain set of libraries that has been assigned by a system administrator. These libraries appear as the top level entries in the file transfer dialog. If you need to access a library that is not on your list, your system administrator can update your configuration so that the new library is added to your list. Occasionally you may need to work with a library on a temporary basis; you do not need it added permanently to your library list.


##### To add a library

On the AS/400 file transfer dialog box, click **Add Library**. This button is available from the library list panel. This addition is not permanent and you will have to add the library again if you close and reopen the file transfer dialog box.

### 5.4.3 FTP

---

With Host Access for the Cloud your local computer can act as an FTP client. Using the FTP client, you can connect to an FTP server running on another machine. Once connected, you can view files on the server and use FTP to transfer files between your local computer (or any networked drive) and the FTP server. Using FTP, a client can upload, download, delete, rename, move and copy files on a server, either singly or as a batch transfer, where you can build lists of files to be transferred as one operation.

 **Hint**

If you plan on using a batch transfer, first select and configure the Enable FTP option.

**To configure FTP**



Select Enable FTP and proceed with the configuration:

- **Protocol**

Use FTP to start a standard FTP session. Use SFTP to start an SFTP session.

You can set up an FTP client to use the SFTP protocol and perform all operations over an encrypted secure shell transport. Host Access for the Cloud uses user name and password to authenticate.

- **Host**

Specify the host name or IP address of the FTP server to which you want to connect.

- **Port**

The port of the FTP server specified.

- **If remote files exists when uploading file**

Specify how you want to handle the transfer if a file with the same name already exists.

Select this option	to do this ....
Append	Append the file being sent to the existing fileB1
Ask user (default)	Prompt for a decision on how to handle the duplicate file name
Cancel	Cancel the file transfer
Fail	Cancel the file transfer and receive a notification of failure
Overwrite	Overwrite the existing file on the remote machine
Skip	When multiple files are in a request, skip the file matching an existing file name, but proceed with the transfer for other files.
Unique	Create a new file with a unique file name

- **Initial remote directory**

Specify the path to a home or default directory for the FTP site. When a connection to the FTP site is opened, the server working directory is set automatically to the specified home path. The files and folders in the server home directory appear in the FTP session window. If the initial remote directory is not found, a warning is reported and the connection continues.

- **Anonymous user**

Select this option to log onto the specified FTP server as a guest, with the user name "Anonymous". If the host you are connecting to does not support anonymous users, it may be necessary to supply your credentials.

- **Session timeout (seconds)**

This value tells the FTP client the maximum number of seconds to wait for data packets being transferred to or from the host. If nothing is received within the period specified, a timeout error displays and the transfer terminates; in this case, try the operation again. If you receive repeated timeout errors, increase the timeout value. Entering 0 (zero) in this box prevents the FTP client from ever timing out when waiting for a response. For SFTP sessions, the default is 0 (zero).

- **Keep Alive time (seconds)**

Select this option and enter a time in seconds if you want to continue your connection to the server beyond the server's automatic timeout value for inactivity. Most servers have an idle time value that specifies how long a user's FTP session can last when no activity is detected. When the user exceeds the time limit, the server connection is closed.


This setting allows you to direct the FTP client to send a NOOP command to the server at timed intervals to prevent the server from closing the connection due to inactivity. Be aware that by continuing your session you may prevent another user from making a connection to the FTP server.

### Host encoding

Specifies the character set used by the host to display the names of files that are transferred. By default Host Access for the Cloud uses UTF-8 (Unicode). If you transfer files with the default setting and the file names are unrecognizable, change the Host encoding option to the character set used by the host. (This option does not affect the encoding for the contents of the files that are transferred; it applies to the file names only.)

### Transferring files (FTP)



After the administrator configures a session to include FTP functionality, click  on the toolbar to open the FTP File Transfer window containing a list of host files that are available to transfer. Directories and files are indicated by an icon when you select the file.

1. Select the transfer method. The options are:

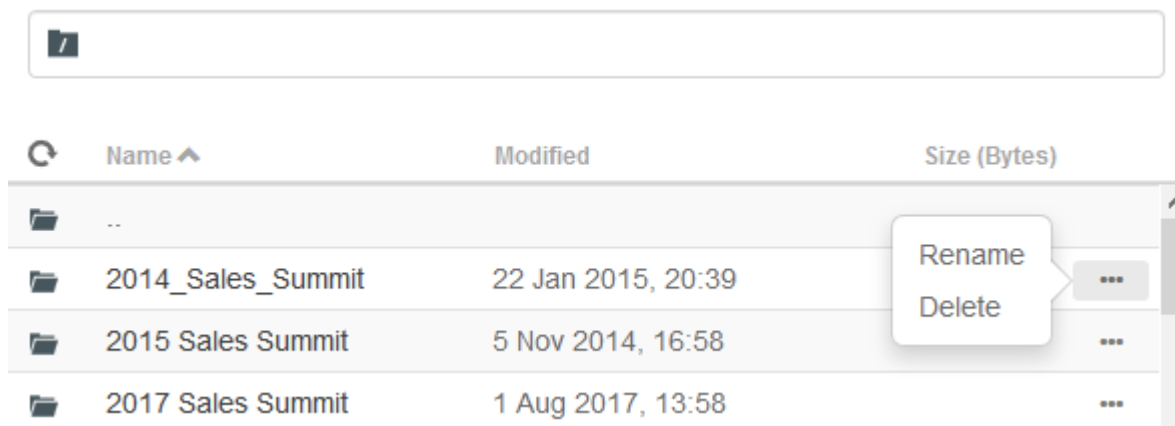
- Binary

Use for program files and other types of files that should not be translated, such as files that have already been formatted for a particular type of printer or files with application-specific formatting. Binary files contain non-printable characters; using this method, a file is not converted or translated during the transfer.

- ASCII

Use to transfer text files with no special formatting. ASCII files on the PC are translated to the EBCDIC character set on the host and host text files are converted from EBCDIC to ASCII when they are downloaded.

2. You can rename, delete, or download a file from the list of files.



3. Refresh the file list at any time by clicking the Refresh icon in the upper left corner of the File Transfer dialog box.

### Downloading files (FTP)

1. From the list, select the file to initiate the transfer.

2. If necessary, you can cancel the transfer from the transfer progress panel.

### Uploading files (FTP)

Choose either method for uploading files:

- From the **File Transfer** dialog box, click **Upload**.

Choose the file you want to upload from the Browse window.

Or

- Drag the file you want to upload from its location to the **File Transfer** dialog box.

Click **Refresh** to verify the file was successfully uploaded.

Click **New Directory** to create a new directory on the remote server. You are prompted to enter the new directory name.

### 5.4.4 Batch transfers

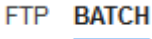


---



#### Note

You must first enable FTP on the File Transfer settings panel FTP tab before you can configure batch transfers.

To transfer multiple files in one operation, use the **Batch** option.

1. From the Settings > File Transfer > FTP panel, check **Enable FTP**.
2. Click  to open the Batch file transfer panel.
3. Select **Cancel batch** when single failure occurs to stop the transfer if a file fails to transfer.
4. Click  to create the list of files you want to transfer.
  - a. Name the list. To aid in building similar lists, you can copy an existing list, rename it, and then add or delete files as needed using the options available when the original list is highlighted.
  - b. From the right panel, click  to open the Add transfer request dialog box.
5. On the Add transfer request panel, begin building the list:

Option	Description
Transfer	Choose whether to upload or download the file.
Local file name	Identify the file you want to transfer. You can enter the name of the file or browse to it.
Remote file path	Provide a location to name and store the file after transfer. You can: <ul style="list-style-type: none"> <li>• <b>Keep original file name and use the initial remote directory</b> - leave the field blank</li> <li>• <b>Use a new file name</b> - enter newfilename.txt. Puts the file in the initial remote directory using the given name.</li> <li>• <b>Keep original file name but use a new directory path</b> - /folder/. Uses the original file name with the new path.</li> <li>• <b>Use new directory and a new file name</b> - /folder/newfilename.txt.</li> </ul>
Transfer method	You can choose from Binary or ASCII transfer methods.
If remote file exists	Decide how to handle file transfer if a remote file already exists. The options are: <ul style="list-style-type: none"> <li>• <b>Overwrite (default)</b> - Overwrite the existing file on the remote machine</li> <li>• <b>Append</b> - Append the file being sent to the existing file</li> <li>• <b>Ask user</b> - Prompt for a decision on how to handle the duplicate file name</li> <li>• <b>Cancel</b> - Cancel the file transfer</li> <li>• <b>Fail</b> - Cancel the file transfer and send notification of the failure</li> <li>• <b>Skip</b> - The file matching an existing file name is skipped, but the transfer proceeds for the other files in the batch</li> <li>• <b>Unique</b> - Create a new file with a unique file name</li> </ul>


6. Click **Save**.

### Transferring files (Batch)







#### Hint

Administrators grant permission to transfer files using the **User Preference Rules** option from the Settings panel.



Click  on the tool bar to open the list that contains the files you want to transfer.

1. Due to browser requirements, you need to specify the location of all files that you want to upload. Locate files as needed using the Search icon. Those files are easily identified with a yellow icon as such:

	Local file name	Transfer	Remote file path
<input checked="" type="checkbox"/>	 Locate "aed.jpg"	<input type="text" value=""/>   Upload	aed.jpg
<input checked="" type="checkbox"/>	 Locate "ascii.txt"	<input type="text" value=""/>   Upload	ascii.txt

2. Files in the batch list are selected by default. To edit the file prior to transfer, you can eliminate files from the transfer operation by clearing their respective check boxes, or by selecting All from the drop down menu. You can also filter the list of transferable files based on their download or upload status.
3. Click Start to initiate the transfer.

## 5.5 Specify Edit Options

---

Edit options to use for different copy, paste and cut operations.

### Copy options

Select text by left-clicking and dragging over it with the mouse or by pressing and holding the shift key while modifying the selection with the arrow keys. By default, different terminal types use different selection modes when copying text. VT terminals use a linear selection mode while all others use block mode selection. To toggle between block and linear selection modes on any terminal type, press and hold down the Alt key, then select the text.

- **Copy input fields only** - Select this option to only copy data from input fields. Data from protected fields is replaced with spaces when placed on the clipboard.
- **Use entire display when there is no selection** - This option applies the Copy command to the entire terminal display when nothing is selected.

### Paste options


Choose Paste to write the contents of the clipboard at the cursor location.

- **Skip protected fields** - Specifies how pasted text is mapped onto the screen:
  - If unselected (the default), the text is interpreted as a linear stream that can contain new lines and delimiters and is pasted accordingly.
  - If selected, the text is interpreted as a host screen data and overlaid onto the current screen starting at the current cursor position. Where the current screen contains an unprotected field, the source text is pasted; where the current screen contains a protected field, the source text is skipped.
- **Wrap to next field on current line** - Select this option to have data pasted from the clipboard fill as much of the current field as possible. Any remaining data will be pasted to the next field on the same row, until the end of the row or the data has been exhausted. If **Wrap text to next line** is also selected, additional data will be pasted to subsequent fields on the next row, and the data is aligned vertically with the starting cursor position.
- **Wrap to next line** - If selected, the Paste command fills the first field with as much Clipboard data as the field will hold. Any remaining text is pasted to the line immediately below, assuming it is writable (for example, an input field). Otherwise, the remaining text is truncated. Subsequent lines of data are pasted to align vertically with the starting cursor position.
 

By default, this option is not selected and text that overflows the field is truncated.
- **Restore starting cursor position after paste** - By default, the host cursor is positioned at the end of the data following a paste operation. Select this option to restore the host cursor to its starting position after the paste operation is complete.

### Cut options

The Cut operation is available for all supported terminals except for VT host types. Select the

area you want to cut, then click the  button on the toolbar. You can use either the context menu or key combination to cut the data from the screen and save it to the clipboard. Data in protected fields is copied to the clipboard but is not removed from the screen.

### Key combinations

Commonly used key combinations for editing functions are supported in HACloud. These keys pass through to the browser, which generates the appropriate editing functions.

Key combination	Host type	Action
Ctrl + C	3270, 5250, UTS, T27, and ALC	Copy
Ctrl + V	3270, 5250, UTS, T27, and ALC	Paste
Ctrl + X	3270, 5250, UTS, T27, and ALC	Cut

These key combinations are mapped in HACloud to various screen edit actions:

Key combination	Host type	Action
Ctrl + A	3270, 5250, UTS, T27, ALC	Select all text on the screen
Shift + Arrow key	All	Change the extent of the current selection
Ctrl + Shift + A	VT	Select all
Ctrl + Shift + C	VT	Copy
Ctrl + Shift + V	VT	Paste

#### Note

In HACloud you can use the key mapper to map editing actions to key combinations. You can access Edit actions using a context menu on the terminal by right-clicking on the screen. Editing actions may be restricted by browser permissions. When an action is unavailable to the user, the associated toolbar buttons and context menu items will not appear.

#### More information

[Editing the Screen](#)



## 5.6 Working with Sessions

---

All the sessions you have access to are available in the Available Sessions list. Sessions are initially created and configured by your system administrator and accessed through a distributed URL (for example, `https://<sessionserver>:7443`).

### To open a session

1. Select the session and click to open.
2. Interact with your host application using the open session.
3. You can create multiple instances of a configured session.

You can have multiple sessions open at a time and easily switch between them using the tabs arranged across the top of the screen. The current session is always the left-most tab and is indicated by a white background and bold text. Each session remains active for 30 minutes.

Use the toolbar to access the various options available to you as you interact with the session. You can disconnect from a session, close the session, turn on Quick Keys, and access other settings. Some options may be only available once your administrator has granted permission.

### 5.6.1 Using Quick Keys

---

The Quick Key terminal keyboard provides a graphical representation of the keys on a host keyboard and gives you quick access to terminal keys.

Click a terminal key on the Quick Key keyboard to send the key to the host. Tool tips, which are available by hovering over a key, provide a description of the mapping.

Quick keys are available for each supported host type and are accessed by clicking the tool bar

icon .

### 5.6.2 Editing the Screen

---

#### Note

Each browser handles copy, paste and cut functions differently and in some cases will not support the use of the toolbar buttons or the right-click context menu. It is highly recommended that you use keyboard commands for those functions. Although keyboard commands vary depending on your operating system, in Windows they are: CTRL+C to copy, CTRL+V to paste, and CTRL+X to cut.

It is far more common to encounter problems with the paste function rather than either cut or copy. If the Paste toolbar button is not visible, it is likely that browser security is preventing read access to the system clipboard. Different browsers behave differently when it comes to providing access to the clipboard. However, pasting is almost always available using the keyboard commands, (Control + V on Windows and Command + V on Macs). This assumes you have not remapped those keys. You can also use the browser's built-in paste menu item or button.

### To copy from the terminal

1. Highlight the area on the terminal screen that you want to copy.
2. Click **Copy** from the toolbar or select **Copy** from the right-click context menu available within the terminal screen. You can alternatively use the keyboard command, **CTRL+C**.

#### To paste into the terminal screen

1. Position the cursor where you want to paste content.
2. If the browser supports the paste function, click **Paste** from the toolbar or select **Paste** from the right-click context menu available within the terminal screen. If your browser does not support this functionality, these options will not be available and you should use the keyboard command, **CTRL+V**.

#### To cut areas from the terminal screen



#### Note

This function is available for all supported terminal types except for VT hosts.

1. Highlight the area on the terminal screen that you want to cut.
2. Click **Cut** from the toolbar or select **Cut** from the right-click context menu available within the terminal screen. You can alternatively use the keyboard command, **CTRL+X**.

#### More information

- [Specify Edit Options](#)

### 5.6.3 Logging Out

---

In the upper right corner of the screen, open the drop-down list associated with your user name and select **Logout** to stop working with the host application.

## 5.7 Creating Macros

---

A macro is a series of keyboard actions that you record and then run. You can use these JavaScript macro programs to automate user interactions with the terminal. You can access and run macros from all supported devices.

Host Access for the Cloud records and saves advanced macros as JavaScript, making it easy to edit and enhance your recorded macros. You can record macros to playback later, run macros at startup or when the session connects or disconnects from the host. You can also write macros from scratch to perform complex tasks that the recorder cannot capture.

Macros are made available to users in two ways; created by an administrator or recorded by users for their own private use. All advanced macros are associated with a session and they all accomplish the same goal, automating host interaction. The only difference between the two flavors is simply who can access them and who manages their creation and availability:

- **Macros created by administrators**

Administrators create macros when they create the session. They are specific to a session and are available to all users who have access to the session from the Macro icon on the toolbar. Administrators can designate macros to run at startup or when the session connects or disconnects from the host.

- **Macros created by users**

End-user macros are created by individuals for sessions they are authorized to access. The administrator grants permission to create macros by setting a User Preference Rule. Users can access the session under their own credentials or in a Guest role. Macros that Guest users create are available to all Guest users. Users who are logged in using their own credentials can only see macros that they have created.

Advanced macros are listed in alphabetical order in the drop down list available from the toolbar. Macros created by the end-user are listed first and followed by an indicator of three vertical grey dots, which when selected, displays the Edit and Delete options. Macros created by the administrator are listed without the indicator as those macros cannot be modified by the end-user.

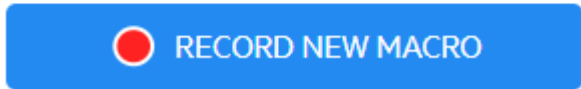
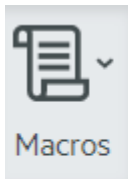
### 5.7.1 Working with Macros


---

Follow these steps to record, edit, and run macros.

### Record

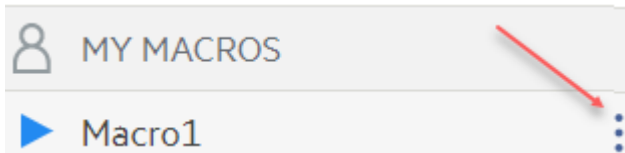
1. Click the Macro icon on the toolbar, and then click Record New Macro.

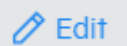


2. Navigate the host application to record the series of steps you want included in the macro.
3. Click  on the toolbar to stop recording. The red dot pulses to indicate the recording is in process.
4. When prompted, type a name for the macro.


### Edit

1. From the Macro drop-down list, select the macro you want to edit.




2. Click the three vertical dots to expand the field.
3. Click  to open the Macro Editor (in the left panel).
4. Use JavaScript to make whatever changes are necessary. You can run and save the modified macro using the toolbar icons in the upper panel of the editor.

### Run

To run a macro, choose the macro from the drop-down list and click .

You can also map keys that will automatically trigger an already recorded macro. In the Key Map settings dialog box, choose **Run Macro** from the **Action** drop down list. Choose a macro to associate with the key mapping from the **Value** list.

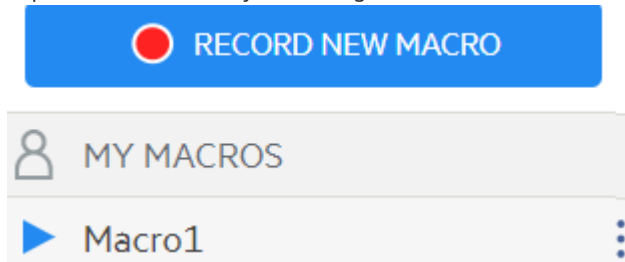
### Stop

You can stop a macro before it completes from either the Macro Editor or the toolbar. Click  to stop the macro. To rerun the macro, navigate back to the macro starting screen.

### Delete

1. From the Macro drop down list, select the macro you want to delete.

- Expand the field, by clicking the three vertical dot icon.



- Click **Delete**.

### View

The Macro drop down list is available from the toolbar to all users who have permission to record macros or are accessing a session where macros have been pre-recorded by the administrator for use with that session.

Macros are listed under either **MY MACROS** or **MACROS** depending on how they were recorded.

All users, whether they are logged in using their credentials or as Guest, can see the macros associated with the session. Macros listed under the **MY MACROS** heading are listed in alphabetical order by name and are visible to those users that recorded them. Macros recorded by the administrator and attached to a session are listed alphabetically under **MACROS**.

## 5.7.2 Debugging Macros

---

Since macros are written in JavaScript and executed in the browser, the best way to debug and troubleshoot them is by using your web browser's built-in tools. Modern browsers come with a very capable set of tools for debugging JavaScript code. You can place breakpoints, step through code, and output debug information.

### Hint

JavaScript is case sensitive. Keep that in mind when editing JavaScript code.

**To debug a macro:**

1. Open the macro for editing. See [Working with macros](#) for instructions.
2. Open your browser's development tools.

Browser	Open debugger
Mozilla Firefox 40.0.3	<ul style="list-style-type: none"> <li>• From the toolbar, open the Menu, and choose Developer.</li> <li>• From the Web Developer menu, choose Debugger. The debugger opens in a lower panel.</li> </ul>
Google Chrome 45.0	<ul style="list-style-type: none"> <li>• From the toolbar, open the Menu, and choose More tools.</li> <li>• Choose Developer Tools to open the Debugger.</li> </ul>
Microsoft Internet Explorer 11	<ul style="list-style-type: none"> <li>• From the toolbar, open Settings, and choose F12 Developer Tools.</li> <li>• Open the Debugger tab.</li> </ul>

3. Use one of the these tools in your macro code, and run the code.

- debugger

The most thorough approach to debugging is to use the `'debugger;'` statement. When you insert these statements into your macro code then run it, with the browser's development tools open, the execution will stop on those lines. You can step through your macro, view the value of local variables and whatever else you need to check.

You are encouraged to place multiple `debugger;` statements in your code to help get to the correct line. The asynchronous nature of JavaScript can make stepping through code challenging. This can be offset by using multiple, carefully placed `debugger;` statements.

**Example 1:** `debugger`

```
var hostCommand = menuSelection + '[enter]';
debugger; // ← Browser's debugger will stop here
ps.sendKeys(hostCommand);
```

- `console.log()`, `alert()`

These two functions are commonly used for debugging JavaScript. While not as flexible as the debugger statement they provide a quick way to output debug information. These functions output the information to the JavaScript "Console" tab in the browser's developer tools.

**Example 2:** `console.log()`, `alert()`

```
var hostCommand = menuSelection + '[enter]';
console.log('Command:' + hostCommand); // ← Will output the string to "Console" tab
alert('Command:' + hostCommand); // Will pop up a small window containing the data
ps.sendKeys(hostCommand);
```

- `ui.message()`

The Host Access for the Cloud Macro API provides an `ui.message()` function that is very similar to JavaScript's `alert()` function. You can also use `ui.message()` to output debug information.

**Example 3:** `ui.message()`

```
var hostCommand = menuSelection + '[enter]';
ui.message('Command:' + hostCommand); // ← Will pop up a message window
ps.sendKeys(hostCommand);
```

**Keep in mind:**

- Stepping and “yields”

While the yield statements in macros make them easier to understand, they can make the code more challenging to step through with the debugger. Consider either using multiple debugger statements or carefully placed debugger statements of `console.log()` calls to output the right debug information.

- Internet Explorer

Debugging in Internet Explorer involves transformed code and may be more difficult than on other browsers.

### 5.7.3 Using the Macro API

---

In Host Access for the Cloud, macros are recorded and written using JavaScript.

The Macro API consists of a set of objects which you can use to interact with the host, wait for screen states, and interact with the user.

#### About promises and yields

Because JavaScript is single-threaded and uses 'callback functions' and 'promises' to help manage the flow of execution through code, often code can be difficult to follow. Host Access for the Cloud combines the concept of 'promises' with the 'yield' keyword so macro code can be organized in a more linear fashion.

- **Promises**

Promises are patterns to help simplify functions that return their result asynchronously, at some point in the future. All 'wait' and 'ui' functions in the Macro API return promise objects.

- **Yield**

Macros use the yield keyword to block the execution of the macro until a promise is resolved, or done. So putting yield in front of any 'wait' or 'ui' function makes the macro execution pause until that function has finished executing. You can place the yield keyword in front of any function that returns a promise, even your own custom functions.

#### Note

The ability to make macro execution block by combining yield with promises is enabled by the `createMacro()` function.

#### Errors

Errors are handled in macros using a try / catch statement. Some API functions may throw errors if, for example, conditions can't be met or a timeout occurs. The thrown error is 'caught' in the catch statement. You can wrap smaller blocks of code in a try / catch statement to handle errors at a more granular level.

Macro developers can also throw errors by using `throw new Error('Helpful error message');`

#### More information

- [Macro API Objects](#)
- [Sample Macros](#)

## 5.8 Macro API Objects

---

You can create macros using the Macro API. By default for use in macros, there are four primary objects available:

- **Session** - the main entry point for access to the host. You use the Session object to connect, disconnect and provide access to the PresentationSpace object.
- **PresentationSpace** - represents the screen and provides many common capabilities such as getting and setting the cursor location, sending data to the host and reading from the screen. It is obtained by calling `session.getPresentationSpace()`.
- **Wait** - provides a simple way to wait for various host states to occur before continuing to send more data or read from the screen. For example, you can wait for the cursor to be located at a certain position, text to be present in a position on the screen or simply wait for a fixed amount of time. All 'Wait' function calls require the yield keyword, which is explained below.
- **User Interface** - automatically available in your macro as the "ui" variable. It provides basic user interface capabilities. You can use this object to display data to the user or prompt them for information. All 'UI' function calls require the yield keyword.

### All available objects

See the list of available objects in the right navigation, "On this page." (You may need to expand your browser.)

---



## 5.8.1 Attribute

---

Use the `Attribute`, along with the `AttributeSet`, to decode the formatting information present on the data cell.

<b>Attribute</b>	<b>Indicates...</b>
PROTECTED	a protected data cell
MODIFIED	a modified data cell
NUMERIC_ONLY	the beginning of a numeric only data cell
ALPHA_NUMERIC	an alpha numeric data cell.
HIGH_INTENSITY	whether the data cell contains high intensity text
HIDDEN	whether the data cell contains hidden text
PEN_DETECTABLE	whether the data cell is pen detectable
ALPHA_ONLY	an alpha only data cell
NUMERIC_SHIFT	the beginning of a numeric shift field
NUMERIC_SPECIAL	the data cell marks the beginning of a numeric special field
KATAKANA_SHIFT	a section of Katakana text
MAGNETIC_STRIPE	the data cell marks the beginning of a magnetic strip field
SIGNED_NUMERIC_ONLY	the data cell is a signed numeric field
TRANSMIT_ONLY	the data cell is a transmit only field
FIELD_END_MARKER	the data cell marks the end of a modified field
FIELD_START_MARKER	the data cell marks the start of a modified field
SPECIAL_EMPHASIS_PROTECTED	a special emphasis protected field
TAB_STOP	that the data cell contains a tab stop
REVERSE	the data cell displays in reverse video mode
BLINKING	the data cell contains blinking text
RIGHT_JUSTIFIED	the data cell marks the beginning of a right justified field
LEFT_JUSTIFIED	the data cell marks the beginning of a left justified field
LOW_INTENSITY	the data cell contains low intensity text
UNDERLINE	the data cell contains underlined text
DOUBLE_BYTE	the data cell contains double byte text
COLUMN_SEPARATOR	the data cell contains a column separator
BOLD	the data cell contains bold text
DOUBLE_WIDTH	the data cell marks a double width field
DOUBLE_HEIGHT_TOP	a double height top data cell
DOUBLE_HEIGHT_BOTTOM	a double height bottom data cell
CONTROL_PAGE_DATA	the data cell contains control page data
RIGHT_COLUMN_SEPARATOR	the data cell contains a right column separator
LEFT_COLUMN_SEPARATOR	a data cell containing a left column separator
UPPERSCORE	the data cell contains an upperscore
STRIKE_THROUGH	the data cell contains strike through text

## 5.8.2 AttributeSet

---

The AttributeSet object allows the user to decode the attributes that are present on the data cell. The AttributeSet object returns values defined in the [Attribute](#) object and when used together, you can get formatting information from the data cell.

Method	Description
<code>contains(attribute)</code>	<p>Determines if the set contains the specified <a href="#">Attribute</a>.</p> <p><b>Parameters</b></p> <p><code>{Number}</code> attribute to check</p> <p><b>Returns</b></p> <p><code>{Boolean}</code> True if the attribute is in the set</p>
<code>isEmpty()</code>	<p>Determines if the attribute set is empty.</p> <p><b>Returns</b></p> <p><code>{Boolean}</code> True if the set is empty.</p>
<code>size()</code>	<p>Indicates the number of attributes in a set.</p> <p><b>Returns</b></p> <p><code>{Number}</code> The attribute count.</p>
<code>toArray()</code>	<p>Converts the internal attribute set to an array.</p> <p><b>Returns</b></p> <p><code>{Number[]}</code> Array of values of attributes in the set.</p>
<code>toString()</code>	<p>Converts the internal attribute set to a string.</p> <p><b>Returns</b></p> <p><code>{String}</code> Space-delimited names of attributes in the set.</p>
<code>forEach(callback, thisArg)</code>	<p>Function to iterate over each element in the attribute set.</p> <p><b>Parameters</b></p> <p><code>{forEachCallback}</code> Callback to perform the specific operation. Called with the name of each attribute in the set.</p> <p><code>{Object}</code> <code>thisArg</code> optional pointer to a context object.</p>
<code>forEachCallback(string, object)</code>	<p>A user provided callback function where you provide the behavior, to be used as the callback parameter to <code>forEach</code>.</p> <p><b>Parameters</b></p> <p><code>{String}</code> String name of an attribute in the attribute set.</p> <p><code>{Object}</code> <code>thisArg</code> optional pointer to a context object.</p>

---

## 5.8.3 AutoSignOn

---

Method	Description
<code>getPassTicket()</code>	<p>Obtains a pass ticket to be used for signing onto a mainframe application. Multiple pass tickets may be requested using different application IDs.</p> <p><b>Parameters</b> {String} application ID tells the host which application the sign on is for</p> <p><b>Returns</b> {Promise} fulfilled with the pass ticket key or rejected if the operation fails. The pass ticket obtained from DCAS only works with the current host session and is valid for ten minutes.</p>
<code>sendUserName()</code>	<p>Applies the user name contained in the pass ticket to the field at the current cursor location on the current host screen. The user name must be sent before the password. Sending the password first will invalidate the pass ticket, and you will need to get another one.</p> <p><b>Parameters</b> {String} passTicketKey obtained from getPassTicket</p> <p><b>Returns</b> {Promise} fulfilled if the user name is successfully sent. Rejected if the operation fails.</p>
<code>sendPassword()</code>	<p>Applies the password contained in the pass ticket to the field at the current cursor location on the current host screen. The user name must be sent before the password. Sending the password first will invalidate the pass ticket, and you will need to get another one.</p> <p><b>Parameters</b> {String} passTicketKey obtained from getPassTicket</p> <p><b>Returns</b> {Promise} fulfilled if the password is successfully sent. Rejected if the operation fails.</p>

---

## 5.8.4 Color

---

Color constants to use for the DataCell object foreground and background colors.

Color	Description	Numeric Value
BLANK_UNSPECIFIED	No color specified	0
BLUE	Blue	1
GREEN	Green	2
CYAN	Cyan	3
RED	Red	4
MAGENTA	Magenta	5
YELLOW	Yellow	6
WHITE_NORMAL_INTENSITY	Normal intensity white	7
GRAY	Gray	8
LIGHT_BLUE	Light blue	9
LIGHT_GREEN	Light green	10
LIGHT_CYAN	Light cyan	11
LIGHT_RED	Light red	12
LIGHT_MAGENTA	Light magenta	13
BLACK	Black	14
WHITE_HIGH_INTENSITY	High intensity white	15
BROWN	Brown	16
PINK	Pink	17
TURQUOISE	Turquoise	18

---

## 5.8.5 ControlKey

---

The ControlKey object defines constants for sending cursor control keys and host commands using the sendKeys method. Constants are available for these host types:

- [IBM 3270](#)
  - [IBM 5250](#)
  - [VT](#)
  - [UTS](#)
-

**IBM 3270**

<b>Key word</b>	<b>Description</b>
ALTVIEW	Alternate view
ATTN	Attention
BACKSPACE	Back space
BACKTAB	Back tab
CLEAR	Clear or clear display
CURSOR_SELECT	Cursor select
DELETE_CHAR	Delete, delete character
DELETE_WORD	Delete word
DEST_BACK	Destructive backspace
DEV_CANCEL	Device cancel
DOWN	Cursor down
DSPSOSI	Display S0/SI
DUP	Duplicate field
END_FILE	End of field
ENTER	Enter
ERASE_EOF	Erase end of field
ERASE_FIELD	Erase field
ERASE_INPUT	Erase input
FIELD_MARK	Field mark
HOME	Cursor home
IDENT	Ident
INSERT	Insert
LEFT_ARROW	Cursor left
LEFT2	Left two positions
NEW_LINE	New line
PA1 - PA3	PA1 - PA3
PF1 - PF24	PF1 - PF24
PAGE_DOWN	Page down
PAGE_UP	Page up
RESET	Reset, reset terminal
RIGHT2	Right 2 positions
RIGHT_ARROW	Cursor right, right
SYSTEM_REQUEST	System request
TAB	Tab key
UP	Cursor up

**IBM 5250**

<b>Key word</b>	<b>Description</b>
ALTVIEW	Alternate view
ATTN	Attention
AU1 - AU16	AU1 - AU16
BACKSPACE	Back space
BACKTAB	Back tab
BEGIN_FIELD	Begin field
CLEAR	Clear or clear display
DELETE_CHAR	Delete, delete character
DEST_BACK	Destructive backspace
DOWN	Cursor down
DSPSOSI	Display S0/SI
DUP	Duplicate field
END_FILE	End of field
ENTER	Enter
ERASE_EOF	Erase end of field
ERASE_FIELD	Erase field
ERASE_INPUT	Erase input
FIELD_EXT	Field exit
FIELD_MINUS	Field minus
FIELD_PLUS	Field plus
FIELD_MARK	Field mark
HELP	Help request
HEXMODE	Hex mode
HOME	Cursor home
INSERT	Insert
LEFT_ARROW	Cursor left
NEW_LINE	New line
PA1 - PA3	PA1 - PA3
PF1 - PF24	PF1 - PF24
PAGE_DOWN	Page down
PAGE_UP	Page up
[print]	Print
RESET	Reset, reset terminal
RIGHT_ARROW	Cursor right, right
SYSTEM_REQUEST	System request
TAB	Tab key



<b>Key word</b>	<b>Description</b>
-----------------	--------------------

UP	Cursor up
----	-----------

---

## VT

Key word	Description
BACKSPACE	Back space
BREAK	Break
CLEAR	Clear or clear display
CURSОР_SELECT	Cursor select
DELETE_CHAR	Delete, delete character
DOWN	Cursor down
EK_FIND	Edit keypad find
EK_INSERT	Edit keypad insert
EK_NEXT	Edit keypad next
EK_PREV	Edit keypad previous
EK_REMOVE	Edit keypad remove
EK_SELECT	Edit keypad select
END_FILE	End of field
ENTER	Enter
F1 - F24	F1 - F24
HOLD	Hold
HOME	Home
INSERT	Insert
KEYPAD_COMMA	Keypad comma
KEYPAD_DOT	Keypad decimal
KEYPAD_ENTER	Keypad enter
KEYPAD_MINUS	Keypad minus
KEYPAD0 - KEYPAD9	Keypad 0 - Keypad 9
LEFT_ARROW	Cursor left
PF1 - PF20	PF1 - PF20
PAGE_DOWN	Page down
PAGE_UP	Page up
RESET	Reset, reset terminal
RETURN	Return, carriage return
RIGHT_ARROW	Cursor right, right
TAB	Tab key
UDK16 - UDK20	User defined key 6 - User defined key 20
UP	Cursor up

**UTS**

<b>Key word</b>	<b>Description</b>
BACKSPACE	Back space
BACKTAB	Back tab
CHAR_ERASE	Erases character at the cursor and advances the cursor.
CLEAR_DISPLAY	Clear display
CLEAR_EOD	Clear to end of display
CLEAR_EOF	Clear to end of field
CLEAR_EOL	Clear to end of line
CLEAR_FCC	Clear Field Control Character
CLEAR_HOME	Clear display and cursor home
CONTROL_PAGE	Toggles the control page
DELETE_LINE	Deletes the line containing the cursor and shifts remaining lines up one row
DELIN_LINE	Deletes character under cursor and shifts remaining characters on line to the left.
DELIN_PAGE	Deletes character under cursor and shifts remaining characters on page to the left.
DOWN	Moves the cursor down one line. Wraps at bottom.
DUP_LINE	Creates a copy of the current line and overwrites the next line with the duplicate.
END_FIELD	Moves the cursor to the end of the current field.
END_PAGE	Moves the cursor to the end of the current page.
EURO	Inserts the Euro character
F1 - F22	Function keys F1-F22
HOME	Moves the cursor to beginning of current page (row 1, col 1)
INSERT	Toggles insert/overwrite mode.
INSERT_IN_LINE	Inserts space at cursor position and shifts the remaining characters on the line to the right. The character in the far right column on the line is discarded.
INSERT_IN_PAGE	Inserts space at cursor position and shifts the remaining characters on the page to the right. The character in the far right column on each line is discarded.
INSERT_LINE	Inserts a new line at the cursor row and shifts the remaining lines down. The last row on the page is discarded.
LEFT_ARROW	Moves the cursor one position to the left wrapping if necessary.
LOCATE_FCC	Finds the next field control character on the screen.
MSG_WAIT	Retrieves messages queued to the terminal.
RETURN	Carriage return
RIGHT_ARROW	Moves the cursor one position to the right, wrapping if necessary.
SOE	Inserts the Start of Entry character
START_OF_FIELD	Moves the cursor to the beginning of the field.

Key word	Description
START_OF_LINE	Moves the cursor to column 1 of current line.
TAB	Moves the cursor to the next tab position of the screen.
TOGGLE_COL_SEP	Toggles the column separator attribute.
TOGGLE_STRIKE_THRU	Toggles the strike-through attribute on the current data cell.
TOGGLE_UNDERLINE	Toggles the underline attribute on the current data cell.
TRANSMIT	Transmits changed field data to the host.
UNLOCK	Sends the UNLOCK key to the host.
UP	Moves the cursor up one row, wrapping if necessary.

## 5.8.6 DataCell

The DataCell object provides information about a particular position on a terminal screen.

Method	Description
getPosition()	Returns the position of this data cell on the screen. <b>Returns</b> {Position} the position of the data cell on the screen
getChar()	Obtains the character associated with the cell. <b>Returns</b> {String} The character associated with the cell.
getAttributes()	Returns the set of attributes specified for this data cell instance. See <a href="#">AttributeSet</a> . <b>Returns</b> {AttributeSet} The set of attributes for this data cell instance.
getForegroundColor()	Returns the foreground color, as defined in the Color object, for this data cell. <b>Returns</b> {Number} Foreground color for this data cell. The color is defined in the <a href="#">Color</a> object.
getBackgroundColor()	Returns the background color, as defined in the Color object, for this data cell. <b>Returns</b> {Number} Background color for this data cell. The color is defined in the <a href="#">Color</a> object.
toString	Converts the internal data cell to a string. <b>Returns</b> {String} The string representation of a data cell.
isFieldDelimiter()	Tests if this cell represents a field delimiter. <b>Returns</b> {Boolean} True if this cell is a field delimiter, false if otherwise.

## 5.8.7 Dimension

---

Represents the size of the screen or screen area.

Method	Description
<code>Dimension(rows,cols)</code>	Creates a new Dimension instance. <b>Parameters</b> {Number} rows screen rows dimension {Number} cols screen columns dimension

---

## 5.8.8 Field

---

Use the `Field` object, along with `FieldList`, to obtain the information present in a field on the screen.

Method	Description
<code>getAttributes()</code>	Returns the set of attributes specified for this field instance. See <a href="#">AttributeSet</a> . <b>Returns</b> {AttributeSet} The set of attributes for this field.
<code>getForegroundColor()</code>	Returns the foreground color for the field. <b>Returns</b> {Number} Foreground color for this field. These values are defined in the <a href="#">Color</a> object.
<code>getBackgroundColor()</code>	Returns the background color of the field. <b>Returns</b> {Number} Background color for this field. These values are defined in the <a href="#">Color</a> object.
<code>getStart()</code>	Returns the starting position of the field. The starting position is the position of the first character of the field. Some host types use a character position to store field level attributes. In this case, the attribute position is not considered the start position. <b>Returns</b> {Position} Starting position of the field. <b>Throws</b> {RangeError} For zero length fields.
<code>getEnd()</code>	Returns the ending position of the field. The ending position is the position in the presentation space containing the last character of the field. <b>Returns</b> {Position} Ending position of the field. <b>Throws</b> {RangeError} For zero length fields.
<code>getLength()</code>	Returns the length of the field. For host types that use a character position to store the field attributes, the field length does not include the field attribute position <b>Returns</b> {Number} Length of the field.
<code>getDataCells()</code>	Obtains the data cells that comprise this field. See <a href="#">DataCell</a> . <b>Returns</b> {DataCell[]} Data cells that comprise this field.
<code>getText()</code>	Obtains the text from the field. <b>Returns</b> {String} field text.
<code>setText()</code>	Sets the field text. For certain host types, like VT, the text is transmitted to the host right away, but in other host types, the text is not transmitted to the host until an Aid key is invoked. If the text is shorter than the field, the text is placed in the host field, and the remainder of the field is cleared. If the text is longer than the host field, then as much text as will fit is placed in the field. <b>Parameter</b> {String} Text to set on the field. <b>Throws</b> {Error} If the field is protected.
<code>clearField()</code>	Clears the current field in an emulation-specific manner. <b>Throws</b> {Error} If the field is protected or clear is not supported.
<code>getPresentationSpace()</code>	



Obtains the `PresentationSpace` that created this field.

**Returns**

{`PresentationSpace`} Parent of this field instance.

`toString()`

Creates a user-friendly description of the field.

**Returns**

{`String`} A user readable rendition of the field.

## 5.8.9 FieldList

Use the `FieldList` object, along with `Field` object, to obtain field list information.

Method	Description
<code>getPresentationSpace()</code>	Obtains the <code>PresentationSpace</code> that created this field. <b>Returns</b> { <code>PresentationSpace</code> } Parent of this field instance.
<code>findField(position, text, direction)</code>	Returns the field containing the specified text. The search starts from the specified position and proceeds either forward or backward. If the string spans multiple fields, the field containing the starting position is returned. When searching forward the search will not wrap to the top of the screen. When searching backward the search will not wrap to the bottom of the screen. <b>Parameters</b> { <code>Position</code> } Position from which to start the search. See <code>Position</code> object. { <code>String</code> } The text to search for (optional). If not provided, returns the next field to the right of or below the specified position. { <code>Number</code> } direction of the search (optional). Use <code>PresentationSpace.SearchDirection</code> constants for this parameter. For example, <code>PresentationSpace.SearchDirection.FORWARD</code> or <code>PresentationSpace.SearchDirection.BACKWARD</code> . If not provided, searches forward. <b>Returns</b> { <code>Field</code> } containing the string or null if a field meeting the given criteria is not found. <b>Throws</b> { <code>RangeError</code> } If the position is out of range.
<code>get(index)</code>	Obtains the field at the given index. <b>Parameters</b> { <code>Number</code> } index into the field list. <b>Returns</b> { <code>Field</code> } located at the specified index. <b>Throws</b> { <code>RangeError</code> } If the index is out of range.
<code>isEmpty()</code>	Determines if the field list is empty. <b>Returns</b> { <code>Boolean</code> } True if the list is empty.
<code>size()</code>	Indicates the number of fields in the list. <b>Returns</b> { <code>Number</code> } The field count.
<code>toString()</code>	Creates a user-friendly description of the field list. <b>Returns</b> { <code>String</code> } A user readable rendition of the field list.


## 5.8.10 FileTransfer

---

Use the FileTransfer object to list and transfer files between the host system and the client.

The Host Access for the Cloud file transfer API abstracts the file path conventions used by different host file implementations. Follow URL or Linux file system path formats when formatting file paths used by the API. For example, `/root/directory/file`.

It is important to observe any rules specific to host systems, such as allowable characters or name lengths.

 **Note**

Browsers place significant security restrictions around the ability of Javascript to interact with client file systems.

Method	Description
<code>getHostFileListing(remotePath)</code> <code>()</code>	<p>Request a listing of host files. If <code>RemotePath</code> is omitted, a file listing for the current remote working directory is shown.</p> <p><b>Parameters</b></p> <p><code>{String}</code> (optional) If specified will get file listing for specified remote path. If not specified, will get file listing for current remote working directory.</p> <p><b>Returns</b></p> <p><code>{Promise}</code> Resolves to an array of <code>HostFile</code> objects contained at <code>remoteName</code>. Rejected if the remote path cannot be read.</p>
<code>sendFile(localFile, remoteName)</code>	<p>Sends specified file to the host.</p> <p><b>Parameters</b></p> <p><code>{File}</code> Javascript file object pointing to local file to send.</p> <p><code>{String}</code> (optional) Fully-qualified remote file name as allowed by remote system (Unix, Windows, MVS, VAX).</p> <p><b>Returns</b></p> <p><code>{Promise}</code> fulfilled with a <code>HostFile</code> object representing the sent file on success. Rejected if an error occurred sending the file.</p>
<code>getDownloadURL(remoteName)</code>	<p>Constructs a link to download a file from a host system.</p> <p><b>Parameters</b></p> <p><code>{String}</code> Fully-qualified remote file name as allowed by remote system (Unix, Windows, MVS, VAX).</p> <p><b>Returns</b></p> <p><code>{URL}</code> that can be used to retrieve the file from the Host Access for the Cloud session server.</p>
<code>setTransferOptions(options)</code>	<p>Set transfer options for current <code>FileTransfer</code> session. The transfer options are applied to all future transfers until the session is either exited or overridden by another call to <code>setTransferOptions</code>.</p> <p><b>Parameters</b></p> <p><code>{JSON}</code> see <code>FileTransferOptions</code> for allowed names and values.</p> <p><b>Returns</b></p> <p><code>{Promise}</code> fulfilled when the call completes. Rejected if an error occurred setting the options.</p>
<code>cancel()</code>	<p>Cancels the current transfer in progress.</p> <p><b>Parameters</b></p> <p><code>{String}</code> Fully-qualified remote file name as allowed by remote system (Unix, Windows, MVS, VAX).</p> <p><b>Returns</b></p> <p><code>{Promise}</code> fulfilled when the call completes. Rejected if an error occurred canceling the transfer.</p>

## 5.8.11 FileTransferFactory

---

A fileTransferFactory object is available to all macros. If file transfers are configured for the session, you can use it to get a reference to a FileTransfer object.

Method	Description
getIND\$File()	Returns a FileTransfer object for interacting with the configured Ind\$File type for the session. <b>Returns</b> {FileTransfer} <b>Throws</b> {Error} If the session hasn't been configured to allow IND\$File transfers.

---

## 5.8.12 FileTransferOptions

---

File transfer option object specification. Example:

```
`fileTransfer.setTransferOptions({ transferMethod : 'ascii' });``
```

Method	Description
transferMethod	{String} Allowed values: <ul style="list-style-type: none"> <li>'ascii'</li> <li>'binary'</li> </ul>

---

## 5.8.13 HostFile

---

A HostFile object represents a file on the host file system.

Method	Description
getName()	Gets the file name. <b>Returns</b> {String} the file name.
getParent()	Gets the parent of this host file. <b>Returns</b> {String} the parent of this host file. This means different things on different host types. For example on TSO this is the name of the catalog in which the file resides.
getSize()	The byte size of the file. <b>Returns</b> {Number} the size of the file in bytes.
getType()	The type of file represented. <b>Returns</b>

---

## 5.8.14 HostFileType

---

The HostFileType object defines constants for determining the type of a HostFile object.

Value	Description
FILE	Represents a file on the host system.
DIR	Represents a directory on the host system.
UNKNOWN	Represents a host file of unknown origin.

---

## 5.8.15 OIA

---

Operator Information Area (OIA) interface. The OIA object returns values that are defined in the OIAStatus object.

Method	Description
<code>getStatus ()</code>	Returns the set of enabled status flags. See <a href="#">StatusSet</a> . <b>Returns</b> {StatusSet} Containing the status information.
<code>getCommErrorCode()</code>	Returns the current communication error code. <b>Returns</b> {Number} the current communication error code. If one doesn't exist, it will be 0.
<code>getProgErrorCode()</code>	Returns the current program error code. <b>Returns</b> {Number} the current program error code. If one doesn't exist, it will be 0.

---

## 5.8.16 OIAStatus

---

OIAStatus	Description
CONTROLLER_READY	Controller ready
A_ONLINE	Online with a non-SNA connection
MY_JOB	Connected to a host application
OP_SYS	Connected to a SSCP (SNA)
UNOWNED	Not connected
TIME	Keyboard inhibited
SYS_LOCK	System lock following AID key
COMM_CHECK	Communication check
PROG_CHECK	Program check
ELSEWHERE	Keystroke invalid at cursor location
FN_MINUS	Function not available
WHAT_KEY	Keystroke invalid
MORE_THAN	Too many characters entered in the field
SYM_MINUS	Symbol entered not available
INPUT_ERROR	Operator input error (5250 only)
DO_NOT_ENTER	Do not enter
INSERT	Cursor in insert mode
GR_CURSOR	Cursor in graphics mode
COMM_ERR_REM	Communications error reminder
MSG_WAITING	Message waiting indicator
ENCRYPT	Session is encrypted
NUM_FIELD	Invalid character in numeric only field

## 5.8.17 Position

---

Method	Description
<code>Position(row,col)</code>	<p>Creates a new Position instance.</p> <p><b>Parameters</b></p> <p>{Number} row screen row coordinate</p> <p>{Number} col screen column coordinate</p>

## 5.8.18 PresentationSpace

---

Use the PresentationSpace object to interact with the terminal screen. Setting and getting the cursor position, sending keys, and reading text are some of the interactions available.

Method	Description
<code>getCursorPosition()</code>	<p>Returns a <a href="#">Position</a> instance representing the current cursor position. An unconnected session has a cursor position of 0,0.</p> <p><b>Returns</b>  <a href="#">{Position}</a> current cursor location</p>
<code>setCursorPosition(position)</code>	<p>Moves the host cursor to the specified row and column position. For some hosts, such as VT, the host may constrain the movements of the cursor.</p> <p><b>Parameters</b>  <a href="#">{Position}</a> <a href="#">Position</a> new cursor position.</p> <p><b>Returns</b>  None</p> <p><b>Throws</b>  <a href="#">{RangeError}</a> If the position is not valid on the current screen.</p>
<code>isCursorVisible()</code>	<p>Tests that the cursor is currently visible in the presentation space. The cursor is considered not visible if the session is not connected.</p> <p><b>Returns</b>  <a href="#">{Boolean}</a> True if the cursor is visible. False if the cursor is not visible.</p>
<code>sendKeys(keys)</code>	<p>Transmits a text string or <a href="#">ControlKey</a> to the host at the current cursor position in the presentation space. If the cursor is not in the desired position, then use <code>setCursorPosition</code> function first. The text string can contain any number of characters and <a href="#">ControlKey</a> objects.</p> <p>For example: "myname" + <a href="#">ControlKey.TAB</a> + "mypass" + <a href="#">ControlKey.ENTER</a> will transmit a user ID, tab to the next field, transmit a password, and then transmit the Enter key.</p> <p>If you need to transmit a square bracket, double the brackets ([[ or ]]).</p> <p><b>Parameters</b>  <a href="#">{String}</a> keys text and/or control keys to transmit</p>
<code>getText(start,length)</code>	<p>Returns a string representing a linear area of the presentation space. No new line characters are inserted if row boundaries are encountered.</p> <p><b>Parameters</b>  <a href="#">{Position}</a> start position from which to retrieve text  <a href="#">{Number}</a> length the maximum number of characters to return. If the length parameter causes the last position of the presentation space to be exceeded then only those characters up to the last position will be returned.</p> <p><b>Returns</b>  <a href="#">{String}</a> representing a linear area of the presentation space which may be empty if the session is not connected.</p> <p><b>Throws</b>  <a href="#">{RangeError}</a> If the position or length are not valid on the current screen.</p>
<code>getSize()</code>	<p>Gets the dimensions of the screen as a <a href="#">Dimension</a> object.</p> <p><b>Returns</b>  <a href="#">{Dimension}</a> Containing the number of rows and columns. The screen size is [row:0, col:0] if the session is not connected.</p>
<code>getDataCells(start, length)</code>	<p>Returns <a href="#">DataCell</a> instances where the first member will be for the position specified by the start parameter. The maximum number of <a href="#">DataCell</a> instances in the list is specified by the length parameter.</p> <p><b>Parameters</b>  <a href="#">{Position}</a> start the first position on the host screen in which to</p>



retrieve DataCell instances. See [Position](#).

`{Number}` length of the maximum number of DataCell instance to be retrieved. If not specified, returns DataCells from the start position to the end of the screen.

**Returns**

`{DataCell[]}` Instances which may be empty if the session is not connected. If position is not specified, returns all DataCells. If length is not specified, returns DataCells from the start position to the end of the screen.

**Throws**

`{RangeError}` if start or length are out of range.

---

<code>getFields()</code>	<p>Returns a list of the fields in the presentation space. If the host type does not support fields or the current screen is not formatted then the return value will always be an empty list. See <a href="#">FieldList</a>.</p> <p><b>Returns</b></p> <p><code>{FieldList}</code> of host defined fields in the presentation space.</p>
--------------------------	---

---

## 5.8.19 Session

---

The session object is the main entry point for interacting with the host. It contains functions for connecting, disconnecting, and obtaining the PresentationSpace object.

Method	Description
<code>connect()</code>	<p>Connects to the configured host. If needed, use <code>, wait.forConnect()</code> to block macro execution until the session is connected.</p> <p><b>Returns</b></p> <p>None</p>
<code>disconnect()</code>	<p>Disconnects from the configured host. If needed, use <code>wait.forDisconnect()</code> to block macro execution until the session is connected.</p> <p><b>Returns</b></p> <p>None</p>
<code>isConnected()</code>	<p>Determines whether the connection to the host is connected.</p> <p><b>Returns</b></p> <p><code>{Boolean}</code> true if host connection is established; false if not.</p>
<code>getPresentationSpace()</code>	<p>Provides access to the PresentationSpace instance for this session.</p> <p><b>Returns</b></p> <p><code>{PresentationSpace}</code> instance associated with this session.</p>
<code>getDeviceName()</code>	<p>Returns the device name for a connected session or an empty string if the session is disconnected or doesn't have device name.</p> <p><b>Returns</b></p> <p><code>{String}</code> The connected device name.</p>
<code>getType()</code>	<p>Returns the type of host session. See <a href="#">SessionType</a>.</p> <p><b>Returns</b></p> <p><code>{String}</code> The type of host session.</p>
<code>setDeviceName()</code>	<p>Provides a means to modify the device name on a session instance.</p> <p><b>Parameters</b></p> <p><code>{String}</code> name Device name to use when connecting to a host.</p> <p><b>Throws</b></p> <p><code>{Error}</code> If an attempt is made to set the device name while the session is connected.</p>
<code>getOIA()</code>	<p>Provides access to the <a href="#">OIA</a> instance for this session.</p> <p><b>Returns</b></p> <p><code>{OIA}</code> Associated with this session</p>

---

---

## 5.8.20 SessionType

Constants used to identify the type of host to which the connection is being made. See [Session](#) object.

Host Type	Description
IBM_3270	Indicates an IBM 3270 terminal session
IBM_5250	Indicates an IBM 5250 terminal session
VT	Indicates a VT session

---

## 5.8.21 StatusSet

You can use the StatusSet object to decode the OIA status. The StatusSet object returns values defined in the [OIAStatus](#) object and when used together, you can get status information from the OIA.

Method	Description
<code>contains(statusFlag)</code>	Determines if the set contains the specified status flag from OIAStatus constants. <b>Parameters</b> {Number} statusFlag status to check <b>Returns</b> {Boolean} True if the status flag is present in the set.
<code>isEmpty()</code>	Determines if the status set is empty. <b>Returns</b> {Boolean} True if the set is empty.
<code>size()</code>	Indicates the number of status flags in the set. <b>Returns</b> {Number} The status count
<code>toArray()</code>	Converts the internal status set to an array. <b>Returns</b> {Object []} Array of status flags in the set.
<code>toString()</code>	Converts the internal status set to a string. <b>Returns</b> {String} Space delimited names of status flags in the set.
<code>forEach(callback, thisArg)</code>	Function to iterate over each element in the status set. <b>Parameters</b> {forEachCallback} Callback to perform the specific operation. Called with the name of each status in the set.
<code>forEachCallback(string, thisArg)</code>	A user-provided callback function where you provide the behavior, to be used as the callback parameter to forEach. <b>Parameters</b> {String} String The name of a status in the status set. {Object} thisArg Optional pointer to a context object.

---

## 5.8.22 User Interface

The user interface object provides functions for interacting with the user, prompting for and displaying basic information. The UI object is made automatically available in your macro as the "ui" variable".

 **Note**

**Important:** All UI functions require the 'yield' keyword in front of them. This allows the macro to block execution until the conditions of the UI function have been met.

[parameter] denotes an optional parameter.

Method	Description
<pre>prompt(message, [defaultAnswer], [mask])</pre>	<p>Prompts the user for information in the user interface.</p> <p><b>Parameters</b></p> <p>{String} message title to display to the user . Default: blank String.  {String} defaultAnswer to use if user leaves it blank . Default: blank String  String  {Boolean} mask indicates whether to hide the prompt (as with a password)</p> <p><b>Returns</b></p> <p>{Promise} Fulfilled when the user closes the dialog window.  Returns the users input on "OK" or null on "Cancel".</p>
<pre>message([message])</pre>	<p>Display a message in the user interface.</p> <p><b>Parameters</b></p> <p>{String} message to display to the user. Default: blank String.</p> <p><b>Returns</b></p> <p>{Promise} Fulfilled when the user closes the message window.</p>

## 5.8.23 Wait

Use the wait object to wait for a particular session or screen state. For example, you can wait until the cursor is found at a particular location or text is present at a certain location before continuing with the macro execution.

Wait functions are often used in conjunction with asynchronous functions such as connect() and sendKeys().

**Note**

All functions take timeouts as an optional parameter and have a default time out value of 10 seconds (10000ms).

**Important:** All wait functions require the 'yield' keyword in front of them. This allows the macro to block execution until the conditions of the wait function are met.

[parameter] denotes an optional parameter.

Method	Description
<code>setDefaultTimeout(timeout)</code>	<p>Sets the default timeout value for all functions.</p> <p><b>Parameters</b></p> <p>{Number} default timeout to use for all wait functions in milliseconds.</p> <p><b>Returns</b></p> <p>{None}</p> <p><b>Throws</b></p> <p>{RangeError} If the specified timeout is less than zero.</p>
<code>forConnect([timeout])</code>	<p>Waits for a connect request to complete.</p> <p><b>Parameters</b></p> <p>{Number} in milliseconds.</p> <p><b>Returns</b></p> <p>{Promise} Fulfilled if the session is already connected or when connection occurs. Rejected if the wait times out.</p>
<code>forDisconnect([timeout])</code>	<p>Waits for a disconnect request to complete.</p> <p><b>Parameters</b></p> <p>{Number} timeout in milliseconds.</p> <p><b>Returns</b></p> <p>{Promise} Fulfilled if the session is already disconnected or when it finally disconnects. Rejected if the wait times out.</p>
<code>forFixedTime([timeout])</code>	<p>Waits unconditionally for fixed time. Time is in milliseconds (ms).</p> <p><b>Parameters</b></p> <p>{Number} timeout in milliseconds.</p> <p><b>Returns</b></p> <p>{Promise} Fulfilled after time elapses.</p>
<code>forScreenChange([timeout])</code>	<p>Waits for the host screen to change. This function returns when a screen update is detected. It makes no guarantees about the number of subsequent updates that may arrive before the screen is complete. Waiting repeatedly until the screen contents match some known stopping criteria is advisable.)</p> <p><b>Parameters</b></p> <p>{Number} timeout in milliseconds.</p> <p><b>Returns</b></p> <p>{Promise} Resolved if the screen change. Rejected if the wait times out.</p>
<code>forCursor(position, [timeout])</code>	<p>Waits for the cursor to arrive at the specified position.</p> <p><b>Parameters</b></p> <p>{Position} The position specifying the row and column</p> <p><b>Returns</b></p> <p>{Promise} Fulfilled if the cursor is already located or when it is finally located. Rejected if the wait times out.</p>
<code>forText(text, position, [timeout])</code>	<p>Wait for text located at a specific position on the screen</p> <p><b>Parameters</b></p> <p>{String} text to expect</p> <p>{Position} position specifying the row and column</p> <p>{Number} timeout in milliseconds</p> <p><b>Returns</b></p> <p>{Promise} Fulfilled if the text is already at the specified position or whenever it is located. Rejected if the wait times out.</p> <p><b>Throws</b></p> <p>{RangeError} if the position is not valid.</p>
<code>forHostPrompt(text, column, [timeout])</code>	<p>Waits for a command prompt located at a particular column on the screen.</p> <p><b>Parameters</b></p> <p>{String} text prompt to expect</p>

`{Position}` column where cursor is expected

`{Number}` timeout in milliseconds.

**Returns**

`{Promise}` Fulfilled if the conditions are already met or when the conditions are finally met. Rejected if the wait times out.

**Throws**

`{RangeError}` if the column is out of range.

---

`forHostSettle([settleTime],  
[timeout])`

**NOTE:** `wait.forHostSettle` should only be used when other more targeted wait functions are insufficient.

Monitors incoming screen data and resolves `settleTime` ms after the last update and the keyboard is unlocked. This function is useful when data arrives in multiple packets and you want to be sure the whole screen has been received before carrying on.

**Parameters**

`{Number}` time to wait after the last update to make sure more data doesn't arrive unexpectedly. The default is 200 milliseconds.

`{Number}` timeout in milliseconds.

**Returns**

`{Promise}` Fulfilled when the settle time has elapsed after receipt of the last screen update and the keyboard is unlocked.

---

## 5.9 Sample Macros

---

To help you create successful macros that take advantage of all the capabilities of the Macro Editor, these samples are available as a starting point.

### 5.9.1 Basic Host Interaction

---

This sample illustrates basic host interaction, including:

- Sending data to the host
- Waiting for screens to display
- Using the `yield` keyword to wait for asynchronous functions
- Reading text from the screen
- Displaying basic information to the user
- Handling error basics

All macros have the following objects available by default:

1. **session** - main entry point for access to the host. Can connect, disconnect and provides access to the PresentationSpace.

The PresentationSpace object obtained from the session represents the screen and provides many common capabilities, such as getting and setting the cursor location, sending data to the host, and reading from the screen.

2. **wait** - provides a simple way to wait for various host states before continuing to send more data or read from the screen.
3. **UI** - provides basic user interface capabilities. Display data to the user or prompt them for information.

```
// Create a new macro function
var macro = createMacro(function*(){
'use strict';

// All macros have the following objects available by default:
// 1. session - Main entry point for access to the host. Can connect, disconnect and provides access to the PresentationSpace.
//    The PresentationSpace object obtained from the session represents the screen and provides many common capabilities such as getting
and setting the
//    cursor location, sending data to the host and reading from the screen.
// 2. wait - Provides a simple way to wait for various host states before continuing to send more data or read from the screen.
// 3. ui - Provides basic User Interaction capabilities. Display data to the user or prompt them for information.

// Declare a variable for reading and displaying some screen data.
// As a best practice all variables should be declared near the top of a function.
var numberOfAccounts = 0;

// Start by obtaining the PresentationSpace object, which provides many common screen operations.
var ps = session.getPresentationSpace();

try {
// Can set and get the cursor location
ps.setCursorPosition(new Position(24, 2));

// Use the sendKeys function to send characters to the host
ps.sendKeys('cics');

// SendKeys is also used to send host keys such as PA and PF keys.
// See "Control Keys" in the documentation for all available options
ps.sendKeys(ControlKey.ENTER);

// Wait for the cursor to be at the correct position.
// The wait object provides various functions for waiting for certain states to occur
// so that you can proceed to either send more keys or read data from the screen.
yield wait.forCursor(new Position(24, 2));

// You can mix characters and control keys in one sendKeys call.
ps.sendKeys('data' + ControlKey.TAB + ControlKey.TAB + 'more data' + ControlKey.ENTER);

// The "yield" keyword must be used in front of all "wait" and "ui" function calls.
// It tells the browser to pause execution of the macro until the
// (asynchronous) wait function returns. Consult the documentation for which functions
// require the yield keyword.
yield wait.forCursor(new Position(10, 26));
ps.sendKeys('accounts' + ControlKey.ENTER);

// Can also wait for text to appear at certain areas on the screen
yield wait.forText('ACCOUNTS', new Position(3, 36));
ps.sendKeys('1' + ControlKey.ENTER);

// All wait functions will timeout if the criteria is not met within a time limit.
// Can increase timeouts with an optional parameter in the wait functions (in milliseconds)
// All timeouts are specified in milliseconds and the default value is 10 seconds (10000ms).
yield wait.forCursor(new Position(1, 1), 15000);
ps.sendKeys('A' + ControlKey.ENTER);

// PS provides the getText function for reading text from the screen
numberOfAccounts = ps.getText(new Position(12, 3), 5);

// Use the ui object to display some data from the screen
ui.message('Number of active accounts: ' + numberOfAccounts);

// The try / catch allows all errors to be caught and reported in a central location
} catch (error) {
// Again we use the ui object to display a message that an error occurred
yield ui.message('Error: ' + error.message);
}
//End Generated Macro
});

// Run the macro and return the results to the Macro Runner
// The return statement is required as the application leverages
// this to know if the macro succeeded and when it is finished
return macro();
```



## 5.9.2 User Interaction

This sample illustrates how to use the provided API methods to prompt the user for input or alert them with a message.

```

var macro = createMacro(function*(){
  'use strict';

  // The "ui" object provides functions for prompting the user for information and displaying information

  // Declare variables for later use
  var username;
  var password;
  var flavor;
  var scoops;

  //Begin Generated Macro
  var ps = session.getPresentationSpace();

  try {
    // Prompt the user to enter their name and store it in a variable.
    // Note that 'yield' keyword is needed to block execution while waiting for the user input.
    username = yield ui.prompt('Please enter your username');

    // Prompt the user to enter a value with a default provided to them.
    flavor = yield ui.prompt('What is your favorite flavor of ice cream?', 'Chocolate');

    // Prompt the user to enter private information by using the 'mask' option and the input field will be masked as they type.
    // If a parameter is not used, 'null' can be used to specify that it isn't to be used.
    // Here we illustrate that by specifying that we don't need to show a default value .
    password = yield ui.prompt('Please enter your password', null, true);

    // The prompt function returns null if the user clicks the 'Cancel' button instead of the 'OK' button.
    // One way to handle that case is to wrap the call in a try/catch block.
    scoops = yield ui.prompt('How many scoops would you like?');
    if (scoops === null) {
      // This will exit the macro.
      return;
      // Alternatively could throw an Error and have it be caught in the "catch" below
    }
    // Use the collected values to order our ice cream
    ps.sendKeys(username + ControlKey.TAB + password + ControlKey.ENTER);
    yield wait.forCursor(new Position(5, 1));
    ps.sendKeys(flavor + ControlKey.TAB + scoops + ControlKey.ENTER);

    // Display a message to the user. Using the 'yield' keyword in front of the call will block
    // further execution of the macro until the user clicks the 'OK' button.
    yield ui.message('Order successful. Enjoy your ' + scoops + ' scoops of ' + flavor + ' ice cream ' + username + '!');
  } catch (error) {
    // Here we use the ui object to display a message that an error occurred
    yield ui.message(error.message);
  }
  //End Generated Macro
});

return macro();

```

## 5.9.3 Paging Through Data

This sample illustrates how to page through a variable number of screens and process the data on each screen.

```

// Create a new macro function.
var macro = createMacro(function*(){
  'use strict';

  // Create variable(s) for later use
  var password;
  var accountNumber;
  var transactionCount = 0;
  var row = 0;

  // Obtain a reference to the PresentationSpace object.
  var ps = session.getPresentationSpace();

  try {
    // Enter Username and Password to log on to the application.
    yield wait.forCursor(new Position(19, 48));
    ps.sendKeys('bjones' + ControlKey.TAB);

    yield wait.forCursor(new Position(20, 48));
    password = yield ui.prompt('Password:', null, true);
    ps.sendKeys(password);
    ps.sendKeys(ControlKey.ENTER);

    // Enter an application command.
    yield wait.forCursor(new Position(20, 38));
    ps.sendKeys('4');
  }

```

```

ps.sendKeys(ControlKey.ENTER);

// Going to list transactions for an account.
yield wait.forCursor(new Position(13, 25));
ps.sendKeys('2');
// Input an account number. Hard coded here for simplicity.
yield wait.forCursor(new Position(15, 25));
accountNumber = yield ui.prompt('Account Number:', '167439459');
ps.sendKeys(accountNumber);
ps.sendKeys(ControlKey.ENTER);

// Wait until on account profile screen
yield wait.forText('ACCOUNT PROFILE', new Position(3, 33));

// Search for text that indicates the last page of record has been reached
while (ps.getText(new Position(22, 12), 9) !== 'LAST PAGE') {

    // While the last page of record has not been reached, go to the next page of records.
    ps.sendKeys(ControlKey.PF2);
    yield wait.forCursor(new Position(1, 1));

    // If the cursor position does not change between record screens, and there is no text
    // on the screen you can check to confirm a screen is updated, you may wait for a
    // fixed time period after an aid key is sent for the screen to settle.
    // For example:
    // yield wait.forFixedTime(1000);

    // For each of the rows, increment the count variable if it contains data.
    for (row = 5; row <= 21; row++) {

        // There are 2 columns on the screen. Check data on column 1.
        // In this example we know that if there is a space at a particular
        // position then there is a transaction.
        if (ps.getText(new Position(row, 8), 1) !== ' ') {
            transactionCount++;
        }
        // Check data on column 2.
        if (ps.getText(new Position(row, 49), 1) !== ' ') {
            transactionCount++;
        }
    }
}

// After going through all record pages, display the number of records in a message box.
yield ui.message('There are ' + transactionCount + ' records found for account ' + accountNumber + '.');

// Log out of the application
ps.sendKeys(ControlKey.PF13);
ps.sendKeys(ControlKey.PF12);

// The try / catch allows all errors to be caught and reported in a central location
} catch (error) {
    // Here we use the ui object to display a message that an error occurred
    yield ui.message(error.message);
}
});

// Here we run the macro and return the results to the Macro Runner
// The return statement is required as the application leverages
// this to know if the macro succeeded
return macro();

```

## 5.9.4 Invoking a Web Service

This sample illustrates how to make an AJAX / REST call directly from a macro to a web service. You can integrate data from your host application into the web service call or from the web service into your host application.

In this example, we are calling the Verastream Host Integrator (VHI) CICSAcctsDemo REST service. However, you can easily adapt the code to call any web service. You are not limited to VHI.

In the example, the call goes through a proxy configured in the session server (shown below) to avoid a “Same Origin Policy” complication. If you are using a web service that supports [Cross-origin Resource Sharing \(CORS\)](#) and are using a modern browser, the proxy is unnecessary.

Since the jQuery library is available in macros, you may use the \$.post() function directly to invoke REST services.

This example also demonstrates how to wrap a jQuery REST call in a new Promise. The promise returned from the custom function below allows "yield" to be used in the main macro code. This allows the main macro execution to wait until the service call is complete before continuing.

```

var macro = createMacro(function*() {
    'use strict';

```

```

// Create a few variables for later user
var username;
var password;
var accountNumber;
var accountDetails;

// Create a function that will make an AJAX / REST call to a VHI Web Service.
// Could be adjusted to call any web service, not just VHI.
// If not using CORS, the request will likely need to pass through a
// proxy on the session server. See sample notes for more information.
/**
 * Hand-coded helper function to encapsulate AJAX / REST parameters, invoke the
 * REST service and return the results inside a Promise.
 * @param {Number} acctNum to send to the REST query.
 * @param {String} username to access the REST service.
 * @param {String} password to access the REST service.
 * @return {Promise} containing $.post() results that are compatible with yield.
 */
var getAccountDetails = function (acctNum, username, password) {
  var url = "proxy1/model/CICSAcctsDemo/GetAccountDetail";
  var args = {"filters": {"AcctNum": acctNum}, "envVars": {"Username": username, "Password": password}};

  // Wrap a jQuery AJAX / HTTP POST call in a new Promise.
  // The promise being returned here allows the macro to yield / wait
  // for its completion.
  return Promise.resolve($.post(url, JSON.stringify(args)))
    .catch(function (error) {
      // Map errors that happen in the jQuery call to our Promise.
      throw new Error('REST API Error: ' + error.statusText);
    });
};

// Begin Generated Macro
var ps = session.getPresentationSpace();
try {
  // Could interact with the host here, log into a host app, etc...
  // Gather username and password
  username = yield ui.prompt('Username:');
  password = yield ui.prompt('Password:', null, true);
  accountNumber = yield ui.prompt('Account Number:');
  if (!username || !password || !accountNumber) {
    throw new Error('Username or password not specified');
  }

  // Invoke external REST service, and yields / waits for the call to complete.
  accountDetails = yield getAccountDetails(accountNumber, username, password);

  // We now have the data from our external service.
  // Can integrate the data into our local host app or simply display it to the user.
  // For this sample we simply display the resulting account details.
  if (accountDetails.result && accountDetails.result.length > 0) {
    yield ui.message(accountDetails.result[0].FirstName + ' $' + accountDetails.result[0].AcctBalance);
  } else {
    yield ui.message('No records found for account: ' + accountNumber);
  }
} catch (error) {
  // If an error occurred during the AJAX / REST call
  // or username / password gathering we will end up here.
  yield ui.message(error.message);
}
});

// Run our macro
return macro();

```

### Cross Origin Scripting Proxy Support

If you have web services that do not support CORS, any AJAX/REST calls will fail if they attempt to access a server other than the one where the Host Access for the Cloud application originated. This is a browser security feature.

The Host Access for the Cloud server provides a way to explicitly proxy to trusted remote servers.

- Open `..\<install_dir>\sessionserver\microservice\sessionserver\service.yml` for editing.
- In the `env` section add:

```
name: zfe.proxy.mappings
value: proxy-path=proxy-to-address
```

Where `proxy-path` refers to the desired url-mapping and `proxy-to-address` refers to the URL where the call will be proxied.

- In this example:

```
name: zfe.proxy.mappings
value: proxy1=http://remote-vhi-server:9680/vhi-rs/
```

Calls made to `<server:port>/proxy1` will be proxied to `http://remote-vhi-server:9680/vhi-rs/`

- Multiple proxy mappings can be specified using a comma to separate the individual proxy mappings
- Keep in mind that even when a REST server supports CORS headers, some older browsers may not, so this example may still be relevant.

### Hint

Your `service.yml` file may be replaced whenever you redeploy Host Access for the Cloud. Always back up your files.

## 5.9.5 Working with Data Cells and Attributes

This macro illustrates how to use `DataCells` and `AttributeSet` to inspect a given row/column on the screen for text and attributes. In this sample you can see:

- How to get a collection of `DataCells` for a given position and length.
- How to iterate through `DataCells` to build up a text string
- How, for comparison, you can also do a similar thing using `getText()`.
- And finally, how to work with attributes, get a string listing, or determine whether specific ones are set at a given screen location.

```
var macro = createMacro(function*() {
  'use strict';

  // Obtain the PresentationSpace for interacting with the host
  var ps = session.getPresentationSpace();

  // Declare variables for later use
  var cells;
  var text;
  var attrs;

  // Set the default timeout for "wait" functions
  wait.setDefaultTimeout(10000);

  // Sample macro for working with DataCells and Attributes
  try {
    yield wait.forCursor(new Position(24, 2));

    // Get DataCells from the presentation space.
    // Row 19, col 3 is the prompt, 35 characters long
    // "Choose from the following commands:"
    cells = ps.getDataCells({row:19, col:3}, 35);
    text = '';

    // You can display text using getText
    yield ui.message("Screen text: " + ps.getText({row:19, col:3}, 35));

    // Or you can assemble the text from the DataCells at each position
    for(var index = 0; index < cells.length; index++) {
      text = text.concat(cells[index].getChar());
    }
  }
  // And display the text
```

```

yield ui.message("Cells text: " + text);

// Get the attributes for the first DataCell (cell[0])
attrs = cells[0].getAttributes();

// Display whether we have any attributes on the data cell
yield ui.message("Attribute set is empty: " + attrs.isEmpty());

// Display how many attributes are set
yield ui.message("Number of attributes: " + attrs.size());

// Display which attributes are set
yield ui.message("Attributes: " + attrs.toString());

// Now display whether the high intensity attribute is set
yield ui.message("Is high intensity: " +
    attrs.contains(Attribute.HIGH_INTENSITY));

// Now display whether the underline attribute is set
yield ui.message("Is underline: " +
    attrs.contains(Attribute.UNDERLINE));

// Now display whether alphanumeric, intensified and pen-detectable attributes are set
yield ui.message("Is alphanumeric, intensified and pen-detectable: " +
    attrs.containsAll([Attribute.ALPHA_NUMERIC, Attribute.HIGH_INTENSITY, Attribute.PEN_DETECTABLE]));

// Now display whether underline, intensified and pen-detectable attributes are set
yield ui.message("Is underline, intensified and pen-detectable: " +
    attrs.containsAll([Attribute.UNDERLINE, Attribute.HIGH_INTENSITY, Attribute.PEN_DETECTABLE]));
} catch (error) {
    yield ui.message(error);
}
}
//End Generated Macro
});

// Run the macro
return macro();

```

## 5.9.6 Using Fields and Field Lists

This macro sample illustrates how to use common functions to interact with the fields in the Macro API. For example, how to get field text, view field information, and use `field.setText` as an alternative to `sendKeys` to interact with the host.

### Note

Due to browser considerations, `ui.message` collapses strings of spaces down to a single space. The spaces are preserved in the actual JavaScript.

```

var macro = createMacro(function*() {
    'use strict';

    // Obtain the PresentationSpace for interacting with the host
    var ps = session.getPresentationSpace();

    // Declare variables for later use
    var fields;
    var field;
    var searchString = 'z/VM';

    // Set the default timeout for "wait" functions
    wait.setDefaultTimeout(10000);

    // Sample macro for working with FieldList and Fields
    try {
        yield wait.forCursor(new Position(24, 2));

        // Get the field list.
        fields = ps.getFields();

        // Run through the entire list of fields and display the field info.
        for(var index = 0; index < fields.size(); index++) {
            field = fields.get(index);

            yield ui.message("Field " + index + " info: " + field.toString());
        }

        yield ui.message("Now, find a field containing the text '" + searchString + "'");
        field = fields.findField(new Position(1, 1), searchString);

        if(field !== null) {
            yield ui.message("Found field info: " + field.toString());
            yield ui.message("Found field foreground is green? " + (Color.GREEN === field.getForegroundColor()));
            yield ui.message("Found field background is default? " + (Color.BLANK_UNSPECIFIED === field.getBackgroundColor()));
        }
    }
}

```

```

// Now, find command field and modify it.
field = fields.findField(new Position(23, 80));
if(field != null) {
    field.setText("cics");
}

yield ui.message("Click to send 'cics' to host.");
ps.sendKeys(ControlKey.ENTER);

// Wait for new screen; get new fields.
yield wait.forCursor(new Position(10, 26));
fields = ps.getFields();

// Find user field and set it.
field = fields.findField(new Position(10, 24));
if(field != null) {
    field.setText("myusername");
}

// Find password field and set it.
field = fields.findField(new Position(11, 24));
if(field != null) {
    field.setText("mypassword");
}

yield ui.message("Click to send login to host.");
ps.sendKeys(ControlKey.ENTER);

// Wait for new screen; get new fields.
yield wait.forCursor(new Position(1, 1));
fields = ps.getFields();

// Find command field and set logoff command.
field = fields.findField(new Position(24, 45));
if(field != null) {
    field.setText("cesf logoff");
}

yield ui.message("Click to send logoff to host.");
ps.sendKeys(ControlKey.ENTER);

} catch (error) {
    yield ui.message(error);
}
//End Generated Macro
});

// Run the macro
return macro();

```

## 5.9.7 Automatic Sign-On Macro for Mainframes

In this example the Autosignon object is used to create a macro that uses the credentials associated with a user to obtain a pass ticket from the Digital Certificate Access Server (DCAS).

```

var macro = createMacro(function*() {
    'use strict';

    // Obtain the PresentationSpace for interacting with the host
    var ps = session.getPresentationSpace();

    // Variable for login pass ticket
    var passTicket;

    // Login application ID
    var appId = 'CICSV41A';

    // Set the default timeout for "wait" functions
    wait.setDefaultTimeout(10000);

    // Begin Generated Macro
    try {
        yield wait.forCursor(new Position(24, 2));

        // Obtain a pass ticket from DCAS.
        passTicket = yield autoSignon.getPassTicket(appId);

        ps.sendKeys('cics');
        ps.sendKeys(ControlKey.ENTER);

        yield wait.forCursor(new Position(10, 26));

        // Replace generated username with sendUserName(passTicket) ...
        yield autoSignon.sendUserName(passTicket);

        // ps.sendKeys('bvtst01' + ControlKey.TAB + ControlKey.TAB);
        ps.sendKeys(ControlKey.TAB + ControlKey.TAB);

        yield wait.forCursor(new Position(11, 26));

        // Replace generated password with sendPassword(passTicket) ...
        yield autoSignon.sendPassword(passTicket);
    }
});

```

```

    // var userInput3 = yield ui.prompt('Password:', '', true);
    // if (userInput3 === null) {
    //     // throw new Error('Password not provided');
    // }
    // ps.sendKeys(userInput3);
    ps.sendKeys(ControlKey.ENTER);

    yield wait.forCursor(new Position(1, 1));
    yield ui.message('Logged in. Log me off. ');
    ps.sendKeys('cesf logoff');
    ps.sendKeys(ControlKey.ENTER);
} catch (error) {
    yield ui.message(error);
}
//End Generated Macro
});

// Run the macro
return macro();

```

## 5.9.8 Using File Transfer (IND\$File)

This series of sample macros demonstrate how to use the File Transfer API to retrieve a list of files, download a file, and upload a file to a 3270 host.

### Note

You must be logged in and at a ready prompt before running these macros.

- [List files](#)
- [Download file](#)
- [Upload file](#)

### List files

This macro demonstrates how to use the File Transfer API to retrieve a list of files on a 3270 host using IND\$File transfer. The IND\$File transfer object is retrieved from the file transfer factory and then used to obtain an array of HostFile objects from either TSO or CMS.

```

var macro = createMacro(function*() {
    'use strict';

    try {
        var fileTransfer = fileTransferFactory.getInd$File();
        var hostFiles = yield fileTransfer.getHostFileListing();

        yield ui.message('Found ' + hostFiles.length + ' files');
        if (hostFiles.length > 0) {
            var firstFile = hostFiles[0];
            var msg1 = 'The catalog name is ' + firstFile.getParent() + ' . ';
            var msg2 = 'The first file is ' + firstFile.getName();
            yield ui.message(msg1 + msg2);
        }
    } catch (error) {
        yield ui.message(error);
    }
});

// Run the macro
return macro();

```

### Download file

This macro shows how to use the File Transfer API to download a file from a 3270 host using IND\$File transfer. The IND\$File transfer object is retrieved from the file transfer factory. In this example, the transfer method is set to ASCII to demonstrate use of the setTransferOptions function.

The sample macro downloads the first file returned from a call to getHostFileListing by creating a download URI with a call to the getDownloadUrl function. The macro can be used in either a CMS or

TSO environment but the choice must be specified on the first line or the code modified slightly for the intended system.

```

var hostEnvironment = 'CMS'; // 'TSO'
// Construct file path, ie catalog/file.name or catalog/partition/file
function getPath (fileNode) {
  var prefix = fileNode.getParent() ? fileNode.getParent() + '/' : '';
  return prefix + fileNode.getName();
}

var macro = createMacro(function*() {
  'use strict';

  try {
    var fileTransfer = fileTransferFactory.getInd$File();

    // The transferMethod options are 'binary' and 'ascii'
    fileTransfer.setTransferOptions({transferMethod: 'ascii'});

    // This demo retrieves the first file returned in the listing
    var hostFiles = yield fileTransfer.getHostFileListing();
    var firstHostFile = hostFiles[0];

    if (hostEnvironment === 'CMS') {
      yield wait.forText('Ready', new Position(1,1), 5000);
    }

    // Download
    // If you already know the path of the file you want, just pass that to getDownloadURL()
    var downloadUrl = fileTransfer.getDownloadURL(getPath(firstHostFile));

    // This changes the browser location. You may experience different results on different browsers
    window.location = downloadUrl;

    // If you want to read the file contents into a variable instead of downloading
    // it, you can use jQuery
    // var fileContents = yield $.get(downloadUrl);

  } catch (error) {
    yield ui.message(error);
  }
});

// Run the macro
return macro();

```

## Upload file

This macro illustrates how to use the File Transfer API to upload a file to a 3270 host using IND\$File transfer. The sample macro prompts the user to choose a file from the local file system by triggering the browser's file selection dialog. It then retrieves the current catalog on TSO or drive identifier on CMS by calling `getHostFileListing`. Finally, the `sendFile` function is called to deliver the selected local file to the host.

The macro can be used in either a CMS or TSO environment but the choice should be specified on the first line. In this example, the transfer method is set to `ascii`; you may want to change this to `binary`.

```

var hostEnvironment = 'CMS'; // 'TSO'
// Open the browser's file chooser dialog programmatically
function promptForFileToUpload () {
  return new Promise(function (resolve, reject) {
    // We are not notified if the user cancels the file chooser dialog so reject after 30 seconds
    var timerId = setTimeout(reject.bind(null, 'Timed out waiting for file selection'), 30000);
    var fileSelector = document.createElement('input');
    fileSelector.setAttribute('type', 'file');
    fileSelector.onchange = function (evt) {
      var file = evt.target.files[0];
      clearTimeout(timerId);
      resolve(file);
    };
    fileSelector.click();
  });
}

var macro = createMacro(function*() {
  'use strict';

  try {
    var fileTransfer = fileTransferFactory.getInd$File();

    // The transferMethod options are 'binary' and 'ascii'
    fileTransfer.setTransferOptions({transferMethod: 'ascii'});

    var localFile = yield promptForFileToUpload();

```



```
// Retrieve the current catalog name and append the selected file name to it
var hostFiles = yield fileTransfer.getHostFileListing();
var destination = hostFiles[0].getParent() + '/' + localFile.name;

if (hostEnvironment === 'CMS') {
  yield wait.forText('Ready', new Position(1,1), 5000);
}

var result = yield fileTransfer.sendFile(localFile, destination);

} catch (error) {
  yield ui.message(error);
}
});

// Run the macro
return macro();
```

## 5.10 Run Macro on Event

---

Use the Macro panel to select which macros to run and when to run them.

- Run macro on startup - Choose a macro to run automatically when the session is opened.
- Run macro on connect - Choose a macro to run automatically when the session connects to the host.
- Run macro on disconnect - Choose a macro to run automatically when the session disconnects from the host.

### More information

- [Creating Macros](#)
- [Using the Macro API](#)
- [Sample Macros](#)

## 5.11 Printing

---

There are various printing options available for 3270, 5250, and UTS hosts. You can take screen captures, print a selected screen, and enable and configure host printing capabilities:

The settings available to you regarding page setup and orientation are dependent on your browser options.

### 5.11.1 Capture a screen


---

Use the screen capture feature to capture multiple screens then save them as a file for printing or sharing. This option is available to all users once the administrator selects it using **User Preferences**.

1. Navigate to the screen you want to capture.

2.



Click  to capture the screen. The counter displays the number of captures you've taken. Each capture will print to a separate page.

3. Click **Save** to browse to the location where you want to save the capture. Your browser determines how the save option functions. For example, in Chrome, depending on your browser settings, the file will be saved in the download file or you will see a **Save As** dialog to select a location to save the captured file.
4. To append your newly saved screens to an existing screen capture file, click **Append and save**. When you print the appended file, each screen capture is printed to a separate page.
5. You can clear the captures whenever you want by clicking **Clear**.

### 5.11.2 Print a screen

---

The print screen option prints the contents of the terminal screen. It does not print the toolbar or other display information.

1. Navigate to the screen you want to print.
2. Click **Print Screen** on the tool bar.
3. Use your browser's print dialog to select the printer and page setup options.

### 5.11.3 Host printing

---

This feature is available to 3270, 5250, and UTS host sessions. You can create one or more printer sessions and associate them with the current terminal session. Each printer session is bound to a Device ID on the host system and any subsequent print jobs sent to that Device ID will be directed to the Host Access for the Cloud web client.

The host session builds a PDF file that contains the file to print and sends it to the web client. After receiving the file, the web client downloads the file following your browser's configured download options. Different browsers provide different options for handling downloaded files. When the PDF file is received, you can direct it to any printer that you have access to.

#### Note

An administrator can provide end users with the ability to print by setting the **Host Printing User Preferences** option.

## To configure host printing

1. From a host session, click **Settings** on the tool bar to open the left navigation panel.
2. In the left panel, click **Print**.
3. Click **Add** to open the configuration dialog box. Customize your printer session by using the settings on each tab: Connection settings, Page Setup settings, and Advanced settings.
4. Click **Save** to return to your session. The settings take effect when the session is reopened.

### Host Printing topics

- [Connection settings: 3270, 5250, UTS](#)
- [Page Setup settings](#)
- [Advanced settings](#)
- [To print your host printer session](#)

### Connection settings

By default, printer sessions are available from the printer icon on the terminal session tool bar. If you do not want end users to have access to this printer session, clear **Enable this printer session** on the Connection tab.

These settings differ depending on your host type.

- [3270 connection settings](#)
- [5250 connection settings](#)
- [UTS connection settings](#)

---

#### 3270 CONNECTION SETTINGS

Setting	Description
Name	Provide an easily identifiable name for your printer session. Required.
Protocol	Select the protocol to use. The options are: <ul style="list-style-type: none"> <li>• <b>TN3270E</b> - TN3270E of Telnet Extended, is for users of TCP/IP software who connect to their IBM mainframe through a Telnet gateway that implements RFC 1647.</li> <li>• <b>TN3287</b> - TN3287 is for users of TCP/IP software who connect to their IBM mainframe through a Telnet gateway that implements RFC 1646.</li> </ul>
Device ID	Specify whether you want to use a Device ID, prompt for Device Name or, if you select TN3270E, a TN Association, to link the terminal session with the print session. Required. Select one: <ul style="list-style-type: none"> <li>• <b>Specify Device ID</b> - Specify the Device ID to use when the printer session connects to the host.</li> <li>• <b>Use TN Association</b> - (TN3270E) If you choose to use a TN association, Host Access for the Cloud uses the device name specified in the connection settings to link the 3270 and 3287 session together. TN Association is only available if you select TN3270E as the protocol.</li> <li>• <b>Prompt the user</b> - When the printer session connects, the user is prompted to supply the Device ID for the printer session.</li> </ul>

## 5250 CONNECTION SETTINGS

Setting	Description
Name	Provide an easily identifiable name for your printer session. Required.
Device ID	Specify whether you want to use a Device ID or prompt for Device ID: <ul style="list-style-type: none"> <li>Specify Device ID - Specify the Device ID to use when the printer session connects to the host.</li> <li>Prompt the user - When the printer session connects, the user is prompted to supply the Device ID for the printer session.</li> </ul>

## UTS CONNECTION SETTINGS

Setting	Description
Name	Provide an easily identifiable name for your printer session. Required.
Protocol	The choice of DEMAND or MAPPER protocols depends on the type of UTS session you create. UTS session types are determined by the values you supply for the TSAP and Application options on the Connection panel. For example, if you enter values that create a UTS MAPPER or DEMAND session, you should select MAPPER or DEMAND as the protocol. Specify which protocol to use: <ul style="list-style-type: none"> <li><b>MAPPER</b> - You can choose to specify the Device ID to use when the printer session connects to the host or prompt the user to supply the Device ID for the printer session, then continue configuring the session.</li> <li><b>DEMAND</b> - After providing a name for the session, you can continue configuring the session using Page Setup and Advanced tabs.</li> </ul>

## Page Setup settings

The Page Setup tab contains setting options for paper size and orientation, along with dimensions, margins, and scaling values.

Setting	Description
Paper size	Select the size of paper used by the printer.
Orientation	Choose from three modes: <b>Portrait</b> (vertical), <b>Landscape</b> (horizontal) or <b>Auto</b> , which is the default. With Auto selected, the printer evaluates the print job and uses the most appropriate format.
Units of measurement	Select the unit of measurement you want to use for page margins and page sizes. The values are inches or millimeters.
Dimensions	Enter the number of rows and columns to display per printed page. 60 is the default row value and the column value defaults to 80.
Margins	Sets the left, right, top, and bottom page margins.
Scaling	Sets the horizontal and vertical scaling for printed output. Increase the percentage to increase the horizontal or vertical space used by the printout.

## Advanced settings

Choose when you want the PDF file to be downloaded.

- **Automatically** - (default) The PDF is downloaded automatically when the print job is complete. When this option is selected, the Inactivity timeout setting is not available.
- **Manually** - Once a print job commences, you can initiate a download anytime by locating the print job in the download list available from the Print icon on the tool bar and clicking Flush. The print job is aggregated into a single PDF file and downloaded.
- **After inactivity timeout** - Using this option you can print multiple print jobs, have them aggregated into a single PDF, and then automatically downloaded when you specify.

If you decide on a value greater than 0 (for example, 5 seconds) any print jobs assigned to a printer that arrive within 5 seconds of each other will be appended to the same PDF. After 5 seconds and no remaining print jobs, the PDF is downloaded.

If you specify 0 for the inactivity timeout, each print job is downloaded immediately upon completion. You can always interrupt a print job by clicking **Flush**.

## To print your host printer session

When the terminal session opens, you can now:

1. Select the printer session you want to use. All print sessions associated with the opened



terminal session are available to you. Click on the tool bar to see a list.

2. The host session receives the print data from the host and builds a PDF file to print. A link to this file is sent to the web client indicating it is available for download.

You can monitor the various print jobs using the tool bar page counter or the counter associated with separate printers in the print drop down list.

The page counter on the tool bar reflects the total number of pages either being actively printed or complete but waiting for the file to download from the server. You can trigger a download by selecting **Flush** from the printer list.

The page counter attached to printers in the printer drop down list displays the same value but on a per printer basis. The sum of these separate print jobs is reflected in the tool bar count. The count is cleared once the print jobs are downloaded.

3. After the PDF file becomes available, the file either begins downloading or waits for you to trigger a download using the **Flush** option, depending on the options you configured.

If necessary, due to an overlong running print job or some other issue, you can flush your current print job. The **Flush** option is available from the list of printer sessions accessed from the printers icon on the tool bar. When you flush a print job, whatever has been accumulated so far is printed and processing of print data continues.

## 6. Developing

---

### 6.1 Developing

---

Host Access for the Cloud has a collection of APIs and libraries that help you develop efficient client/server and Web applications that integrate host data into various development environments.

You can also extend the web client without affecting the installed files. This ability provides you with a wide range of options to tailor the web client to your own needs.

- [Using the Java SDK](#) you can use the provided Java API to enhance the presentation of host data using server side events.
- [Using the Connector for Windows](#) you can interact with host sessions in your .NET application or within Visual Basic for Applications using the API and samples provided.
- [Using the JavaScript API](#) you can embed the web client in your own web site.
- [Extending the Web Client](#) you can enhance and broaden the scope of the web client using custom code, such as CSS or JavaScript.

[HACloud API Documentation](#)

## 6.2 Using the Java SDK

---

Working with [server side events](#) and the Host Access for the Cloud SDK you can supply procedural Java code that can extend and improve the presentation of host data. To help you create server side events, Host Access for the Cloud has an SDK and samples that can provide you with a starting point.

The Javadocs are available in your installation directory ( `<install-dir>\sessionserver\jdk\java\javadocs\index.html` ) as well as online here: [HACloud Javadocs](#).

1. Make the Host Access for the Cloud SDK available to your development environment. The SDK is available at `install-dir\sessionserver\jdk`.
2. Write the Java code necessary to accomplish the task and compile the code into a Java class within a JAR (Java Archive) file.
3. Copy the JAR file to `<install-dir>\sessionserver\microservices\sessionserver\extensions\server` and restart the session server.

If you have more than one session server on which you want the event to run, you must copy the JAR file to this location on each server.

4. Add the session you want to associate with the event in the Administrative Console.
5. As you configure the session in the web client, open the Customization panel and type the fully qualified class name to the event.
6. Launch the session and test the event.

### 6.2.1 Examples and documentation

---

To access the SDK for direct viewing and to import to your IDE:

1. Navigate to `<install-dir>\sessionserver\jdk\java`
2. In the SDK directory, access:
  - `\javadoc` - This directory contains javadoc files for direct viewing.
  - `\samples` - This directory contains Java sources for direct viewing.
  - `\zfe-sdk.jar` - The JAR file contains the Java classes to import into your IDE.
  - `\zfe-sdk-javadoc.jar` - The JAR file contains JavaDoc files to import into your IDE.



## 6.3 Using the Connector for Windows

---

Host Access for the Cloud Connector for Windows is a separate installation which you can find on the Micro Focus download site. Using the Connector for Windows, you can interact with host sessions in your .NET application or within Visual Basic for Applications.

The API documentation is available in your installation directory ( `<install dir>\sessionserver\sdk\csharp\apidocs\index.html` ) as well as online here: [HACloud Connector for Windows](#).

Here are a few things to keep in mind as you prepare to install:

- Two install platforms are available: a 32-bit version and a 64-bit version. Depending on which one you install, the default base install path will be `C:\Program Files (x86)\Micro Focus\HACloud\Connector for Windows` OR `C:\Program Files\Micro Focus\HACloud\Connector for Windows`.
- The installation platform you choose also determines the solution platform in which you can develop. For example: If you have installed the 32-bit version of Microsoft Office® and want to use Visual Basic for Applications with the connector, then you must install the 32-bit version of the Host Access for the Cloud Connector for Windows.
- API documentation is available here: `<install dir>\sessionserver\sdk\csharp\apidocs\index.html`
- .NET 4.5.2 is required.

### 6.3.1 Examples and connector documentation

---

Documentation is available to reference from your IDE. There are also samples to help you take advantage of the connector. Both are located here:

1. Navigate to the install directory. In a default install, either `C:\Program Files (x86)\Micro Focus\HACloud\Connector for Windows` OR `C:\Program Files\Micro Focus\HACloud\Connector for Windows` depending on your platform.
2. In the Connector for Windows directory you will find:
  - `MicroFocus.ZFE.Connector.dll` - a .NET Framework assembly to reference in your C# or .NET project.
  - `MicroFocus.ZFE.Connector.tlb` - a Type Library to use in your COM or Visual Basic for Applications project.
  - `\help` - this directory contains information which will aid in using the connector.
  - `\samples` - this directory contains the code samples that provide a starting point for developing your own applications.

### 6.3.2 Using the connector with Microsoft Visual Studio

---

If you are using Microsoft Visual Studio to develop applications, keep these things in mind:

- When using Microsoft Visual Studio with Connector for Windows, make sure your solution platform is set to either x86 or x64, depending on your installation. Because of the native components used within the Connector for Windows SDK, the Any CPU platform is not supported. Use the Configuration Manager for your Visual Studio Solution to create a platform for x86 or x64.
- When adding a reference to the Connector for Windows library, Visual Studio may set the Copy Local reference property to True. This should be set to False so that the library and its dependencies are executed from the SDK install directory.

## 6.4 Using the JavaScript API

---

Using JavaScript in a browser you can embed the web client in a web page. Your end users, by accessing a common web page, can interact with the web client and connect to the host application providing the ability to:

- Programmatically interact with host sessions.
- Run it “headlessly”, meaning you can access all its functionality without having a visible interface embedded in the web page.

There are getting started and other tutorials available for your use. The API documentation, along with the tutorials, is available online here: [HACloud JavaScript API](#) and in `<install dir>\sessionserver\sdk\javascript`



### Note

The SameSite flag must be adjusted when using the JavaScript API. See [Setting the SameSite Attribute](#).

## 6.5 Extending the Web Client

---

You can update, modify, and customize the presentation of the web client by using your own HTML, CSS, or JavaScript from within the browser.

You can take advantage of extensions to make visual changes to the web client and to customize the application. The web client hosts your custom HTML or CSS code, making it easy to modify and support.

Learn more about:

- [Adding an extension](#)
- [Extension sample](#)

### 6.5.1 Adding an extension

---

Before proceeding keep in mind that although Host Access for the Cloud provides the ability to plan and use custom code, the code itself must be supported by the team that produced it.

#### Warning

During a product upgrade extensions are disabled. This means that, after an upgrade, you must verify that the product is working as expected without extensions, and then re-enable the extensions using the steps to add custom code.

When you add extensions to the web client, the modifications are visible to all your users and apply to all sessions.

#### To add an extension

1. Open `<install_dir>/sessionserver/microservices/sessionserver/service.yml`.
2. Append `extensions-enabled` to the existing value of the `SPRING_PROFILES_ACTIVE` property. Use a comma to separate the values.

For example:

```
env:
- name: SPRING_PROFILES_ACTIVE
  value: tls,extensions-enabled
```

3. Restart the session server.
4. Create `<install_dir>/sessionserver/microservices/sessionserver/extensions/client/index.html` to act as your entry point. This is where you add any HTML, CSS, or JavaScript (including references to external scripts).

#### Making extensions available without client authentication

Files within the `/client` directory are protected using the level of authentication you selected in MSS.

#### To share files without requiring authentication:

Create `<install_dir>/sessionserver/microservices/sessionserver/extensions/public/` Place your code in that directory, calling it using the URL `/public/*`

## 6.5.2 Extension sample

In this example, once extensions are enabled (see Step 2 above), you can add some custom CSS and JavaScript code to change the menu label font color and print text to the JavaScript console.

You will create three files; `custom.css`, `custom.js`, and `index.html`

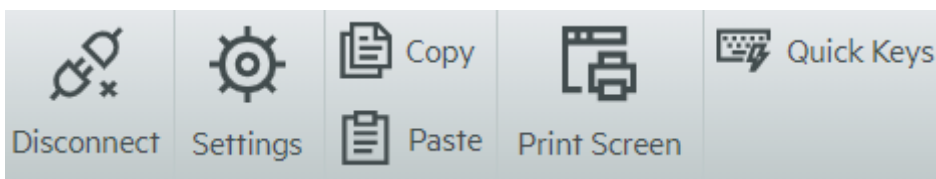
### Step 1

Locate `index.html`, which you created in step 4 above. This is where you will be placing your extension files, creating an entry point:

```
<!-- Define the link to the external style sheet -->
<link href="client/custom.css" rel="stylesheet">
<!-- Define the external JavaScript file -->
<script src="client/custom.js"></script>
```

### Step 2

Change the default black menu labels to orange:



Create `custom.css` to change the color to orange:

```
/* Change link text to Orange */
a span {
  color: #ff5d28;
}
```

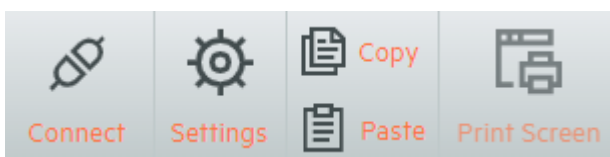
### Step 3

Create `custom.js` to send text to the JavaScript console:

```
//Print message to the JavaScript console
console.log('Hello World!');
```

### Step 4

When the files are in place, `<install_dir>/sessionserver/microservices/sessionserver/extensions/client/index.html`, the results should look like this:



And the "Hello World" text is visible in the JavaScript console:



### More Information:

[HACloud API Documentation](#)

## 7. Technical References

---

### 7.1 Technical References

---

In this section you can find information on specific issues that you may encounter. In the Micro Focus Technical Support Handbook you will find information about how to get technical support for your product, access to our online resources, and how to contact and work with our worldwide technical support organization.

## 7.2 Monitoring Session Servers using Prometheus and Grafana

---

You can monitor Host Access for the Cloud session servers using Prometheus and Grafana. Both of these tools are free, open source, and can be run in Docker containers which makes for easy deployment. Each session server provides a Prometheus endpoint that exposes metrics about that server. Prometheus can be configured to scrape data from this endpoint and store the metrics on an ongoing basis, even from multiple session servers. Grafana then provides a dashboard to query and visualize this data, with very little setup.

### Prerequisites:

You must have Docker and Docker Compose installed.

### Steps:

1. Create a Docker Compose file (.yml) that contains both Grafana and Prometheus images.
2. Link Prometheus to your session server Prometheus endpoint.
3. Configure your Grafana data source to communicate with Prometheus and import the pre-configured dashboards.
4. Configure the Grafana dashboards.
5. Access Grafana.

### 7.2.1 Step 1. Create a Docker Compose file

---

Create a `docker-compose.yml` file containing Grafana and Prometheus images.

```
version: "3.1"
services:
  grafana:
    build: grafana
    ports:
      - '3000:3000'
  prometheus:
    image: prom/prometheus:v2.6.1
    ports:
      - '9090:9090'
    volumes:
      - ./config/prometheus.yml:/etc/prometheus/prometheus.yml
      - ./prometheus:/prometheus
    networks:
      monitoring:
        aliases:
          - prometheus
networks:
  monitoring:
```

### 7.2.2 Step 2. Link Prometheus to your HACloud Prometheus endpoint

---

To link Prometheus to your endpoint, generate a `prometheus.yml` file.

- In our example, the `prometheus.yml` file is saved in the config directory.
- This example config allows you to scrape the Prometheus endpoint using either HTTP or HTTPS (TLS).
- If TLS is disabled on the session server, remove `tls_config` and change the scheme to `http` in the example config.
- Configure the session-server-hostname.

!!!note Due to Docker networking, this must be the actual IP or hostname of your session server host computer. This IP can typically be obtained using `ifconfig/ipconfig`.

- Adjust ports if needed.

*Example config/prometheus.yml*

```
scrape_configs:
- job_name: ' HACloud Session Server with TLS'
  scrape_interval: 15s
  scheme: https
  tls_config:
    insecure_skip_verify: true
  metrics_path: actuator/prometheus
  static_configs:
    - targets: ['session-server-hostname:7443']
```

### 7.2.3 Step 3. Configure communication between Prometheus and the data source

Communication can be configured within the Grafana Docker image between the local instance of Prometheus and your Grafana data source. Pre-loaded dashboards are also available to you at startup.

#### Example grafana/Dockerfile

```
FROM grafana/grafana:5.3.2
ADD ./provisioning /etc/grafana/provisioning
ADD ./config.ini /etc/grafana/config.ini
ADD ./dashboards /var/lib/grafana/dashboards
```

#### Example grafana/config.ini

```
[paths]
provisioning = /etc/grafana/provisioning
```

#### Example grafana/provisioning/datasources/all.yml

```
datasources:
- name: 'Prometheus'
  type: 'prometheus'
  access: 'browser'
  url: 'http://localhost:9090'
  is_default: true
  editable: false
```

#### Example grafana/provisioning/dashboards/all.yml

```
- name: 'default'
  org_id: 1
  folder: ''
  type: 'file'
  options:
    folder: '/var/lib/grafana/dashboards'
```

### 7.2.4 Step 4. Configure the Grafana dashboards

There is a sample JSON file to help you get started configuring your Grafana dashboards.

To have your Docker container load the dashboard on startup:

- Locate `HACloudSessionservers.json` in the `hacloud/utilities/grafana` directory.
- Copy `HACloudSessionservers.json` to your `grafana/dashboards` directory.

### 7.2.5 Step 5. Access Grafana

- Start the Docker container with the command `docker-compose up -d`.
- Verify Prometheus targets are successfully scraping the session servers using `http://localhost:9090/targets`.
- Access Grafana using `http://localhost:3000`.
- Both the user name and password = `admin`. The user name and password can be configured using Docker environment variables.
- Use the command `docker-compose down` to stop the Docker container.

## 7.3 Running the Session Server Service as a Dedicated User with Reduced Privileges

---

In the interest of security, you should run the session server service as a dedicated user with reduced privileges.

 **Note**

You must repeat these steps after any product upgrades.

**To do this on a Windows platform:**

1. After first installing the product, create a standard user.
2. Give the user full control over the following directories:
  - `<install_dir>\sessionserver\logs`
  - `<install_dir>\sessionserver\tmp`
3. From the Windows Services dialog box, stop the Micro Focus Host Access for the Cloud Session Server service.
4. Open the Properties of the Host Access for the Cloud Session Server service.
5. Open the Log On tab and select **This account**.
6. Enter the user name and password of the user who is going to run the service.
7. Start the session server service.

**To do this on Unix / Linux:**

1. Follow the steps above but adjust them accordingly for your environment and distribution.

**More information**

[Running the Micro Focus MSS Server service as a dedicated user](#)



## 7.4 Setting the SameSite Attribute

---

To help prevent cross-site request forgery attacks, the default SameSite attribute on the session server cookie has been updated from **None** (less restrictive) to **Lax** (more restrictive).

With the attribute set to **Lax**, the session server cookie will not be sent on cross-site requests as is often the case with the JavaScript SDK and in SAML authentication.

This change affects two areas of HACloud:

- The JavaScript SDK and
- SAML authentication behind a load balancer

In these cases you'll need to adjust the attribute value to **None** by:

1. Opening `container.properties` in a text editor. The default location for this file is `./sessionserver/conf/`.
2. Adding the following line to `container.properties` :  
`samesite.cookie.attribute=None`
3. Restarting the session server.

## 7.5 Modifying the Size Limit on Upload File Transfer Operations

---

There is a 50MB file size limit on file transfer upload operations. To modify the file size limit, set `spring.servlet.multipart.maxfilesize` and `spring.servlet.multipart.maxrequestsize` in `HACloud/sessionserver/microservices/sessionserver/service.yml`, and restart the session server.

For example:

```
- name: spring.servlet.multipart.maxfilesize
  value: "100MB"
- name: spring.servlet.multipart.maxrequestsize
  value: "100MB"
```

## 7.6 Configuring the MSS callback address

---

MSS provides a callback address to the session server each time it creates or edits a session.

By default, it uses the address specified in `management.server.url`.

If the MSS server is behind a proxy and the session server cannot reach the address:

- Set the `management.server.callback.address` property in each MSS `container.properties` file to an address that the session server, for a specific MSS, can reach.
- Restart the server for the new property values to take effect.

## 7.7 Copying Sessions between Management and Security Servers

You can copy and convert Reflection for the Web sessions and make them available to another Management and Security Server (MSS) and Host Access for the Cloud.

### Note

In the following procedure the Management and Security Server you are copying sessions from is the **source**, and the Management and Security Server you are copying to is the **destination**.

To copy sessions from the source server to the destination server follow these steps:

1. Stop the destination MSS server, if necessary.
2. On both source and destination MSS servers, open SessionDS.xml, located:
  - On Windows: `C:\ProgramData\Micro Focus\MSS\MSSData`
  - On Linux: `/var/opt/microfocus/mss/mssdata`
3. In the source XML file, locate the OBJECT\_ARRAY element.
4. Still in the source XML file, under OBJECT\_ARRAY, locate and copy the Reflection for the Web child Session elements.
5. Open the destination XML file and paste them under the destination file's OBJECT\_ARRAY element.
6. Still in the destination file, locate the OBJECT\_ARRAY size attribute that corresponds to the number of sessions. Increase that value by the number of session elements you added. For example, if you pasted six Session elements in the destination file and the existing OBJECT\_ARRAY size attribute value is 4; increase the value by six. The size attribute should now be ten. And you should now have 10 Session elements listed under the OBJECT\_ARRAY element.
7. Session names must be unique. Check the destination file for duplicate session names. You can find session names in the Session child element, SessionName.
8. Copy the configuration files for every session added to SessionDS.xml from the source to the destination server. The names of the configuration files are located under the Session element in the child element, configuration. The files themselves are located:
  - On Windows: `C:\ProgramData\Micro Focus\MSS\MSSData\deploy\dyncfgs`
  - On Linux: `/var/opt/microfocus/mss/mssdata/deploy/dyncfgs`
9. If you stopped the destination MSS server, restart it. Open the Administrative Console. You should see all your copied Reflection for the Web sessions in the Manage Sessions list.
10. The next step is to save the Reflection for the Web session as a Host Access for the Cloud session. In Manage Sessions, right-click the session you want to export. Session types are identified by an icon in the Type column.
11. See Export a Reflection for the Web session for information on saving a Reflection for the Web session to a Host Access for the Cloud session in the Administrative Console.

## 7.8 Changing Ports

---

See [Ports](#) for a list of the default ports used by Host Access for the Cloud.

To change the default ports:

Component	Instructions
Host Access for the Cloud session server	Open <code>sessionserver/microservices/sessionserver/service.yml</code> to modify: <code>-name : SERVER_PORT</code> and value: <code>"7443"</code>
Management and Security Server	The SSL port MSS uses to make an HTTPS connection is set to 443 by default. If you need to change the port number, start the Management Server. This creates the default <code>PropertyDS.xml</code> file. Then, open <code>PropertyDS.xml</code> in the <code>MssData</code> directory. Change the value from 443 to the appropriate port number in the section below, and then restart the Management Server. <code>&lt;CORE_PROPERTY NAME="sslport"&gt; &lt;STRING&gt;443&lt;/STRING&gt;</code>

## 7.9 Starting and Stopping Services Automatically

All server components are installed as services and can be configured to start during installation.

If you are running on Linux platforms, follow these steps to set the session server to start automatically when your system first boots up.

Create a file called `sessionserver` containing the following and using your installation directory:

```
#!/bin/sh
#
#This script manages the service needed to run the session server
#chkconfig:235 19 08
#description: Manage the Host Access for the Cloud session server

###BEGIN INIT INFO
# Provides:          sessionserver
# Required-Start:    $all
# Required-Stop:     $all
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Description:       Start the Host Access for the Cloud Session Server
### END INIT INFO

INSTALL_DIR=<enter installation directory>
BIN_DIR=$INSTALL_DIR/sessionserver/bin
case "$1" in
start)
echo "Starting Host Access for the Cloud Session Server"
$BIN_DIR/server start

RETVAL=0
;;
stop)
echo "Stopping Host Access for the Cloud Session Server"
$BIN_DIR/server stop

RETVAL=0
;;
status) echo "Current Host Access for the Cloud Session Server status"
$BIN_DIR/server status

RETVAL=0
;;
restart) echo "Restart Host Access for the Cloud Session Server"
$BIN_DIR/server restart

RETVAL=0
;;
*)
echo "Usage: $0 (start|stop|status|restart)"

RETVAL=1
;;
esac
exit $RETVAL
```

Then complete the relevant steps.

On Linux:

1. Copy the file to the `/etc/init.d` directory.
2. Set the file permission. Run `chmod` using the value 755. For example, `chmod 755 sessionserver`
3. Run `chkconfig` to add the initialization script. For example, `/sbin/chkconfig --add sessionserver`

## 7.10 Adjusting the URL Path for the Session Server

---

You can adjust the URL path used to access the session server.

For example, you can change `https://myserver:7443/` to `https://myserver.com:7443/hacloud/`

1. Open `<install_dir>/sessionserver/microservices/sessionserver/service.yml`.
2. Add the following entry (maintain formatting) where *path* is replaced with the value you want to use.

```
-name: SERVER_SERVLET_CONTEXTPATH  
value: "/<path>"
```

3. Restart the session server.
4. Access your session server at `https://<session server>:7443/<path specified>/`

## 7.11 Configuring User Names when Using Anonymous Access Control

Users need access to their macros, user configurations, and other personalized settings whether they are authenticated through Management and Security Server or not. These settings are collectively referred to as User Preferences.

When MSS is configured for authentication, using LDAP or SAML for example, a username is determined when a user logs in. The user's settings are centrally saved in MSS using that username for all future logins.

However when the MSS Authentication Method is set to None, also known as anonymous mode, no unique username is available for the system to identify that particular user when they return in the future. In this configuration, all users share the same settings. If one user changes a setting, that setting will be changed for all other users.

Because that may not always be the desired behavior, Host Access for the Cloud supports a number of ways that, as an administrator, you can configure a unique identifier for each user so their customized settings can be stored and retrieved during future logins.

### Note

These configuration modifications do not alter the security considerations of using Management and Security Server in anonymous mode.

### 7.11.1 Configuration options

There are four different configuration options you can choose from when configuring user name identifiers. You must restart the session server before any changes take effect.

- To use an HTTP request cookie value as the user name

Add the following lines to `<session-server>/conf/container.properties` :

```
zfe.principal.name.provider=com.microfocus.zfe.webclient.security.mss.CookieKeyAnonymousPrincipalNameProvider
zfe.principal.name.identifier=<the-cookie-key-to-be-used>
```

- To use an HTTP request header value as the user name

Add the following lines to: `<session-server>/conf/container.properties` :

```
zfe.principal.name.provider=com.microfocus.zfe.webclient.security.mss.HeaderKeyAnonymousPrincipalNameProvider
zfe.principal.name.identifier=<the-header-key-to-be-used>
```

- To use an HTTP request URL parameter as the user name

Add the following lines to: `<session-server>/conf/container.properties`

```
zfe.principal.name.provider=com.microfocus.zfe.webclient.security.mss.UrlParameterAnonymousPrincipalNameProvider
zfe.principal.name.identifier=<the-url-parameter-key-to-be-used>
```

- To use the client IP address as the user name

Add the following line to: `<session-server>/conf/container.properties`

```
zfe.principal.name.provider=com.microfocus.zfe.webclient.security.mss.RemoteAddrAnonymousPrincipalNameProvider
```



## 7.11.2 Troubleshooting the configuration

---

If any of your users experience problems when connecting to a Host Access for the Cloud web application after you have made the configuration changes, check the following:

- Users experience a *503 Service Unavailable* message when connecting to a Host Access for the Cloud web application. First check the log file ( `<session-server>/logs/sessionserver.log` ), then:
  - If the log file contains this message: *Unable to create AnonymousPrincipalNameProvider instance for class...*, then the `zfe.principal.name.provider` property is probably mis-typed. Check the spelling and letter case to remedy this issue.
  - If the log file contains this message: *zfe.principal.name.identifier is not defined*, then the property is missing. Ensure the property is defined to remedy this issue.
- Users are unable to properly authenticate.

Users should receive an error message indicating the initial HTTP request to the Host Access for the Cloud web application did not contain the required informat

## 7.12 Accessing Host Access for the Cloud using the IIS Reverse Proxy

---

This note describes how to use the IIS Reverse Proxy with Host Access for the Cloud. In order to comply with Common Criteria security requirements, it is necessary to place the Host Access for the Cloud server behind a proxy in this manner.

### Prerequisites

- Internet Information Services (IIS) 8.0 or later is required.
- The IIS WebSockets protocol must be enabled. See [IIS 8.0 WebSocket Protocol Support](#) for information on how to enable this protocol.
- IIS Application Request Routing (ARR) 3.0 or later is required.
- The IIS URL Rewrite module must be installed.

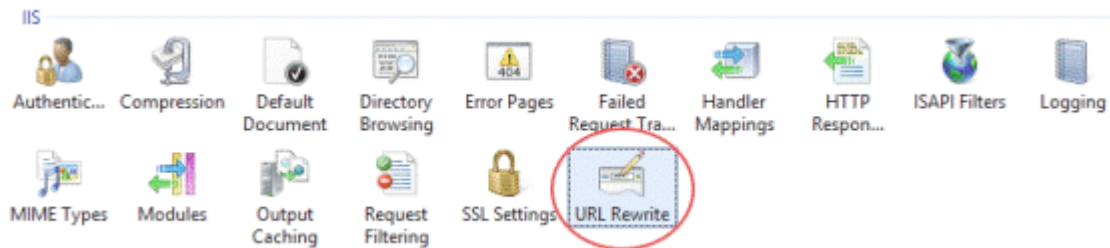
### 7.12.1 Configure the IIS Reverse Proxy for HACloud

---

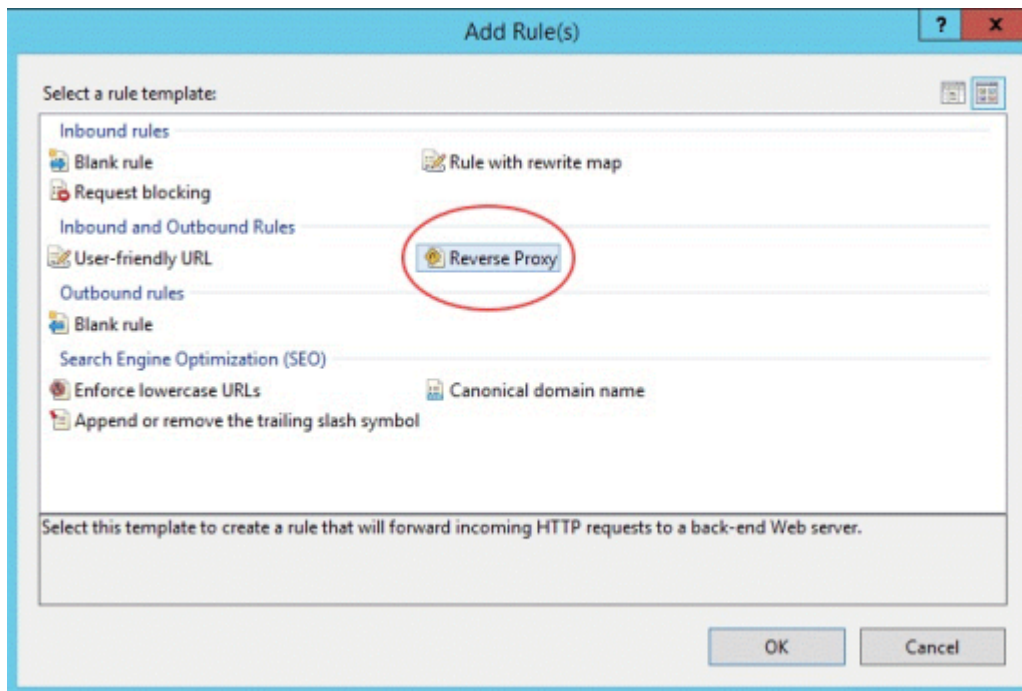
This example illustrates configuring an IIS server with the IP address of 192.168.1.1 to proxy connections to the Host Access for the Cloud session server at `http://10.10.10.1:7070`.

#### Configuring IIS

1. Launch the Internet Information Services (IIS) Manager, navigate to the web site you want to use, and open the URL Rewrite feature.



2. Choose the Add Rule(s) action and add the Reverse Proxy rule.



3. For the inbound rule, enter the Host Access for the Cloud server's IP address or host name and port. For example, if the session server is on the same machine as IIS and is using its default port, enter `localhost:7443`.
4. Check the outbound rule Rewrite the domain names... and enter the host name or IP address of the IIS server in the To: box.
5. Click OK to create the new Reverse Proxy Rule.

### Configuring HACloud

In order to proxy connections, the IIS URL Rewrite module must inspect and rewrite the web pages and WebSocket connections that pass through the proxy. For rewriting to succeed, these items must be sent in an uncompressed form. Note that, if configured, compression will still occur from the IIS server to the client's browser. The session server must also be configured to allow WebSocket connections to originate from the proxy.

1. Open `container.properties` in a text editor. The default location for this file is: `/sessionserver/conf`.
2. Add the following lines to `container.properties`:

```
websocket.compression.enable=false
server.compression.enabled=false
websocket.allowed.origins=http://<IIS server name or IP address>. For example: 192.168.1.1.
```

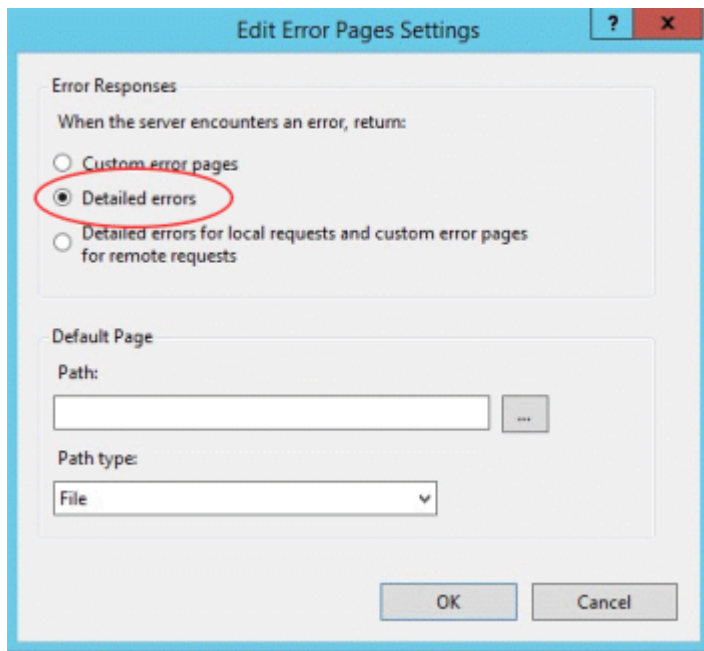
Save changes to the file. The Allowed Origins property is a comma-delimited list of URLs. If web clients will be connecting to your website using an HTTPS connection, adjust the URL accordingly. If both secure and non-secure connections will be used, use both URLs as the value:

`websocket.allowed.origins=http://192.168.1.1,https://192.168.1.1`. To avoid errors, make sure that all possible address formats are included in the Allowed Origins list.

3. Restart the web site and restart the session server and test the proxy by connecting to: `http(s)://192.168.1.1`.

### Troubleshooting

If you receive web server errors, enabling detailed errors may help diagnose the problem. In the IIS Manager, open the Error Pages feature and check Detailed errors:



Typically errors in the 5XX range are caused by issues with compression being enabled or mistakes in the Allowed Origins value.

If the IIS proxy will be connecting to the session server with HTTPS, then the certificate used with the session server must be trusted by the IIS server. If the session server is using a self-signed certificate, this certificate must be added to the Windows trust store. If the session server is using a signed certificate, then the signer must be a trusted CA.

## 7.13 Using the IIS Reverse Proxy with Host Access for the Cloud

---

To comply with Common Criteria security requirements it may be necessary to place the session server behind a proxy. Before configuring, read [Accessing Host Access for the Cloud using the IIS Reverse Proxy](#) for prerequisites and configuration instructions.

### Note

To comply with Common Criteria security requirements it may be necessary to place the session server behind a proxy by following the instructions in [Accessing Host Access for the Cloud using the IIS Reverse Proxy](#).

To proxy Host Access for the Cloud through IIS, when using IIS single sign-on, you need to set an additional property in the same `container.properties` file:

```
servletengine.iis.url=<url>
```

The value takes the same form as the URL above, but uses the Host Access for the Cloud address. For example: `http://server/`. It is not necessary to use the short host name form in this URL.

After you have completed this configuration, you choose this authentication option in Management and Security Server Administrative Console | Assign Access. See the [Administrative Console online help](#) for descriptions of the configuration options.

### More Information:

- [Configure Single Sign-on through IIS](#)

## 7.14 Improving Connection Times on Non-Windows Platforms

---

To improve connection times on non-Windows platforms, follow these steps after installing the Host Access for the Cloud session server, particularly when the system is virtualized or otherwise headless:

1. Stop the session server service.
2. Open the `<installation folder>/sessionserver/conf/container.conf` file in a text editor.
3. Locate this line, and edit as follows:

```
#wrapper.java.additional.x=-Djava.security.egd=file:///dev/urandom
```

- Remove the # to uncomment the line.
- Replace x with `<n+1>`, where `<n>` is the highest number noted in the other `wrapper.java.additional.<n>` lines.
- Save the file.

4. Restart the session server service.

## 7.15 Configuring Cross-Origin Resource Sharing (CORS)

---

As a security measure, modern web browsers restrict the types of interactions that are permitted between distinct web sites. This can cause problems when attempting cross-site integration, for example when embedding the HACloud web client into another website, such as a portal. CORS is a standard mechanism that you can use to specify that the browser permit access from one site to another site.

You can configure the HACloud session server to include the required CORS HTTP header when it responds to the web requests by updating the `service.yml` file.

1. Open `<install directory>/sessionserver/microservices/sessionserver/service.yml`.
2. To the file, add

```
-name: CORS_ALLOWED_ORIGINS
value: "https://integration-server1.com"
```

3. Restart the session server.

You can set this value to a comma-delimited list of allowed origins or use `*` to allow access from all origins (Allowing this kind of open access may be a security risk). If you use the wild card option(`*`), be aware that web browsers impose additional restrictions, such as limited Cookie access. For more information, see [Cross-Origin Resource Sharing \(CORS\) - HTTP/MDN](#).

## 7.16 Adjusting HTTP Session Timeout

---

The default timeout value for an inactive user session is 30 minutes. This means that when a user closes their browser without logging out first, their user session and any open host sessions will be cleaned up after 30 minutes. The minimum timeout value allowed is 600 seconds (10 minutes). You can configure this setting on the server.

1. Open `<install directory>/sessionserver/microservices/sessionserver/service.yml`.
2. Adjust the session timeout value in the `env` section of the file:

```
- name: server.servlet.session.timeout
  value: <desired-time-in-seconds>
```

 **Note**

The indentation formatting is important.

3. Restart the server.



## 7.17 Enabling FIPS Level Security

---

The Federal Information Processing Standards (FIPS) 140-2 validated cryptographic modules are used by the US federal government as a security regulation standard. Host Access for the Cloud supports this standard and you can easily enable FIPS mode by editing a file in the session server.

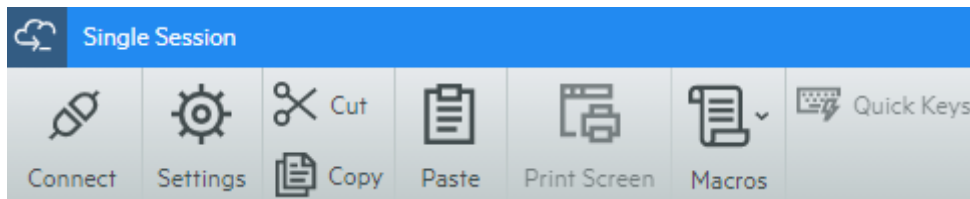
- Open `<install_directory>\sessionserver\microservice\sessionserver\service.yml`.
- Add the flag `-Dcom.attachmate.integration.container.FIPS.enabled=true` to the Java command - `start-command`.
- Restart the server.
- To ensure that FIPS mode is enabled, open `<install_directory>\sessionserver\logs\sessionserver.log` and verify that FIPS mode is set to true; `FIPS mode: true`.

## 7.18 Using Single Session Mode

---

You can use single session mode and provide URLs to particular sessions that are launched using the name parameter, (for example a direct link on a company portal page). To enable the launch of a single session use the query parameter `singleSession`. You can use this parameter on its own to just launch the web client in single session mode, for example, `http://<sessionserver>:7443/?singleSession`, or it can be used in conjunction with a named session parameter to launch a particular named session in single session mode: `http://<sessionserver>:7443/?singleSession&name=HumanResources`. The order of the parameters does not matter.

When your users access a single session, they cannot switch between open sessions and cannot open new sessions. A new session will not be launched if the specified session already exists when the user opens the link.



If you want all sessions on the session server to run in single session mode:

- Open `<install-dir>/sessionserver/conf/container.properties`
- Add `webclient.singleSession=true` to the file.

## 7.19 Known Issues

---

These issues have been identified in previous releases and are known issues.

[Browser issues](#)

[Host specific issues](#)

[Installation issues](#)

### 7.19.1 Browser issues

---

The following notes are specific to different web browsers.

- [Recommended browsers](#)
- [Key mapping issues with different browsers](#)

#### **Recommended browsers**

It is highly recommended that you use Google Chrome or Mozilla Firefox. While Host Access for the Cloud supports Microsoft Internet Explorer (IE) 11, there are known performance issues with Internet Explorer's JavaScript engine that may negatively affect the end user experience with Host Access for the Cloud.

These issues have been identified and have remedies, however the easiest solution is to use a different browser.

- Internet Explorer Unable to Play Recorded Macros

When using certain older versions of Microsoft Internet Explorer (IE) web browser with Host Access for the Cloud, attempts to playback macros may fail with an error. The error message reads: *Macro Error: Error transpiling macro code: TypeError: unknown: Circular reference in value argument not supported.*

This is a problem with this version of Internet Explorer and JavaScript. It may be possible to avoid this error if you delete the createMacro() function and replace it using JavaScript Promises (for example, then() ).

Because this issue is specific to early versions of Internet Explorer, the easiest solution to this problem is to use a different browser (Chrome or Firefox) or a more recent version of Internet Explorer. You can successfully play back macros using Internet Explorer version 11.0.9600.18161, update version 11.0.27. Run Windows Update to update Internet Explorer.

- HTTPS connections between Apple iOS mobile devices and the session server

Host Access for the Cloud users cannot connect to a session server over HTTPS from their Apple iPad when using a self-signed certificate. If feasible, the quickest solution is to use HTTP instead of HTTPS.

If HTTPS is needed, you have the following options:

- Obtain a valid certificate signed by a trusted CA and install it on the session server.
- Find an alternate browser that will accept the self-signed certificate. See [Browser and operating system support](#) for a list of supported browsers.
- Leverage a custom certificate authority:
  - Create a custom CA, CA root certificate, and a server certificate signed by that CA's root certificate.
  - Install the server certificate on the session server.
  - Install the custom CA root certificate on the iPad by means of a profile. The iPad should now accept the server certificate as it was signed by a "trusted CA".

For a list of CAs trusted by Apple iOS, see [Lists of available trusted root certificates in iOS](#).

- Internet Explorer Displays Blank Screens

When using the Microsoft Internet Explorer (IE) web browser with Host Access for the Cloud or Management and Security Server (MSS), you may see a blank screen instead of the expected session.

When using Microsoft Internet Explorer to access Host Access for the Cloud sessions or Management & Security Server, you may experience issues such as the following:

Host Access for the Cloud renders properly for some URLs and not others (a blank screen is displayed). The behavior varies depending on whether the session is using an IP address, short hostname, or fully-qualified name.

In MSS, you can't create or open a Host Access for the Cloud session unless that session is on the same server as MSS. A blank screen displays where you expect to see the session.

#### Explanation

This issue is specific to the way Internet Explorer toggles various settings depending on its interpretation of website security. The settings in question are Compatibility View and Third-party Cookies. Depending on what "zone" Internet Explorer determines your web site to be in, these settings need to be either enabled or disabled. Internet Explorer bases its determination on the site URL. For example, if the server name in the URL does not contain dots (for example, <http://mycorporateserver/mss/AdminStart.html>), Internet Explorer assumes the address belongs in the Local Intranet zone. If it does, the site is assigned to the Internet zone.

#### Local Internet Zone

- Compatibility View enabled (not desired)
- Third-party Cookies enabled (desired)

### Internet Zone

- Compatibility View disabled (desired)
- Third-party Cookies disabled (not desired)

While it is possible for a website to override Compatibility View by specifying Document Mode with an X-UA-Compatible meta HTML tag, and Host Access for the Cloud does use that particular mode, MSS does not. Thus, if a Host Access for the Cloud server and a Management and Security Server are both in the Local Intranet zone (with default Compatibility View enabled), it is likely that Host Access for the Cloud would still perform correctly, but MSS would not.

### Solution

To use Internet Explorer 10 or 11 with Host Access for the Cloud and MSS servers, you need:

You need to determine what zone your web site is in and then make the necessary adjustments to the Internet Explorer settings. Because Internet Explorer can be configured in so many different ways depending on your situation, it is hard to provide one solution for successfully using Internet Explorer with Host Access for the Cloud and MSS. These are some possible configurations to follow:

- If both Host Access for the Cloud and MSS are in the Internet zone, manually add the Host Access for the Cloud server to the Local Intranet or Trusted Sites zone (Internet Options > Security > Local intranet > Sites). Use fully qualified host names or IP addresses.
- If both servers are in the Internet zone, change the default behavior for that zone and enable Third-party Cookies (Internet Options > Privacy > Advanced > Override automatic cookie handling).
- If both servers are in the Local Intranet zone, change the default behavior for that zone and disable Compatibility View (Tools > Compatibility View settings).

### Key mapping issues with different browsers

Certain keys on a numeric keypad and some browser-specific keys cannot be mapped. For example, in Chrome, Ctrl+n and Ctrl+w cannot be mapped.

## 7.19.2 Host specific issues

---

The following are issues that are specific to different host types.

### Displaying the Euro character

If the EURO character does not display correctly on the terminal screen, talk to your system administrator to make sure the host character set for the session is setup correctly. By default, Host Access for the Cloud uses a character set which does not support the Euro character (€). To display the Euro character, change the character set to one that supports the Euro character.

### Issues encountered with VT hosts

Type	Description
Performance issues	<ul style="list-style-type: none"> <li>• Heavy text output, such as form “Is-IR” may cause slow performance</li> <li>• Scrolling regions may appear slow or choppy</li> <li>• Cursor movement may be slow or choppy</li> <li>• Internet Explorer is particularly slow, and performance degrades further when used for rows and columns.</li> </ul>
Character sets	<ul style="list-style-type: none"> <li>• Graphical characters and some character sets are not supported.</li> <li>• Some non-English characters may cause the terminal display to freeze.</li> </ul>
Other VT issues	<ul style="list-style-type: none"> <li>• Insert/delete column (DECIC, DE CDC) may fail.</li> <li>• VT400 will not recognize DECSCL.</li> </ul>

### Field outlines in 3270 sessions

The 3270 attributes for field outlines are not fully supported. Host Access for the Cloud currently supports underline and overline; however, left vertical line, right vertical line, and combinations of the four line types are not yet supported.

## 7.19.3 Installation issues

The [Installing and Upgrading](#) topics include a troubleshooting section which may help diagnose and remedy specific installation issues.

### Setting a TEMP directory for the installation program

The installer requires a writable temp directory. If the default temp directory is not suitable, the installer can be run with an alternate temp directory.

- Windows

If the default temp directory is not writable, set the environment variables TMP or TEMP to an alternate location temporarily while running the installer. Restore the variables when the installation is complete.

- Linux/Unix

The environment variable INSTALL4J\_TEMP determines the base directory that the installer uses for self-extraction. As the installer extracts files and launches Java to perform other tasks, the Java temp location ( /tmp) is used.

To run the Linux installers with an alternate temp directory:


- Define the variable INSTALL4J\_TEMP, specifying the value as the desired temp location.
- Create the temp directory specified for the installer. The installer requires that the directory already exist.
- Add the command line switch `-J-Djava.io.tmpdir={tempdir}` when launching the installer. For example:

```
abcd@linux:~$ INSTALL4J_TEMP=/home/abcd/i4jtemp
abcd@linux:~$ export INSTALL4J_TEMP
abcd@linux:~$ sudo ./hacloud-2.4.2.12345-linux-x64.sh -J-Djava.io.tmpdir=/home/abcd/i4jtemp
```

### Chained HACloud and MSS installations

- On Windows, a chained installation of HACloud and MSS will need no other adjustments, if you temporarily set the TMP or TEMP environment variables described above.

- On Linux/Unix - You cannot run a chained installer on this platform; run them separately, each with administrative permissions, the `INSTALL4J_TEMP` variable set and with the `-J-Djava.io.tmpdir` switch.

 **Note**

If both MSS and HACloud are being installed “unchained”, MSS must be installed first, then HACloud.

### Setting a TEMP directory for the product

HACloud uses an internal temp directory that should be suitable in all cases. However, this directory can be changed if necessary by editing the `container.conf` file.

This location is configurable:

1. Open `<installation folder>/sessionserver/conf/container.conf` in a text editor.
2. Edit the `wrapper.java.additional` property to specify the new location. If the path contains spaces, enclose it in quotes on Windows or use the appropriate syntax for Linux/Unix platforms. For example, `wrapper.java.additional.9=-Djava.io.tmpdir=../tmp`
3. If needed, you can set an additional property to delete the temp directory when the server shuts down.
4. Restart the server.



## 8. Legal Notice

---

© Copyright 2022 Micro Focus or one of its affiliates

The only warranties for products and services of Micro Focus and its affiliates and licensors ("Micro Focus") are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Micro Focus shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

Contains Confidential Information. Except as specifically indicated otherwise, a valid license is required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

For information about legal notices, trademarks, disclaimers, warranties, export and other use restrictions, U.S. Government rights, patent policy, and FIPS compliance, see <https://www.microfocus.com/legal>.