
Host Access for the Cloud Web Client

3.1.1

Table of contents

Le damos la bienvenida al cliente Web de Host Access for the Cloud	4
Ajustes de conexión	5
Ajustes de conexión	5
Parámetros de conexión comunes	5
Parámetros de conexión 3270 y 5250	11
Parámetros de conexión VT	13
Parámetros de conexión UTS	16
Parámetros de conexión T27	18
Parámetros de conexión ALC	19
Trabajar con sesiones	21
Utilizar Teclas Rápidas	21
Edición de la pantalla	22
Salida de la sesión	23
Extra!	24
Crear Macros	24
Objetos de Macro API	30
Ejemplos de Macros	63
Ejecutar macro en evento	0
Parámetros de pantalla	0
Asignación de color	0
Configurar zonas activas	0
Configurar dimensiones de pantalla para hosts VT, UTS y T27	0
Configurar opciones de cursor	0
Configurar opciones de fuente	0
Configurar opciones de búfer de desplazamiento hacia atrás VT	0
Configurar opciones de teclado	0
Configuración del terminal	0

Configurar otras opciones de visualización	0
Asignar teclas	0
Asignar teclas	0
Asignación de teclado de host	0
Transferir archivos	0
IND\$FILE	0
AS/400	0
FTP	0
Transferencias por lotes	0
Especificar las opciones de edición	0
Impresión	0
Capturar una pantalla	0
Imprimir una pantalla	0
Impresión de host	0
Personalización de las sesiones de host	0
Gestión de las Preferencias de usuario	0
Información legal	0

Le damos la bienvenida al cliente Web de Host Access for the Cloud

El cliente Web Rocket® Host Access for the Cloud proporciona acceso HTML5 basado en navegador a las aplicaciones de host 3270, 5250, VT, UTS, ALC y T27. El producto Host Access for the Cloud elimina la necesidad de utilizar el escritorio; no hay software que distribuir, parches que aplicar ni configuraciones que establecer. Con él puede proveer acceso a usuarios a todas sus aplicaciones de host, independientemente de cuál sea la plataforma utilizada.

Ajustes de conexión

Ajustes de conexión

Hay parámetros de conexión comunes que se aplican a todos los tipos de host.

[Parámetros de conexión comunes](#)

Hay parámetros adicionales que son específicos del tipo de host.

[Parámetros de 3270 y 5250](#)

[T27](#)

[UTS](#)

[VT](#)

[ALC](#)

Parámetros de conexión comunes

Estas opciones son comunes para todos los tipos de host soportados.

- **Conectar al iniciar**

De forma predeterminada, las sesiones se configuran para conectarse automáticamente al host cuando usted crea o abre una sesión. También puede configurar una sesión para que no se conecte automáticamente al host. Elija No para conectarse al host manualmente.

- **Reconectar cuando el host finaliza la conexión**

Si se establece en Sí, Host Access for the Cloud intentará volver a conectarse tan pronto como finalice la conexión del host.

- **Protocolo**

Seleccione el protocolo que desee utilizar para comunicar con el host de la lista desplegable. Para establecer una conexión de host, el cliente Web y el equipo host deben utilizar el mismo protocolo de red. Los valores disponibles dependen del host al que se esté conectando. Son los siguientes:

Protocolo	Descripción
-----------	-------------

TN3270	TN3270 es una forma del protocolo Telnet que es un conjunto de especificaciones para la comunicación general entre el escritorio y los sistemas de host. Utiliza TCP/IP como transporte entre las computadoras de escritorio y los mainframes IBM.
TN3270E	TN3270E o Telnet Extendido es para usuarios de TCP/IP que se conectan a su mainframe IBM mediante un gateway Telnet que implementa RFC 1647. El protocolo TN3270E le permite especificar el nombre del dispositivo de conexión (conocido también como nombre LU) y ofrece soporte para la clave ATTN, la clave SYSREQ y la gestión de la respuesta SNA. Si intenta utilizar Telnet Extendido para conectarse a un gateway que no soporta este protocolo, se utilizará el estándar TN3270 en su lugar.
TN5250	TN5250 es una forma del protocolo Telnet que es un conjunto de especificaciones para la comunicación general entre el escritorio y los sistemas de host. Utiliza TCP/IP como transporte entre las computadoras de escritorio y las computadoras AS/400.
Secure Shell (VT)	<p>Puede configurar las conexiones SSH cuando se necesite una comunicación segura y cifrada entre un host VT de confianza y la computadora a través de una red no segura. Las conexiones SSH garantizan que tanto el usuario cliente como la computadora del host se autenticuen, así como el cifrado de todos los datos. Hay dos opciones de autenticación disponibles:</p> <p>Teclado interactivo - Puede utilizar este método de autenticación para implementar distintos tipos de mecanismos de autenticación. Cualquier método de autenticación soportado que requiera sólo la entrada del usuario se puede realizar con el Teclado interactivo.</p> <p>Contraseña - Esta opción le pide al cliente una contraseña al host después de haber establecido la conexión host. La contraseña se envía al host a través del canal cifrado.</p>
Telnet (VT)	Telnet es un protocolo de la suite TCP/IP de protocolos abiertos. Como protocolo de secuencia de caracteres, Telnet transmite de carácter en carácter las entradas del usuario desde aplicaciones de modo de caracteres a través de la red hasta el host, donde se procesan y se devuelven a través de la red.
INT1 (UTS)	Ofrece acceso a hosts Unisys 1100/1200 que utilizan el protocolo de red TCP/IP.

TCPA (T27)	Utilice este protocolo para conectarse a hosts de la serie Unisys ClearPath NX/LX o de la serie A. La autenticación TCPA es el proceso de verificar la información del inicio de sesión del usuario. Cuando está configurada correctamente, puede solicitar una credencial de seguridad de su servidor de credenciales de aplicación y transmitir la credencial de vuelta al servidor. Si la credencial es válida, su aplicación iniciará sesión; no tendrá que introducir un ID de usuario o una contraseña. Si la credencial no es válida, se le pedirá utilizar un ID de usuario y una contraseña.
MATIP (ALC)	Mapping of Airline Traffic Over Internet Protocol (MATIP) utiliza TCP/IP para reservas en líneas aéreas, billeteaje y tráfico de mensajes.

• Seguridad TLS

Los protocolos TLS permiten al cliente y al servidor establecer una conexión segura y cifrada a través de una red pública. Cuando se conecta mediante TLS, Host Access for the Cloud autentica el servidor antes de abrir una sesión y todos los datos transmitidos entre él y el host se cifran mediante el nivel de cifrado seleccionado.

consejo

Cuando la seguridad TLS se ha establecido en TLS 1.3 o TLS 1.2, tiene la opción de verificar el nombre de host en relación con el nombre del certificado del servidor. Le recomendamos expresamente que habilite la verificación del nombre del host para todas las sesiones.

Están disponibles las siguientes opciones:

Opciones de seguridad	Descripción
Nada	No se requiere conexión segura.
TLS 1.3	Conectar mediante TLS 1.3. Cuando Verificar identidad del servidor se ha definido en Sí , el cliente comprueba el nombre del servidor o del host en relación con el nombre del certificado del servidor. Le recomendamos expresamente que habilite la verificación del nombre del host para todas las sesiones.

Opciones de seguridad	Descripción
TLS 1.2	Conectar mediante TLS 1.2. Cuando Verificar identidad del servidor se ha definido en Sí , el cliente comprueba el nombre del servidor o del host en relación con el nombre del certificado del servidor. Le recomendamos expresamente que habilite la verificación del nombre del host para todas las sesiones.

- **Habilitar Rastreo de Emulación**

Puede seleccionar generar rastreos de host para una sesión. El valor por defecto es No.

Seleccione Sí para crear un nuevo rastreo del host de emulación cada vez que se inicie la sesión.

Uso del Administrador de ID de Terminal

MSS Para utilizar el Administrador de ID de Terminal, debe tener configurado un servidor de Administrador de ID de Terminal. Consulte [Terminal ID Manager Guide \(Guía de Administrador de ID de Terminal\)](#).

El Administrador de ID de Terminal proporciona ID a las aplicaciones cliente en tiempo de ejecución y gestiona ID agrupados para diferentes tipos de host. Un ID son datos de conexión únicos para una sesión de host individual.

Si decide utilizar el Administrador de ID de Terminal y ha configurado el servidor del mismo, puede seleccionar entre las opciones siguientes para configurar los criterios para obtener un ID. Se deben cumplir todos los criterios para obtener un ID.

nota

Recuerde que cuando especifica un criterio, indica que el ID se debe asignar solo si se encuentra un ID con ese valor específico. El conjunto de criterios seleccionados debe coincidir exactamente con el conjunto de criterios especificados en una Agrupación de IDs en el Administrador de ID de Terminal para que la solicitud de ID se pueda realizar.

Criterios del Administrador de ID de Terminal

Criterio	Descripción
Nombre de agrupación	Incluya este atributo e ingrese el nombre de la agrupación para limitar la búsqueda de ID a una agrupación específica.
Dirección IP de cliente	La dirección IP del equipo del cliente se incluirá como parte de la solicitud de un ID.
Dirección de host	La dirección del host configurado para esta sesión se incluirá como parte de la solicitud de un ID.
Puerto de host	El puerto para el host configurado para esta sesión se incluirá como parte de la solicitud de un ID.
Nombre de sesión	Cuando esta opción está seleccionada, requiere que el ID se configure para ser utilizado por esta sesión exclusivamente.
Tipo de sesión	El tipo de sesión (por ejemplo, IBM 3270, IBM 5250, UTS, ALC o T27) se incluye siempre como parte de cualquier solicitud de un ID.
Nombre de usuario	<p>Utilice este criterio para asegurarse de que sólo se asignarán IDs creados para el uso exclusivo de usuarios específicos. El nombre de usuario actual, que se debe encontrar en un ID antes de que pueda ser asignado, es el nombre del usuario al que está asignada la sesión en ejecución.</p> <p>Para configurar una sesión basada en nombres de usuario, se dispone de un nombre de usuario por defecto como marcador de posición: <code>tidm-setup</code>.</p> <p>Para que el administrador pueda configurar sesiones mediante <code>tidm-setup</code>, el Administrador de ID de Terminal debe disponer de ID para <code>tidm-setup</code>. Puede sobrescribir el nombre por defecto por uno propio. Para ello, modifique el archivo <code><install-dir>/sessionserver/conf/container.properties</code> del siguiente modo:</p> <p><code>id.manager.user.name=custom-username</code>, donde <code>custom-username</code> se sustituye por el nombre que desea utilizar.</p>
Nombre de aplicación (UTS)	El nombre de la aplicación de host se incluirá como parte de la solicitud de un ID.

Para determinar el comportamiento de intento de conexión si el Administrador de ID de Terminal no asigna con éxito un ID para esta sesión, utilice **Si ID no está asignado**:

- **Fallar intento de conexión:** si la opción está seleccionada, la sesión no intenta establecer conexión si un ID no está asignado.
- **Permitir intento de conexión:** si la opción está seleccionada, la sesión intenta establecer conexión si un ID no está asignado. El intento debe ser rechazado por el host. Hay algunos tipos de host que permiten al usuario conectarse sin ID.

Para confirmar que el Administrador de ID de Terminal puede proveer un ID utilizando las selecciones de criterios y valores que ha hecho, haga clic en [Probar los criterios del Administrador de ID de Terminal](#).

- **Enviar paquetes Keep Alive** - Seleccione esta configuración para realizar una comprobación constante entre la sesión y el host, para así detectar inmediatamente los problemas de conexión. Seleccione entre los siguientes tipos de paquetes Keep Alive:

Esta opción	Tiene esta función...
Nada	El valor por defecto. No se envían paquetes.
Sistema	La pila TCP/IP mantiene el seguimiento de la conexión host y envía paquetes Keep Alive con poca frecuencia. Esta opción utiliza menos recursos del sistema que Enviar paquetes NOP o Enviar paquetes de marca de sincronización.
Enviar paquetes NOP	Se envía periódicamente un comando No Operation (NOP) al host. No es obligatorio que el host responda a estos comandos, pero la pila TCP/IP puede detectar si hay algún tipo de problema con la entrega del paquete.
Enviar paquetes de marca de sincronización	Se envía periódicamente un comando de Marca de sincronización para determinar si la conexión continúa activa. El host debe responder a estos comandos. Si no se recibe respuesta o si se produce un error durante el envío del paquete, la conexión se cierra.

- **Tiempo de espera de Keep Alive (segundos)** - Si elige utilizar una de las opciones Enviar paquetes NOP o Enviar paquetes de marca de sincronización, seleccione el intervalo entre peticiones Keep Alive. Los valores están en el rango de 1 a 36000 segundos (1 hora); el valor predeterminado es 600 segundos.

Probar los criterios del Administrador de ID de Terminal

El Administrador de ID de Terminal provee IDs a las aplicaciones del cliente en ejecución. Para confirmar que el Administrador de ID de Terminal puede proveer un ID utilizando las selecciones de criterios y valores que ha hecho, utilice esta opción.

Los criterios para la sesión actual se especifican en el panel Conexión después de haber seleccionado **Utilizar Administrador de ID de Terminal** en el campo Nombre de Dispositivo (tipos de host 3270, 5250), en el campo ID de terminal (UTS) o en el campo ID de estación (T27). Los criterios seleccionados para la sesión actual se visualizan de forma predeterminada.

Haga clic en **Test** para confirmar que el Administrador de ID de Terminal puede proveer un ID que coincida con las selecciones de criterios y valores configuradas. La prueba devuelve el nombre de un ID disponible que satisface los valores de atributo seleccionados.

Comprobar para otros criterios y valores

También puede utilizar este panel para comprobar criterios diferentes de los asociados a la sesión actual.

1. Seleccione cualquiera de los tipos de sesión de la lista Tipo de sesión y seleccione los criterios que desea comprobar. Puede probar valores alternativos que desee utilizar en un ejemplo de solicitud al Administrador de ID de Terminal.
2. Haga clic en **Test** para confirmar que el Administrador de ID de Terminal puede proveer un ID que coincida con las selecciones de criterios y valores. La prueba devuelve el nombre de un ID disponible que satisface los valores seleccionados.

Parámetros de conexión 3270 y 5250

Además de los [parámetros de configuración comunes](#), los tipos de host 3270 y 5250 requieren estos parámetros específicos.

- **Modelo de terminal**

Especifique el modelo de terminal (conocido también como estación de visualización) que desee que emule Host Access for the Cloud. Dependiendo del tipo de host, hay distintos modelos de terminal.

Si elige **Modelo personalizado**, puede especificar el número de columnas y filas para personalizar el modelo de terminal.

- **Usar inicio de sesión automático de Kerberos (solo para 5250)** 

Si se establece en **Sí** el usuario no tiene que introducir las credenciales de entrada la sesión. El inicio de sesión automático de Kerberos se configura en Consola Administrativa de MSS > Host Access for the Cloud. Al configurar HACloud para utilizar el protocolo de autenticación Kerberos, hay términos que debe conocer y requisitos previos que debe cumplir antes de configurar esta opción. Estas opciones se explican detalladamente en la documentación del panel Consola Administrativa de MSS > Host Access for the Cloud, disponible desde el botón Ayuda. Para obtener más información, consulte "Deployment Guide" (Guía de distribución).

- **ID de terminal** (sólo 3270)

Cuando Host Access for the Cloud se conecta a un host Telnet, el protocolo Telnet y el host negocian un ID de terminal que se utilizará durante la conexión Telnet inicial. En general, de esta negociación resulta el uso del ID de terminal correcto, por lo que este cuadro se debe dejar vacío.

- **Seguridad TLS**

Los protocolos TLS permiten al cliente y al servidor establecer una conexión segura y cifrada a través de una red pública. Cuando se conecta mediante TLS, Host Access for the Cloud autentica el servidor antes de abrir una sesión y todos los datos transmitidos entre él y el host se cifran mediante el nivel de cifrado seleccionado. Consulte [Parámetros de conexión comunes](#) para obtener información sobre este parámetro común.

- **Nombre de dispositivo**

Si ha seleccionado TN3270, TN3270E o TN5250 como protocolo, especifique el nombre de dispositivo a utilizar cuando la sesión se conecte al host. El nombre de dispositivo es conocido también como host LU o pool. También puede elegir:

- **Generar nombre de dispositivo único:** genera automáticamente un nombre de dispositivo único.
- **Utilizar el Administrador de ID de Terminal:** muestra parámetros adicionales para completar. Consulte [Uso del Administrador de ID de Terminal](#).
- **Solicitar siempre el ID al usuario:** se solicita al usuario final el ID del dispositivo cada vez que se intenta establecer una conexión.
- **Solicitar al usuario si no se ha especificado el ID:** se le solicita al usuario final la primera vez que se intenta establecer una conexión; después de este intento, se guardará el valor. El valor guardado se seguirá utilizando sin que se solicite de nuevo.

Si usted no especifica un nombre de dispositivo para la sesión, el host asigna dinámicamente uno a la sesión. Un nombre de dispositivo ajustado dentro de una macro sobrescribe este parámetro.

Parámetros de conexión VT

Además de los [parámetros de conexión comunes](#), los hosts VT requieren estos parámetros adicionales. Estos parámetros varían en función del protocolo que esté utilizando: Telnet o SSH. Los parámetros son aplicables a ambos protocolos, a menos que se indique lo contrario.

Opciones de configuración de sesión VT

Parámetros de VT	Descripción
ID de terminal	Este parámetro determina la respuesta que Host Access for the Cloud envía al host tras una petición de atributos de dispositivo (DA) primaria. Esta respuesta informa al host sobre las funciones de terminal que puede llevar a cabo. La respuesta de Host Access for the Cloud para cada ID de terminal es exactamente la misma que la respuesta del terminal VT; algunas aplicaciones pueden requerir una respuesta de DA específica. Este parámetro de ID de terminal no depende del valor de Tipo de terminal. Las opciones son: VT220, VT420, VT100, DEC-VT100 y VT52.
Todos los hosts desconocidos (SSH)	Esta opción permite al administrador con capacidad de decisión determinar si el cliente Web permitirá los hosts desconocidos. Las opciones son: <p>Sí: se permiten los hosts desconocidos y todas las conexiones SSH. No se pregunta a los usuarios del cliente Web si debe confiarse en los hosts.</p> <p>Preguntar: se le pregunta al usuario del cliente web si debe confiarse en el host cuando se conecte a un host desconocido con el que no se haya encontrado antes. Si decide confiar en el host, su clave pública se almacenará en las preferencias de usuario y las conexiones posteriores no generarán un aviso a menos que la clave del host cambie.</p> <p>No: no se admiten los hosts desconocidos. Solo se permiten los hosts en los que el administrador haya decidido confiar al configurar la sesión. No se le pregunta nunca a los usuarios y la sesión se conecta o no en función de las opciones seleccionadas por el administrador.</p>
Suprimir mensajes de banner (SSH)	Cuando la opción está activada, el banner SSH no se visualiza. Esta opción es útil cuando se graban macros de inicio de sesión SSH.

Eco local (Telnet)	Automático (valor por defecto). Cómo responde Host Access for the Cloud al eco remoto enviado por un host Telnet: la opción Automático intenta negociar el eco remoto, pero realiza lo que ordena el comando. La opción Sí implica que Host Access for the Cloud negocia el eco local con el host, pero siempre establece el eco, mientras que la opción No implica que Host Access for the Cloud negocia el eco remoto con el host, pero no establece el eco.
Volver a negociar eco (Telnet)	No (predeterminado). Si el valor es Sí, las contraseñas no se visualizan en la pantalla local, pero todo el texto que se escriba está visible. Host Access for the Cloud admite la opción de Telnet Suppress Local Echo (SLE) (Suprimir eco local) cuando está conectado al host en el modo half-dúplex. Esto significa que Host Access for the Cloud suprimirá el eco de caracteres en el equipo host y, con compatibilidad con SLE, Host Access for the Cloud puede recibir instrucciones para suprimir el eco localmente.
Definir Tamaño de Ventana de Host	Sí (predeterminado). Este parámetro envía el número de filas y columnas al host Telnet siempre que cambian. Esto permite al host Telnet controlar correctamente el cursor si el tamaño de la ventana cambia.
Algoritmos de clave de host (SSH)	Define los algoritmos de clave de host que se utilizarán al conectarse a un servidor SSH. Las opciones son: EC, RSA, DSS (por defecto): esta es la opción más segura. Use esta opción durante una actualización para proporcionar más seguridad. RSA, DSS: seleccione RSA, DSS a fin de seguir utilizando una clave existente para las nuevas sesiones SSH.
Utilizar Modo Binario (Telnet)	No (predeterminado). Telnet define una ruta de datos de 7 bits entre el host y el terminal. Este tipo de ruta de datos no es compatible con algunos juegos de caracteres nacionales. Afortunadamente, muchos hosts utilizan los datos de 8 bits sin poner a cero el bit 8, lo que permite resolver este problema. Sin embargo, en algunos casos puede que sea necesario seleccionar esta casilla de verificación para forzar al host a utilizar una ruta de datos de 8 bits.
Enviar Salto de línea después de Retorno de carro (Telnet)	No (predeterminado). Un "auténtico" host Telnet espera ver una secuencia de caracteres CrNu (retorno de carro/nulo) para indicar el final de línea enviado desde el terminal. Algunos hosts en Internet no son auténticos hosts Telnet, por lo que esperan ver un carácter Lf (salto de línea) después de un carácter Cr (retorno de carro) al final de una línea. Si se está conectando a este tipo de host Telnet, seleccione Sí.

Ctrl-Interrumpir envía (Telnet)	Seleccione qué envía la secuencia Ctrl-Interrumpir al host cuando se pulsa. Las opciones son: secuencia Interrupción Telnet (predeterminada), Interrupción del proceso o Nada.
Juego de Caracteres de Host	El valor predeterminado para el Juego de caracteres de host depende del tipo de terminal que esté emulando. Este parámetro refleja el estado actual del terminal de Juego de Caracteres de Host VT, que puede ser cambiado por el host. El parámetro predeterminado asociado guardado con el modelo es DEC Suplementario.
Respuesta Automática	No (predeterminado). Este parámetro especifica si el mensaje de respuesta (configurado con la propiedad Respuesta) se envía automáticamente al host tras una conexión de línea de comunicaciones.
Cadena de Respuesta	<p>Este ajuste le permite ingresar un mensaje de respuesta si el host espera contestación como respuesta a un carácter ENQ.</p> <p>La cadena de respuesta soporta caracteres con códigos inferiores o iguales a 0xFFFF mediante secuencias de escape Unicode. La secuencia de escape empieza con <code>\u</code> seguida de exactamente cuatro dígitos hexadecimales. Usted puede integrar secuencias de escape Unicode en cualquier cadena. Por ejemplo, <code>this embedded \u0045</code> se interpretará como <code>this embedded E</code> ya que 45 es el código hexadecimal para el carácter E.</p> <p>Para enviar secuencias de escape Unicode al host, escape la secuencia anteponiendo una barra invertida. Por ejemplo, para enviar la cadena literal <code>\u001C</code> al host, asigne una tecla a <code>\u001C</code>. Host Access for the Cloud convertirá esto a la cadena <code>\u001C</code> cuando se pulse esa tecla y enviará los seis caracteres de la cadena resultante al host.</p>

Más información

[Descripciones de TLS](#)

Parámetros de conexión UTS

Además de los [parámetros de conexión comunes](#), los hosts UTS requieren estos parámetros adicionales:

Opciones de configuración de sesión UTS INT1

Opciones de UTS INT1	Descripción
Aplicación	<p>Nombre de la aplicación de host o del modo operativo del host al que se debe acceder.</p> <p>Esta es la palabra o la frase que la máquina local envía al host cuando establece comunicación con el host por primera vez. Si ha estado utilizando un terminal de host, éste debe ser el nombre \$OPEN de la aplicación. El nombre de la aplicación suele ser el mismo que el nombre del entorno, pero también pueden ser diferentes. Por ejemplo, el nombre del entorno puede ser MAPPER y el de la aplicación puede ser UDSSRC. Durante una sesión de emulación de terminal usted podría escribir \$OPEN MAPPER en el indicador e INT1 enviaría UDSSRC al host una vez que la conexión se estableciera.</p>
TSAP	<p>Transport Service Access Point (TSAP) que se desea, hasta 32 caracteres (como TIPCSU para conexiones TIP, RSDCSU para conexiones Demand). Se requiere un TSAP sólo si se está conectando a un Host LAN Controller (HLC) o a un Distributed Communications Processor (DCP) en modo router IP. Si no está seguro de qué valor utilizar, póngase en contacto con su administrador de host.</p>
Transacción inicial	<p>Carácter, palabra o frase que la máquina local enviará al host cuando se establezca por primera vez la comunicación con el host (hasta 15 caracteres). Este parámetro es opcional y se utiliza principalmente con TIP. Por ejemplo, puede escribir ^ para ejecutar MAPPER. Este parámetro se puede utilizar también para transmitir contraseñas.</p>

<p>Iniciar transacción</p>	<p>Cuando usted configura una transacción inicial, de forma predeterminada los datos se envían en cuanto se ha establecido la conexión de sesión. Usted puede decidir cuándo se envía una transacción inicial utilizando una cadena particular para activar la transacción inicial.</p> <p>Por ejemplo, para esperar un inicio de sesión correcto antes de enviar los datos de la transacción inicial, escriba una cadena que se utilice para identificar un inicio de sesión correcto.</p> <p>Puede utilizar este parámetro en combinación con Enviar transacción inicial.</p>
<p>Enviar transacción inicial</p>	<p>Puede determinar cuándo se envía la transacción inicial:</p> <p>Inmediatamente - predeterminado.</p> <p>Cuando se recibe el carácter de inicio de entrada (start of entry, SOE) - esta opción es útil cuando se deben completar transacciones multilínea antes de enviar la cadena.</p> <p>Después de los milisegundos especificados</p>
<p>ID de terminal</p>	<p>Seleccione las opciones para especificar un ID de terminal o utilizar el Administrador de ID de Terminal. Para especificar un ID de terminal, escríbalo en el campo Especificar ID de Terminal.</p> <p>Especificar el ID de Terminal El ID de Terminal, un identificador de terminal (normalmente de hasta 8 caracteres alfanuméricos) que se utilizará para la sesión de comunicación asociada a esta ruta. Conocido también como TID o PID, cada ID de terminal debe ser único para el host.</p> <p>Solicitar al usuario si no se ha especificado el ID Se le solicitará al usuario final la primera vez que se intente establecer una conexión; después de este intento, se guardará el valor. El valor guardado se seguirá utilizando sin que se solicite de nuevo.</p> <p>Solicitar siempre el ID al usuario Si se selecciona esta opción, se le solicita al usuario final el ID de terminal cada vez que se intenta establecer una conexión.</p> <p>Utilizar el Administrador de ID de Terminal Si elige Utilizar Administrador de ID de Terminal, se le pedirá que seleccione los atributos del ID de Terminal que desea utilizar para obtener un ID. Consulte Atributos del Administrador de ID de Terminal.</p> <p>Para probar los atributos, haga clic en Test.</p>

Más información

[Atributos del Administrador de ID de Terminal](#)

[Descripciones de TLS](#)

Parámetros de conexión T27

Junto a los [parámetros de conexión comunes](#), puede configurar estas opciones de conexión de T27 adicionales:

Parámetros de conexión T27

Opciones T27	Descripción
Tipo de terminal	Seleccione el tipo de terminal a emular durante la sesión. La emulación T27 soporta los tipos de terminal Unisys TD830, TD830 ASCII, TD830 INTL y TD830 NDL.
Utilizar Modo Binario	Debe habilitar la opción Utilizar Modo binario si usted requiere la impresión pass through. El valor por defecto es No. TCPA define una ruta de datos de 7 bits entre el host y el emulador de terminal. Este tipo de ruta de datos no es compatible con algunos juegos de caracteres nacionales. De todos modos, muchos hosts utilizan los datos de 8 bits sin poner a cero el bit 8, lo que permite resolver este problema. Sin embargo, puede que sea necesario seleccionar esta opción para forzar al host a utilizar una ruta de datos de 8 bits.
Ancho de línea	Seleccione el número de caracteres que el host enviará al cliente. El valor por defecto es 80 caracteres.
Seguridad TLS	Consulte Descripciones de TLS para obtener una descripción de las diversas opciones.

ID de estación	<p>Seleccione una opción para especificar un ID de estación o utilizar el Administrador de ID de Terminal. Para especificar un ID de estación, seleccione Especificar ID de estación y teclee el nombre en el campo ID de estación. Cada ID de estación debe ser único para el host y suele constar de un máximo de ocho caracteres alfanuméricos.</p> <p>Solicitar al usuario si no se ha especificado el ID Se le solicitará al usuario final la primera vez que se intente establecer una conexión; después de este intento, se guardará el valor. El valor guardado se seguirá utilizando sin que se solicite de nuevo.</p> <p>Solicitar siempre el ID al usuario Si se selecciona esta opción, se le solicita al usuario final el ID de estación cada vez que se intenta establecer una conexión.</p> <p>Utilizar el Administrador de ID de Terminal Si selecciona Utilizar el Administrador de ID de Terminal, verá un número de criterios de ID de terminal para configurar. Consulte Criterios del Administrador de ID de Terminal para obtener descripciones de las distintas opciones.</p> <p>Si no especifica un ID de estación para la sesión, el host asigna dinámicamente uno a la sesión.</p>
----------------	--

Más información

- [Descripciones de TLS](#)
- [Criterios del Administrador de ID de Terminal](#)

Parámetros de conexión ALC

Además de los [parámetros de conexión comunes](#), los hosts ALC requieren estos parámetros adicionales:

Parámetros de conexión ALC

Opciones ALC	Descripciones
Seguridad TLS	Consulte Descripciones de TLS para obtener una descripción de las diversas opciones.
Codificación de caracteres	Elija ASCII, EBCDIC o IPARS (predeterminado) como conjunto de códigos.
Archivo de configuración	Introduzca el archivo de configuración (CNF) que asocia la información de configuración apropiada para un host específico.
Dirección de terminal	<p>Seleccione si desea especificar la dirección de terminal o utilizar el Administrador de ID de Terminal.</p> <p>Dirección de terminal: especifique si desea utilizar el modo de direccionamiento de 2 o 4 bytes.</p> <p>Aunque se requiere una dirección única de 5 bytes cuando se especifica el ID del terminal en lugar de utilizar el Administrador de ID, esta opción especifica cuántos bytes de la dirección ID del terminal de 5 bytes se envían con cada mensaje con el fin de multiplexar. Si especifica el modo de direccionamiento de 2 bytes, sólo se envían los últimos 2 bytes de la dirección de grupo ASCU (Agent Set Control Unit) (A1, A2). Si especifica el modo de direccionamiento de 4 bytes, se envía la dirección de grupo ASCU completa (H1, H2, A1, A2).</p> <p>Especifique la dirección de terminal única de 5 bytes para esta sesión. La dirección del terminal se compone de cinco valores de 2 dígitos hex en este orden: H1, H2, A1, A2 y TA (dirección de terminal). Esta dirección única suele ser asignada por el administrador de la red.</p> <p>Administrador de ID de Terminal - proporciona ID a las aplicaciones del cliente en ejecución. Si elige esta opción, hay opciones de configuración adicionales a completar. Véase Criterios del Administrador de ID de Terminal para la descripción de estas opciones.</p>

Más información

- [Descripciones de TLS](#)
- [Criterios del Administrador de ID de Terminal](#)

Trabajar con sesiones

Todas las sesiones a las que usted tiene acceso están disponibles en la lista Sesiones Disponibles. El administrador del sistema crea y configura inicialmente las sesiones y se accede a ellas mediante una URL distribuida (por ejemplo, `https://<clusterdns>/webclient`).

Para abrir una sesión

1. Seleccione la sesión y haga clic para abrirla.
2. Interactúe con su aplicación de host utilizando el panel Abrir sesión.
3. Puede crear múltiples instancias de una sesión configurada.

Puede tener múltiples sesiones abiertas simultáneamente y cambiar fácilmente entre ellas con ayuda de las fichas dispuestas en la parte superior de la pantalla. La sesión actual es siempre la ficha que se encuentra más a la izquierda y se identifica por un fondo blanco y texto en negrita. Cada sesión permanece activa durante 30 minutos.

Utilice la barra de herramientas para acceder a las distintas opciones disponibles para usted cuando interactúe con la sesión. Puede desconectarse de una sesión, cerrar la sesión, activar Teclas Rápidas y acceder a otras configuraciones. Es posible que algunas opciones estén sólo disponibles cuando su administrador le haya concedido permiso.

Utilizar Teclas Rápidas

El teclado del terminal de Teclas Rápidas ofrece una representación gráfica de las teclas en un teclado del host y le da acceso rápido a las teclas del terminal.

Haga clic en una tecla del terminal en el teclado de Teclas Rápidas para enviar la tecla al host. Las sugerencias de herramientas, que se visualizan pasando el cursor por una tecla, ofrecen una descripción de la asignación.

Las teclas rápidas están disponibles para cada tipo de host y se accede a ellas haciendo clic en el icono



de la barra de herramientas.

Edición de la pantalla

💡 nota

Cada navegador gestiona las funciones para copiar, pegar y cortar de un modo distinto y, en algunos casos, no se admitirá el uso de los botones de la barra de herramientas o el menú contextual. Se recomienda el uso de comandos de teclado para esas funciones. Aunque los comandos del teclado varían según el sistema operativo, en Windows son: CTRL+C para copiar, CTRL+V para pegar y CTRL+X para cortar.

Es mucho más frecuente encontrar problemas con la función para pegar que con la función para cortar o copiar. Si no se muestra el botón Pegar de la barra de herramientas, es probable que la seguridad del navegador impida el acceso de lectura al portapapeles del sistema. Los diversos navegadores presentan un comportamiento diferente cuando se trata de proporcionar acceso al portapapeles. Sin embargo, la opción para pegar está casi siempre disponible mediante los comandos de teclado, (Control + V en Windows y Comando + V en Mac). En este caso, se presupone que no se han asignado de nuevo esas teclas. También puede utilizar el elemento de menú o el botón Pegar integrados del navegador.

Para copiar del terminal

1. Realce el área en la pantalla del terminal que desea copiar.
2. Haga clic en **Copiar** en la barra de herramientas o seleccione **Copiar** en el menú contextual disponible en la pantalla del terminal. También puede utilizar el comando de teclado, **CTRL + C**.

Para pegar en la pantalla del terminal

1. Posicione el cursor en el lugar en el que desea pegar el contenido.
2. Si el navegador admite la función para pegar, haga clic en **Pegar** en la barra de herramientas o seleccione **Pegar** en el menú contextual disponible en la pantalla del terminal. Si el navegador no admite esta función, estas opciones no estarán disponibles y deberá utilizar el comando de teclado, **CTRL + V**.

Para cortar áreas de la pantalla del terminal

💡 nota

Esta función está disponible para todos los tipos de terminales compatibles, excepto para los hosts VT.

1. Resalte el área en la pantalla del terminal que desea cortar.
2. Haga clic en **Cortar** en la barra de herramientas o seleccione **Cortar** en el menú contextual disponible en la pantalla del terminal. También puede utilizar el comando de teclado, **CTRL+X**.

Más información

[Especificar las opciones de edición](#)

Salida de la sesión

En la esquina superior derecha de la pantalla, abra la lista desplegable asociada a su nombre de usuario y seleccione **Cerrar sesión** para dejar de trabajar con la aplicación del host.

Extra!

Crear Macros

Una macro es una serie de acciones de teclado que usted graba y ejecuta después. Puede utilizar estos programas de macro JavaScript para automatizar las interacciones del usuario con el terminal. Puede acceder a macros y ejecutarlas desde todos los dispositivos compatibles.

Host Access for the Cloud graba y guarda macros avanzadas como JavaScript, lo que simplifica la edición y la mejora de las macros grabadas. Puede grabar macros para reproducirlas posteriormente, ejecutar macros al iniciar o cuando la sesión se conecta o desconecta del host. También puede escribir macros en el bloc de notas para realizar trabajos complejos que la grabadora no puede capturar.

Las macros se ponen a disposición de los usuarios de dos formas: creadas por un administrador o grabadas por los usuarios para su uso privado. Todas las macros están asociadas a una sesión y cumplen el mismo objetivo de automatizar la interacción con el host. La única diferencia entre ambas es sólo quién puede acceder a ellas y quién gestiona su creación y disponibilidad:

- **Macros creadas por administradores**

Los administradores crean macros cuando crean la sesión. Son específicas de una sesión y están disponibles para todos los usuarios que tienen acceso a la sesión desde el icono Macro en la barra de herramientas. Los administradores pueden designar macros para ejecutarlas al iniciar o cuando la sesión se conecta o desconecta del host.

- **Macros creadas por usuarios**

Los usuarios crean macros de usuarios finales para las sesiones para las que tienen autorización de acceso. El administrador concede permiso para crear macros configurando una Regla de Preferencias del Usuario. Los usuarios pueden acceder a la sesión utilizando sus propias credenciales o con función de Invitado. Las macros creadas por usuarios Invitados están disponibles para todos los usuarios Invitados. Los usuarios que han iniciado sesión utilizando sus propias credenciales pueden ver sólo las macros que han creado ellos.

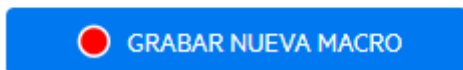
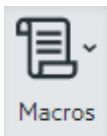
Las macros avanzadas se listan en orden alfabético en la lista desplegable de la barra de herramientas. Las macros creadas por el usuario final se listan primero y van seguidas de un indicador de tres puntos grises en vertical que, cuando se ha seleccionado, muestra las opciones de Editar y Eliminar. Las macros creadas por el administrador se listan sin el indicador ya que esas macros no pueden ser modificadas por el usuario final.


Trabajo con macros

Siga estos pasos para grabar, editar y ejecutar macros.

Grabar

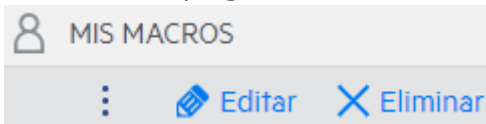
1. Haga clic en el icono Macro de la barra de herramientas y haga clic en Grabar Nueva Macro.




2. Navegue por la aplicación del host para grabar las series de pasos que desea incluir en la macro.
3. Haga clic en  en la barra de herramientas para detener la grabación. El punto rojo parpadea para indicar que la grabación está en curso.
4. Cuando se le pida, escriba un nombre para la macro.


Editar

1. En la lista desplegable Macro, seleccione la macro que desea editar.




2. Haga clic en los tres puntos verticales para expandir el campo.
3. Haga clic en  **Editar** para abrir el Editor de macros (en el panel izquierdo).
4. Utilice JavaScript para realizar los cambios que sean necesarios. Puede ejecutar y guardar la macro modificada utilizando los iconos de la barra de herramientas en el panel superior del editor.

Ejecute

Para ejecutar una macro, elija la macro en la lista desplegable y haga clic en .

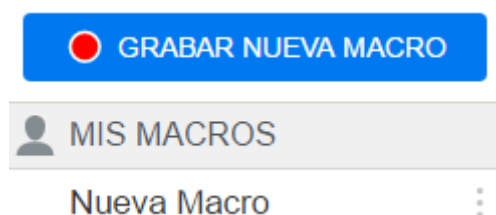
También puede asignar teclas que activarán automáticamente una macro ya grabada. En el cuadro de diálogo de configuración Asignar Tecla, seleccione **Ejecutar Macro** de la lista desplegable **Acción**. Seleccione una macro a asociar con la asignación de tecla de la lista **Valor**.

Detener

Puede detener una macro antes de que se complete desde el Editor de Macro o la barra de herramientas. Haga clic en  para detener la macro. Para volver a ejecutar la macro, navegue de vuelta a la pantalla de inicio de macro.

Suprimir

1. Seleccione en la lista desplegable Macro la macro que desea eliminar.
2. Expanda el campo haciendo clic en el icono de los tres puntos en vertical.



3. Haga clic en **Eliminar**.

Ver

La lista desplegable Macro está disponible desde la barra de herramientas para todos los usuarios que tienen permiso para grabar macros o acceden a una sesión en la que las macros han sido grabadas previamente por el administrador para que se utilicen en esa sesión.

Las macros se enumeran en **MIS MACROS** o **MACROS** en función de cómo se hayan grabado.

Todos los usuarios, tanto si han iniciado sesión con sus credenciales o como Invitado, pueden ver las macros asociadas a la sesión. Las macros enumeradas bajo el encabezado **MIS MACROS** aparecen en orden alfabético por su nombre y están visibles para los usuarios que las hayan grabado. Las macros grabadas por el administrador y asociadas a una sesión se muestran en orden alfabético en **MACROS**.

Depuración de macros

Como las macros están escritas en JavaScript y se ejecutan en el navegador, la mejor forma de depurarlas y de solucionar los problemas con ellas es utilizar las herramientas integradas en su navegador web. Los navegadores modernos vienen con un set de herramientas muy completo para depurar el código de JavaScript. Puede colocar puntos de interrupción, comprobar el código y obtener información de depuración.

sugerencia

JavaScript distingue entre mayúsculas y minúsculas. Recuérdelo a la hora de editar el código de JavaScript.

Para depurar una macro:

1. Abra la macro que se va a editar. Consulte [Trabajar con macros](#) para obtener instrucciones.
2. Abra las herramientas de desarrollo del navegador.

Navegador	Abrir depurador
Mozilla Firefox	Desde la barra de herramientas, abra el Menú y seleccione Desarrollador. Desde el Menú Desarrollador Web, seleccione Depurador. El depurador se abre en un panel inferior.
Google Chrome	Desde la barra de herramientas, abra el Menú y seleccione Más herramientas. Seleccione Herramientas de Desarrollador para abrir el Depurador.

3. Utilice una de las herramientas en el código de macro y ejecute el código.

- **debugger**

El enfoque más minucioso para depurar es utilizar la instrucción `"debugger"`. Cuando usted inserta estas instrucciones en su código de macro y lo ejecuta con las herramientas de desarrollo del navegador abiertas, la ejecución se detiene en esas líneas. Puede comprobar su macro, ver el valor de las variables locales y cualquier cosa que necesite comprobar.

Le animamos a colocar múltiples instrucciones depurador; en su código para ayudarle a obtener la línea correcta. La naturaleza asíncrona de JavaScript puede hacer difícil la comprobación del código. Esto se puede compensar utilizando múltiples instrucciones depurador; colocadas cuidadosamente.

Ejemplo 1: `debugger`

```
var hostCommand = menuSelection + '[enter]';
debugger; // ← El depurador se detendrá aquí.
ps.sendKeys(hostCommand);
```

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

Estas dos opciones se suelen utilizar para depurar JavaScript. Aunque no son tan flexibles como la instrucción Depurador, ofrecen una vía rápida para obtener información de depuración. Estas

funciones transmiten la información a la ficha “Consola” de JavaScript en las herramientas de desarrollador del navegador.

Ejemplo 2: `console.log()`, `alert()`

```
var hostCommand = menuSelection + '[enter]';
console.log('Command: ' +
hostCommand); // ← Proporcionará la cadena a la pestaña "Console".
alert('Command:' + hostCommand); // Aparecerá una pequeña ventana que
contiene los datos.
ps.sendKeys(hostCommand);
```

- `ui.message()`

La API de macros de Host Access for the Cloud proporciona una función de `ui.message()` que es muy similar a la función `alert()` de JavaScript. También puede utilizar la `ui.message()` para obtener información de depuración.

Ejemplo 3: `ui.message()`

```
var hostCommand = menuSelection + '[enter]';
ui.message('Command:' + hostCommand); // ← Aparecerá una ventana de
mensaje.
ps.sendKeys(hostCommand);
```

Tenga en cuenta lo siguiente:

- Comprobar y “yields”

Mientras que las instrucciones `yield` en las macros las hacen más fáciles de entender, pueden hacer la comprobación del código con el depurador más difícil. Considere o bien utilizar múltiples instrucciones de depurador o instrucciones de depurador cuidadosamente colocadas de llamadas `console.log()` para obtener la información de depuración correcta.

Utilizar la API de macros

En Host Access for the Cloud, las macros se graban y escriben mediante JavaScript.

La API de macros consiste en un conjunto de objetos que puede utilizar para interactuar con el host, esperar estados de pantallas e interactuar con el usuario.

Acerca de promises y yields

Debido a que JavaScript se controla mediante un solo subproceso y utiliza "callback functions" y "promises" para ayudar a gestionar el flujo de ejecución del código, a veces puede ser difícil seguir el código. Host Access for the Cloud combina el concepto de "promises" con la clave "yield" para que el código de la macro se pueda organizar de forma más lineal.

- **Promises**

Promises son patrones que ayudan a simplificar funciones que devuelven sus resultados de forma asíncrona en algún momento en el futuro. Todas las funciones "wait" y "ui" de la API de macros devuelven objetos "promise".

- **Yield**

Las macros utilizan la palabra clave "yield" para bloquear la ejecución de la macro hasta que se resuelva o realice una "promise". Así, si se coloca yield enfrente de cualquier función 'wait' o 'ui', se detiene la ejecución hasta que esa función ha terminado de ejecutarse. Puede colocar la clave yield enfrente de cualquier función que devuelva una promise, también de sus propias funciones personalizadas.

nota

La capacidad de bloquear la ejecución de la macro mediante la combinación de "yield" con "promises" está habilitada por la función `createMacro()`.

Errores

Los errores se pueden tratar en las macros utilizando una instrucción try / catch. Algunas funciones API pueden arrojar errores si, por ejemplo, no se cumplen las condiciones o si se sobrepasa el tiempo de espera. El error arrojado es 'atrapado' en la instrucción catch. Puede ajustar bloques de código más pequeños en una instrucción try / catch para tratar los errores a un nivel más granular.

Los desarrolladores de macro pueden generar también errores con `throw new Error('Mensaje de error útil');`

Más información

[Objetos de Macro API](#)

Ejemplos de Macros

Objetos de Macro API

Puede crear macros utilizando la Macro API. De forma predeterminada para el uso en macros, se dispone de cuatro objetos primarios:

- **Session:** el punto de entrada principal para acceder al host. El objeto Session se utiliza para conectar, desconectar y proveer acceso al objeto PresentationSpace.
- **PresentationSpace:** representa la pantalla y proporciona muchas funciones comunes, como obtener y ajustar la ubicación del cursor, enviar datos al host y leer de la pantalla. Se obtiene llamando `session.getPresentationSpace()`.
- **Wait:** facilita una forma sencilla de esperar a que se produzcan varios estados del host antes de seguir enviando más datos o leer de la pantalla. Por ejemplo, puede esperar a que el cursor esté situado en una posición determinada, a que haya texto presente en una posición de la pantalla o simplemente esperar una cantidad de tiempo fija. Todas las llamadas de la función 'Wait' requieren la palabra clave `yield`, que se explica más adelante.
- **Interfaz de usuario:** disponible automáticamente en la macro como la variable "ui". Provee capacidades básicas de interfaz de usuario. Puede utilizar este objeto para mostrar datos al usuario o para indicarlos a modo de información. Todas las llamadas de la función "UI" requieren la palabra clave "yield".

Todos los objetos disponibles

Consulte la lista de objetos disponibles en el panel de navegación derecho, "En esta página". (Es posible que deba expandir el navegador).

Attribute

Utilice el objeto Attribute, junto con el objeto AttributeSet, para descifrar la información de formato presente en la celda de datos.

Attribute	Indica...
PROTECTED	Una celda de datos protegida.
MODIFIED	Una celda de datos modificada.
NUMERIC_ONLY	El inicio de una celda de datos solo de caracteres numéricos.

Attribute	Indica...
ALPHA_NUMERIC	Una celda de datos alfanuméricos.
HIGH_INTENSITY	Si la celda de datos contiene texto de alta intensidad.
HIDDEN	Si la celda de datos contiene texto oculto.
PEN_DETECTABLE	Si la celda de datos es detectable por lápiz.
ALPHA_ONLY	Una celda solo de datos alfabéticos.
NUMERIC_SHIFT	El inicio de un campo numérico mayúscula.
NUMERIC_SPECIAL	Que la celda de datos marca el inicio de un campo numérico especial.
KATAKANA_SHIFT	Una sección de texto Katakana.
MAGNETIC_STRIPE	Que la celda de datos marca el inicio de un campo de banda magnética.
SIGNED_NUMERIC_ONLY	Que la celda de datos es un campo numérico firmado.
TRANSMIT_ONLY	Que la celda de datos es un campo solo de transmisión.
FIELD_END_MARKER	Que la celda de datos marca el final de un campo modificado.
FIELD_START_MARKER	Que la celda de datos marca el inicio de un campo modificado.
SPECIAL_EMPHASIS_PROTECTED	Un campo protegido de énfasis especial.
TAB_STOP	Que la celda de datos contiene una posición de tabulación.
REVERSE	Que la celda de datos se visualiza en modo de vídeo inverso.
BLINKING	Que la celda de datos contiene texto intermitente.
RIGHT_JUSTIFIED	Que la celda de datos marca el inicio de un campo justificado a la derecha.
LEFT_JUSTIFIED	Que la celda de datos marca el inicio de un campo justificado a la izquierda.
LOW_INTENSITY	Que la celda de datos contiene texto de baja intensidad.

Attribute	Indica...
UNDERLINE	Que la celda de datos contiene texto subrayado.
DOUBLE_BYTE	Que la celda de datos contiene texto de doble byte.
COLUMN_SEPARATOR	Que la celda de datos contiene un separador de columnas.
BOLD	Que la celda de datos contiene texto en negrita.
DOUBLE_WIDTH	Que la celda de datos marca un campo de anchura doble.
DOUBLE_HEIGHT_TOP	Una celda de datos superior de altura doble.
DOUBLE_HEIGHT_BOTTOM	Una celda de datos inferior de altura doble.
CONTROL_PAGE_DATA	Que la celda de datos contiene datos de página de control.
RIGHT_COLUMN_SEPARATOR	Que la celda de datos contiene un separador de columnas derecho.
LEFT_COLUMN_SEPARATOR	Que la celda de datos contiene un separador de columnas izquierdo.
UPPERSCORE	Que la celda de datos contiene sobrrayado.
STRIKE_THROUGH	Que la celda de datos contiene texto tachado.

AttributeSet

El objeto AttributeSet permite al usuario descifrar los atributos presentes en la celda de datos. El objeto AttributeSet devuelve valores definidos en el objeto [Attribute](#) y cuando se utilizan juntos se puede obtener información de formato de la celda de datos.

Método	Descripción
<code>contains(attribute)</code>	<p>Determina si el conjunto contiene el atributo especificado.</p> <p>Parámetros</p> <p><code>{Number}</code> Atributo que se comprobará.</p> <p>Devuelve</p> <p><code>{Boolean}</code> "True" (verdadero) si el atributo se encuentra en el conjunto.</p>

<code>isEmpty()</code>	<p>Determina si el conjunto de atributos está vacío.</p> <p>Devuelve</p> <p><code>{Boolean}</code> "True" si el conjunto está vacío.</p>
<code>size()</code>	<p>Indica el número de atributos en un conjunto.</p> <p>Devuelve</p> <p><code>{Number}</code> El número de atributos.</p>
<code>toArray()</code>	<p>Convierte el conjunto de atributos interno en una matriz.</p> <p>Devuelve</p> <p><code>{Number[]}</code> Matriz de valores de atributos en el conjunto.</p>
<code>toString()</code>	<p>Convierte el conjunto de atributos interno en una cadena.</p> <p>Devuelve</p> <p><code>{String}</code> Nombres de atributos con espacio delimitado en el conjunto.</p>
<code>forEach(callback, thisArg)</code>	<p>Función para iterar sobre cada elemento en el conjunto de atributos.</p> <p>Parámetros</p> <p><code>{forEachCallback}</code> Devolución de llamada para realizar la operación específica. Se llama con el nombre de cada atributo en el conjunto.</p> <p><code>{Object} thisArg</code> Puntero opcional a un objeto de contexto.</p>

<code>forEachCallback(string, object)</code>	<p>Un usuario ha provisto la función callback donde usted provee el comportamiento de ser utilizado como el parámetro callback para <code>forEach</code>.</p> <p>Parámetros</p> <p><code>{String}</code> Nombre de cadena de un atributo en el conjunto de atributos.</p> <p><code>{Object}</code> <code>thisArg</code> Puntero opcional a un objeto de contexto.</p>
--	--

AutoSignOn

Método	Descripción
<code>getPassTicket()</code>	<p>Obtiene un ticket de paso que se utiliza para iniciar sesión en una aplicación de mainframe. Se pueden solicitar múltiples tickets de paso utilizando distintos IDs de aplicación.</p> <p>Parámetros</p> <p><code>{String}</code> El ID de aplicación indica al host la aplicación para la que es la entrada.</p> <p>Devuelve</p> <p><code>{Promise}</code> Se cumple con la clave de ticket de paso o se rechaza si la operación falla. El ticket de paso obtenido del DCAS funciona sólo con la sesión de host actual y es válido durante diez minutos.</p>
<code>sendUserName()</code>	<p>Aplica el nombre de usuario incluido en el ticket de paso al campo en la posición actual del cursor en la pantalla de host actual. El nombre de usuario se debe enviar antes que la contraseña. Si se envía la contraseña primero, el ticket de paso quedará invalidado y usted tendrá que obtener otro.</p> <p>Parámetros</p> <p><code>{String}</code> <code>passTicketKey</code> obtenida del <code>getPassTicket</code>.</p> <p>Devuelve</p> <p><code>{Promise}</code> Se cumple si se envía correctamente el nombre de usuario. Rechazado si la operación falla.</p>

```
sendPassw  
ord()
```

Aplica la contraseña incluida en el ticket de paso al campo en la posición actual del cursor en la pantalla de host actual. El nombre de usuario se debe enviar antes que la contraseña. Si se envía la contraseña primero, el ticket de paso quedará invalidado y usted tendrá que obtener otro.

Parámetros

{String} passTicketKey obtenida de getPassTicket.

Devuelve

{Promise} Se cumple si se envía correctamente la contraseña. Rechazado si la operación falla.

Color

Constantes de color a utilizar para los colores de primer plano y de fondo del objeto DataCell.

Color	Descripción	Valor Numérico
BLANK_UNSPECIFIED	Ningún color especificado	0
BLUE	Azul	1
GREEN	Verde	2
CYAN	Cian	3
RED	Rojo	4
MAGENTA	Magenta	5
YELLOW	Amarillo	6
WHITE_NORMAL_INTENSITY	Blanco de intensidad normal	7
GRAY	Gris	8
LIGHT_BLUE	Azul claro	9
LIGHT_GREEN	Verde claro	10
LIGHT_CYAN	Cian claro	11
LIGHT_RED	Rojo claro	12
LIGHT_MAGENTA	Magenta claro	13
BLACK	Negro	14

Color	Descripción	Valor Numérico
WHITE_HIGH_INTENSITY	Blanco de alta intensidad	15
BROWN	Marrón	16
PINK	Rosa	17
TURQUOISE	Turquesa	18

ControlKey

El objeto ControlKey define constantes para enviar teclas de control de cursor y comandos de host utilizando el método sendKeys. Las constantes están disponibles para estos tipos de host:

[IBM 3270](#)

[IBM 5250](#)

[VT](#)

[UTS](#)

IBM 3270

Palabra clave	Descripción
ALTVIEW	Alternar vista
ATTN	Atención
BACKSPACE	Retroceso
BACKTAB	TabAtrás
CLEAR	Borrar o Borrar pantalla
CURSOR_SELECT	Selección de cursor
DELETE_CHAR	Eliminar o Eliminar carácter
DELETE_WORD	Borrar palabra
DEST_BACK	Borrar al utilizar Retroceso
DEV_CANCEL	Cancelar dispositivo

Palabra clave	Descripción
DOWN	Cursor abajo
DSPSOSI	Mostrar SO/SI
DUP	Campo duplicado
END_FILE	Final de campo
INTRO	Intro
ERASE_EOF	Eliminar final de campo
ERASE_FIELD	Eliminar campo
ERASE_INPUT	Eliminar entrada
FIELD_MARK	Marca de campo
HOME	Inicio de cursor
IDENT	Ident
INSERT	Insertar
LEFT_ARROW	Cursor izquierda
LEFT2	Dos posiciones a la izquierda
NEW_LINE	Nueva línea
PA1 - PA3	PA1 - PA3
PF1 - PF24	PF1 - PF24
PAGE_DOWN	Avance página
PAGE_UP	Retroceso página
RESET	Reset, reset terminal
RIGHT2	2 posiciones a la derecha
RIGHT_ARROW	Cursor derecha, derecha
SYSTEM_REQUEST	Solicitud de sistema
TAB	Tecla tabulación

Palabra clave	Descripción
UP	Cursor arriba

IBM 5250

Palabra clave	Descripción
ALTVIEW	Alternar vista
ATTN	Atención
AU1 - AU16	AU1 - AU16
BACKSPACE	Retroceso
BACKTAB	TabAtrás
BEGIN_FIELD	Principio de campo
CLEAR	Borrar o Borrar pantalla
DELETE_CHAR	Eliminar o Eliminar carácter
DEST_BACK	Borrar al utilizar Retroceso
DOWN	Cursor abajo
DSPSOSI	Mostrar SO/SI
DUP	Campo duplicado
END_FILE	Final de campo
INTRO	Intro
ERASE_EOF	Eliminar final de campo
ERASE_FIELD	Eliminar campo
ERASE_INPUT	Eliminar entrada
FIELD_EXT	Salir del campo
FIELD_MINUS	Campo resta
FIELD_PLUS	Campo suma

Palabra clave	Descripción
FIELD_MARK	Marca de campo
HELP	Solicitud de ayuda
HEXMODE	Modo Hex
HOME	Inicio de cursor
INSERT	Insertar
LEFT_ARROW	Cursor izquierda
NEW_LINE	Nueva línea
PA1 - PA3	PA1 - PA3
PF1 - PF24	PF1 - PF24
PAGE_DOWN	Avance página
PAGE_UP	Retroceso página
[imprimir]	Imprimir
RESET	Reset, reset terminal
RIGHT_ARROW	Cursor derecha, derecha
SYSTEM_REQUEST	Solicitud de sistema
TAB	Tecla tabulación
UP	Cursor arriba

VT

Palabra clave	Descripción
BACKSPACE	Retroceso
BREAK	Interrumpir
CLEAR	Borrar o Borrar pantalla
CURSOR_SELECT	Selección de cursor

Palabra clave	Descripción
DELETE_CHAR	Eliminar o Eliminar carácter
DOWN	Cursor abajo
EK_FIND	Editar buscar de teclado numérico
EK_INSERT	Editar insertar de teclado numérico
EK_NEXT	Editar siguiente de teclado numérico
EK_PREV	Editar anterior de teclado numérico
EK_REMOVE	Editar quitar de teclado numérico
EK_SELECT	Editar seleccionar de teclado numérico
END_FILE	Final de campo
INTRO	Intro
F1 - F24	F1 - F24
HOLD	Retención
HOME	Inicio
INSERT	Insertar
KEYPAD_COMMA	Teclado numérico Coma
KEYPAD_DOT	Teclado numérico decimal
KEYPAD_ENTER	Teclado numérico Intro
KEYPAD_MINUS	Teclado numérico -
KEYPAD0 - KEYPAD9	Teclado numérico 0 - Teclado numérico 9
LEFT_ARROW	Cursor izquierda
PF1 - PF20	PF1 - PF20
PAGE_DOWN	Avance página
PAGE_UP	Retroceso página
RESET	Reset, reset terminal

Palabra clave	Descripción
RETURN	Retorno, retorno de carro
RIGHT_ARROW	Cursor derecha, derecha
TAB	Tecla tabulación
UDK16 - UDK20	Tecla definida por el usuario 6 - Tecla definida por el usuario 20
UP	Cursor arriba

UTS

Palabra clave	Descripción
BACKSPACE	Retroceso
BACKTAB	TabAtrás
CHAR_ERASE	Borra el carácter en la posición del cursor y avanza el cursor.
CLEAR_DISPLAY	Borrar pantalla
CLEAR_EOD	Eliminar hasta final de pantalla
CLEAR_EOF	Eliminar hasta final de campo
CLEAR_EOL	Eliminar hasta final de línea
CLEAR_FCC	Borrar carácter de control de campo
CLEAR_HOME	Borrar pantalla y mover el cursor a la posición inicial
CONTROL_PAGE	Alternar la página de control
DELETE_LINE	Elimina la línea en la que se encuentra el cursor y sube las líneas restantes una fila arriba
DELIN_LINE	Borra el carácter que se encuentra debajo del cursor y mueve los caracteres restantes una línea hacia la izquierda.
DELIN_PAGE	Borra el carácter que se encuentra debajo del cursor y mueve los caracteres restantes una página hacia la izquierda.
DOWN	Mueve el cursor una línea hacia abajo. Se ajusta en la parte inferior.

Palabra clave	Descripción
DUP_LINE	Crea una copia de la línea actual y sobrescribe la línea siguiente con la duplicada.
END_FIELD	Mueve el cursor al final del campo actual.
END_PAGE	Mueve el cursor al final de la página actual.
EURO	Inserta el carácter del Euro
F1 - F22	Teclas de función F1-F22
HOME	Mueve el cursor al principio de la página actual (fila 1, col 1)
INSERT	Alterna el modo insertar/sobrescribir.
INSERT_IN_LINE	Inserta un espacio en la posición del cursor y mueve los caracteres restantes de la línea a la derecha. El carácter en la columna más a la derecha de la línea se descarta.
INSERT_IN_PAGE	Inserta un espacio en la posición del cursor y mueve los caracteres restantes de la página a la derecha. El carácter en la columna más a la derecha de cada línea se descarta.
INSERT_LINE	Inserta una nueva línea en la flecha del cursor y mueve las líneas restantes hacia abajo. La última fila de la página se descarta.
LEFT_ARROW	Mueve el cursor una posición hacia la izquierda ajustándolo de ser necesario.
LOCATE_FCC	Encuentra el carácter de control de campo siguiente en la pantalla.
MSG_WAIT	Recupera la cola de mensajes al terminal.
RETURN	Retorno de carro
RIGHT_ARROW	Mueve el cursor una posición hacia la derecha ajustándolo de ser necesario.
SOE	Inserta el carácter de Inicio de Entrada
START_OF_FIELD	Mueve el cursor al principio del campo.
START_OF_LINE	Mueve el cursor a la columna 1 de la línea actual.
TAB	Mueve el cursor a la posición de tabulación siguiente de la pantalla.

Palabra clave	Descripción
TOGGLE_COL_SEP	Alterna el atributo de separador de columna.
TOGGLE_STRIKE_THRU	Alterna el atributo de tachado en la celda de datos actual.
TOGGLE_UNDERLINE	Alterna el atributo de subrayado en la celda de datos actual.
TRANSMIT	Transmite datos de campo modificados al host.
UNLOCK	Envía la tecla UNLOCK al host.
UP	Mueve el cursor una fila hacia arriba, ajustándolo si es necesario.

DataCell

El objeto DataCell ofrece información sobre una posición particular en una pantalla del terminal.

Método	Descripción
<code>getPosition()</code>	Devuelve la posición de esta celda de datos en la pantalla. Devuelve {Position} La posición de la celda de datos en la pantalla.
<code>getChar()</code>	Obtiene el carácter asociado a la celda. Devuelve {String} El carácter asociado a la celda.
<code>getAttributes()</code>	Devuelve el conjunto de atributos especificado para esta instancia de celda de datos. Consulte AttributeSet . Devuelve {AttributeSet} El conjunto de atributos de esta instancia de celda de datos.
<code>getForegroundColor()</code>	Devuelve el color de primer plano, como está definido en el objeto Color, para esta celda de datos. Devuelve {Number} Color en primer plano de esta celda de datos. El color se define en el objeto Color .

<code>getBackgroundColor()</code>	Devuelve el color de fondo, como está definido en el objeto <code>Color</code> , para esta celda de datos. Devuelve {Number} Color de fondo de esta celda de datos. El color se define en el objeto <code>Color</code> .
<code>toString</code>	Convierte la celda de datos interna en una cadena. Devuelve {String} La representación en forma de cadena de una celda de datos.
<code>isFieldDelimiter()</code>	Comprueba si esta celda representa un delimitador de campo. Devuelve {Boolean} "True" (verdadero) si esta celda es un delimitador de campo; de lo contrario, "False" (falso).

Dimensión

Representa el tamaño de la pantalla o de la región de la pantalla.

Método	Descripción
<code>Dimension(rows, cols)</code>	Crea una nueva instancia <code>Dimension</code> . Parámetros {Number} <code>filas</code> Dimensión de las filas de la pantalla. {Number} <code>columnas</code> Dimensión de las columnas de la pantalla.

Field

Utilice el objeto `Field` con `FieldList` para obtener la información presente en un campo de la pantalla.

Método	Descripción
<code>getAttributes()</code>	Devuelve el conjunto de atributos especificado para esta instancia de campo. Consulte <code>AttributeSet</code> . Devuelve {AttributeSet} El conjunto de atributos de este campo.

<code>getForegroundColor()</code>	<p>Devuelve el color en primer plano para el campo.</p> <p>Devuelve <code>{Number}</code> Color en primer plano de este campo. Estos valores se definen en el objeto Color.</p>
<code>getBackgroundColor()</code>	<p>Devuelve el color de fondo para el campo.</p> <p>Devuelve <code>{Number}</code> Color de fondo de este campo. Estos valores se definen en el objeto Color.</p>
<code>getStart()</code>	<p>Devuelve la posición inicial del campo. La posición inicial es la posición del primer carácter del campo. Algunos tipos de host utilizan una posición de carácter para guardar los atributos de nivel de campo. En este caso, la posición del atributo no está considerada la posición inicial.</p> <p>Devuelve <code>{Position}</code> Posición inicial del campo.</p> <p>Genera <code>{RangeError}</code> Para campos de longitud cero.</p>
<code>getEnd()</code>	<p>Devuelve la posición final del campo. La posición final es la posición en el espacio de representación que contiene el último carácter del campo.</p> <p>Devuelve <code>{Position}</code> Posición final del campo.</p> <p>Genera <code>{RangeError}</code> Para campos de longitud cero.</p>
<code>getLength()</code>	<p>Devuelve la longitud del campo. Para los tipos de host que utilizan una posición de carácter para guardar los atributos de campo, la longitud del campo no incluye la posición del atributo de campo.</p> <p>Devuelve <code>{Number}</code> Longitud del campo.</p>
<code>getDataCells()</code>	<p>Obtiene las celdas de datos que comprende este campo. Consulte DataCell.</p> <p>Devuelve <code>{DataCell[]}</code> Las celdas de datos que comprenden este campo.</p>
<code>getText()</code>	<p>Obtiene el texto del campo.</p> <p>Devuelve <code>{String}</code> Campo de texto.</p>

<code>setText()</code>	<p>Establece el texto del campo. Para algunos tipos de host, como los VT, el texto se transmite al host de inmediato, pero en otros tipos de host el texto no se transmite al host hasta que se invoca una tecla de ayuda. Si el texto es más corto que el campo, el texto se coloca en el campo del host y el resto del campo se borra. Si el texto es más largo que el campo del host, se coloca tanto texto como quepa en el campo.</p> <p>Parámetro <code>{String}</code> Texto que se colocará en el campo.</p> <p>Genera <code>{Error}</code> Si el campo está protegido.</p>
<code>clearField()</code>	<p>Borra el campo actual de forma específica de la emulación.</p> <p>Genera <code>{Error}</code> Si el campo está protegido o no se admite el borrado.</p>
<code>getPresentationSpace()</code>	<p>Obtiene el objeto <code>PresentationSpace</code> que ha creado este campo.</p> <p>Devuelve <code>{PresentationSpace}</code> Elemento principal de esta instancia de campo.</p>
<code>toString()</code>	<p>Crea una descripción sencilla del campo.</p> <p>Devuelve <code>{String}</code> Una interpretación del campo legible por el usuario.</p>

FieldList

Utilice el objeto `FieldList` junto con el objeto `Field` para obtener información de la lista de campos.

Método	Descripción
<code>getPresentationSpace()</code>	<p>Obtiene el objeto <code>PresentationSpace</code> que ha creado este campo.</p> <p>Devuelve <code>{PresentationSpace}</code> Elemento principal de esta instancia de campo.</p>

<code>findField(position, text, direction)</code>	<p>Devuelve el campo que contiene el texto especificado. La búsqueda empieza desde la posición especificada y se realiza hacia delante o hacia detrás. Si la cadena contiene múltiples campos, se devuelve el campo que contiene la posición inicial. Cuando se busca hacia delante, la búsqueda no se ajusta a la parte superior de la pantalla. Cuando se busca hacia atrás, la búsqueda no se ajusta a la parte inferior de la pantalla.</p> <p>Parámetros</p> <p><code>{Position}</code> Posición desde la que se debe iniciar la búsqueda. Consulte el objeto <code>Position</code>.</p> <p><code>{String}</code> El texto a buscar (opcional). Si no se indica, devuelve el campo siguiente a la derecha o debajo de la posición especificada.</p> <p><code>{Number}</code> Dirección de la búsqueda (opcional). Utilice PresentationSpace. Constantes <code>SearchDirection</code> para estos parámetros. Por ejemplo, <code>PresentationSpace.SearchDirection.FORWARD</code> o <code>PresentationSpace.SearchDirection.BACKWARD</code>. Si no se indica, se busca hacia delante.</p> <p>Devuelve</p> <p><code>{Field}</code> Contiene la cadena o cero si no se encuentra un campo que cumpla los criterios especificados.</p> <p>Genera</p> <p><code>{RangeError}</code> Si la posición está fuera del rango.</p>
<code>get(index)</code>	<p>Obtiene el campo en el índice dado.</p> <p>Parámetros</p> <p><code>{Number}</code> Índice en la lista de campos.</p> <p>Devuelve</p> <p><code>{Field}</code> Ubicado en el índice especificado.</p> <p>Genera</p> <p><code>{RangeError}</code> Si el índice está fuera del rango.</p>
<code>isEmpty()</code>	<p>Determina si la lista de campos está vacía.</p> <p>Devuelve</p> <p><code>{Boolean}</code> "True" (verdadero) si la lista está vacía.</p>
<code>size()</code>	<p>Indica el número de campos en la lista.</p> <p>Devuelve</p> <p><code>{Number}</code> El número de campos.</p>

`toString()` Crea una descripción sencilla la lista de campos.
Devuelve
`{String}` Una interpretación de la lista de campos legible por el usuario.

FileTransfer

Utilice el objeto FileTransfer para listar y transferir archivos entre el sistema del host y el cliente.

La API de transferencia de archivos de Host Access for the Cloud abstrae las convenciones de ruta de archivos utilizadas por diferentes implementaciones de archivos del host. Siga los formatos de ruta de sistema de archivos URL o Linux a la hora de formatear las rutas de archivo utilizadas por la API. Por ejemplo, `/root/directory/file`.

Es importante observar todas las reglas específicas de los sistemas de host, como los caracteres permitidos o las longitudes de los nombres.

💡 nota

Los navegadores imponen importantes restricciones de seguridad sobre la capacidad de Javascript para interactuar con los sistemas de archivos de los clientes.

Método	Descripción
<code>getHostFileListing(remotePath())</code>	<p>Solicitar un listado de archivos de host. Si se omite <code>RemotePath</code>, se muestra un listado de archivos para el directorio de trabajo remoto actual.</p> <p>Parámetros <code>{String}</code> (opcional) Si se especifica, se obtiene un listado de archivos para la ruta remota especificada. Si no se especifica, se obtiene un listado de archivos para el directorio de trabajo remoto actual.</p> <p>Devuelve <code>{Promise}</code> Se resuelve en una matriz de objetos HostFile incluidos en <code>remoteName</code>. Se rechaza si la ruta remota no se puede leer.</p>

<code>sendFile(localFile, remoteName)</code>	<p>Envía el archivo especificado al host.</p> <p>Parámetros</p> <ul style="list-style-type: none"><code>{File}</code> Objeto de archivo de Javascript que señala al archivo local que se va a enviar.<code>{String}</code> (opcional) Nombre completo de archivo remoto tal y como lo permite el sistema remoto (Unix, Windows, MVS o VAX). <p>Devuelve</p> <ul style="list-style-type: none"><code>{Promise}</code> Se cumple con un objeto <code>HostFile</code> que indica que el archivo se ha enviado correctamente. Se rechaza si se ha producido un error al enviar el archivo.
<code>getDownloadURL(remoteName)</code>	<p>Construye un vínculo para descargar un archivo desde un sistema de host.</p> <p>Parámetros</p> <ul style="list-style-type: none"><code>{String}</code> Nombre completo de archivo remoto, tal y como lo permite el sistema remoto (Unix, Windows, MVS o VAX). <p>Devuelve</p> <ul style="list-style-type: none"><code>{URL}</code> URL que se puede utilizar para recuperar el archivo desde el servidor de sesión de Host Access for the Cloud.
<code>setTransferOptions(options)</code>	<p>Establece las opciones de transferencia para la sesión FileTransfer actual. Las opciones de transferencia se aplican a todas las transferencias futuras cuando o bien se cierra la sesión, o bien ésta se sobrescribe con otra llamada a <code>setTransferOptions</code>.</p> <p>Parámetros</p> <ul style="list-style-type: none"><code>{JSON}</code> Consulte FileTransferOptions para conocer los nombres y valores permitidos. <p>Devuelve</p> <ul style="list-style-type: none"><code>{Promise}</code> Se cumple cuando finaliza la llamada. Se rechaza si se ha producido un error al configurar las opciones.

`cancelar()`

Cancela la transferencia actual en curso.

Parámetros

`{String}` Nombre completo de archivo remoto, tal y como lo permite el sistema remoto (Unix, Windows, MVS o VAX).

Devuelve

`{Promise}` Se cumple cuando finaliza la llamada. Se rechaza si se ha producido un error al cancelar la transferencia.

FileTransferFactory

Un objeto `fileTransferFactory` está disponible para todas las macros. Si se han configurado transferencias de archivos para la sesión, puede utilizarlas para obtener una referencia a un objeto `FileTransfer`.

Método	Descripción
<code>getIND\$File()</code>	<p>Devuelve un objeto <code>FileTransfer</code> para interactuar con el tipo <code>Ind\$File</code> configurado para la sesión.</p> <p>Devuelve</p> <p><code>{FileTransfer}</code></p> <p>Genera</p> <p><code>{Error}</code> Si la sesión no se ha configurado para permitir transferencias <code>IND\$File</code>.</p>

FileTransferOptions

Especificación del objeto de opción de transferencia de archivos. Ejemplo:

```
`fileTransfer.setTransferOptions({ transferMethod : 'ascii' });``
```

Método	Descripción
--------	-------------

```
transferMethod    {String} Valores permitidos:
                  'ascii'
                  'binario'
```

HostFile

Un objeto HostFile representa un archivo en el sistema de archivos del host.

Método	Descripción
<code>getName()</code>	Obtiene el nombre de archivo. Devuelve <code>{String}</code> El nombre de archivo.
<code>getParent()</code>	Obtiene el creador de este archivo del host. Devuelve <code>{String}</code> El elemento principal de este archivo del host. Esto significa cosas diferentes en tipos de host diferentes. Por ejemplo, en TSO éste es el nombre del catálogo en el que reside el archivo.
<code>getSize()</code>	El tamaño en bytes del archivo. Devuelve <code>{Number}</code> El tamaño del archivo en bytes.
<code>getType()</code>	El tipo de archivo representado. Devuelve

HostFileType

El objeto HostFileType define constantes para determinar el tipo de un objeto HostFile.

Valor	Descripción
FILE	Representa un archivo en el sistema de host.
DIR	Representa un directorio en el sistema de host.

Valor	Descripción
DESCONOCIDO	Representa un archivo del host de origen desconocido.

OIA

Interfaz Operator Information Area (OIA). El objeto OIA devuelve valores que se han definido en el objeto OIAStatus.

Método	Descripción
<code>getStatus ()</code>	Devuelve el conjunto de indicadores de estado habilitado. Consulte StatusSet . Devuelve {StatusSet} Contiene la información de estado.
<code>getCommErrorCode ()</code>	Devuelve el código de error de comunicación actual. Devuelve {Number} El código de error de comunicación actual. Si no existe, será 0.
<code>getProgErrorCode ()</code>	Devuelve el código de error del programa actual.. Devuelve {Number} El código de error de programa actual. Si no existe, será 0.

OIAStatus

OIAStatus	Descripción
CONTROLLER_READY	Controlador listo
A_ONLINE	Online con una conexión no-SNA
MY_JOB	Conectada a una aplicación de host
OP_SYS	Conectada a SSCP (SNA)
UNOWNED	No conectada
TIME	Teclado inhibido

OIAStatus	Descripción
SYS_LOCK	Bloqueo del sistema tras tecla AID
COMM_CHECK	Prueba de comunicación
PROG_CHECK	Prueba de programa
ELSEWHERE	Pulsación de tecla no válido en la posición del cursor
FN_MINUS	Función no disponible
WHAT_KEY	Pulsación de tecla no válido
MORE_THAN	Demasiados caracteres ingresados en el campo
SYM_MINUS	Símbolo introducido no disponible
INPUT_ERROR	Error de entrada de operador (5250 sólo)
DO_NOT_ENTER	No introducir
INSERT	Cursor en modo insertar
GR_CURSOR	Cursor en modo gráfico
COMM_ERR_REM	Recordatorio de error de comunicación
MSG_WAITING	Indicador de mensaje en espera
ENCRYPT	La sesión está cifrada
NUM_FIELD	Carácter no válido en campo sólo numérico

Posición

Método	Descripción
---------------	--------------------

`Position(row,col)`

Crea una nueva instancia Position.

Parámetros`{Number}` Coordenada de fila de la pantalla de filas.`{Number}` Coordenada de columna de la pantalla de columnas.

PresentationSpace

Utilice el objeto PresentationSpace para interactuar con la pantalla del terminal. Entre las interacciones disponibles están ajustar y obtener la posición del cursor y la lectura de texto.

Método	Descripción
<code>getCursorPosition()</code>	Devuelve una instancia de Position que representa la posición actual del cursor. Una sesión no conectada tiene una posición de cursor de 0,0. Devuelve <code>{Position}</code> Ubicación actual del cursor.
<code>setCursorPosition(position)</code>	Mueve el cursor del host a la posición de fila y columna especificado. En algunos hosts, como los VT, el host puede restringir los movimientos del cursor. Parámetros <code>{Position}</code> Position Nueva posición del cursor. Devuelve None Genera <code>{RangeError}</code> Si la posición no es válida en la pantalla actual.
<code>isCursorVisible()</code>	Comprueba que el cursor está actualmente visible en el espacio de presentación. El cursor se considera no visible si la sesión no está conectada. Devuelve <code>{Boolean}</code> "True" (verdadero) si el cursor está visible. False si el cursor no está visible.

`sendKeys (keys
)`

Transmite una cadena de texto o `ControlKey` al host en la posición actual del cursor en el espacio de presentación. Si el cursor no se encuentra en la posición deseada, utilice primero la función `setCursorPosition`. La cadena de texto puede contener cualquier número de caracteres y objetos `ControlKey`.

Por ejemplo: "myname" + `ControlKey.TAB` + "mypass" + `ControlKey.ENTER` transmitirá un ID de usuario, tabulará al campo siguiente, transmitirá una contraseña y, a continuación, transmitirá la tecla Intro. Si necesita transmitir un corchete, duplique los corchetes ([[o]]).

Parámetros

`{String}` Texto de teclas o teclas de control que se transmitirán.

`getText (start
,length)`

Devuelve una cadena que representa un área lineal del espacio de presentación. Cuando se encuentran los límites de la fila, no se insertan caracteres de nueva línea.

Parámetros

`{Position}` Posición inicial desde la que se debe recuperar el texto.

`{Number}` Longitud del número máximo de caracteres que se devolverán. Si el parámetro de longitud provoca que se supere la última posición del espacio de presentación, solo se devuelven los caracteres hasta la última posición.

Devuelve

`{String}` Representa un área lineal del espacio de presentación que puede estar vacía si la sesión no está conectada.

Genera

`{RangeError}` Si la posición o la longitud no son válidas en la pantalla actual.

`getSize()`

Obtiene las dimensiones de la pantalla como objeto `Dimension`.

Devuelve

`{Dimension}` Contiene el número de filas y columnas. El tamaño de la pantalla es `[row:0, col:0]` si la sesión no está conectada.

`getDataCells (start, length)` Devuelve instancias de [DataCell](#) en las que el primer miembro será para la posición especificada por el parámetro de inicio. El número máximo de instancias DataCell en la lista viene especificado por el parámetro de longitud.

Parámetros

`{Position}` Inicio de la primera posición en la pantalla del host en la que se recuperan instancias de DataCell. Consulte [Position](#).

`{Number}` Longitud del número máximo de instancias de DataCell que se van a recuperar. Si no se especifica, devuelve DataCells de la posición inicial a la posición final de la pantalla.

Devuelve

`{DataCell[]}` Instancias que pueden estar vacías si no se ha conectado la sesión. Si la posición no está especificada, devuelve todas las DataCells. Si la longitud no se especifica, devuelve DataCells de la posición inicial a la posición final de la pantalla.

Genera

`{RangeError}` Si el inicio o la longitud están fuera del rango.

`getFields()` Devuelve una lista de los campos en el espacio de presentación. Si el tipo de host no soporta campos o si la pantalla actual no está formateada el valor de retorno será siempre una lista vacía. Consulte [FieldList](#).

Devuelve

`{FieldList}` Para los campos definidos del host en el espacio de presentación.

Sesión

El objeto Session es el punto de entrada principal para acceder al host. Contiene las funciones para conectar, desconectar y obtener el objeto PresentationSpace.

Método	Descripción
<code>connect()</code>	<p>Conecta con el host configurado. Si es necesario, utilice <code>wait.forConnect()</code> para bloquear la ejecución de la macro hasta que la sesión esté conectada.</p> <p>Devuelve None .</p>

<code>disconnect()</code>	<p>Se desconecta del host configurado. Si es necesario, utilice <code>wait.forDisconnect()</code> para bloquear la ejecución de la macro hasta que la sesión esté conectada.</p> <p>Devuelve <code>None</code>.</p>
<code>isConnected()</code>	<p>Determina si la conexión con el host está establecida o no.</p> <p>Devuelve <code>{Boolean}</code> "True" (verdadero) si se ha establecido la conexión con el host; de lo contrario, "False" (falso).</p>
<code>getPresentationSpace()</code>	<p>Proporciona acceso a la instancia de <code>PresentationSpace</code> para esta sesión.</p> <p>Devuelve <code>{PresentationSpace}</code> Instancia asociada a esta sesión.</p>
<code>getDeviceName()</code>	<p>Devuelve el nombre del dispositivo para una sesión conectada o una cadena vacía si la sesión está desconectada o no tiene un nombre de dispositivo.</p> <p>Devuelve <code>{String}</code> El nombre del dispositivo conectado.</p>
<code>getType()</code>	<p>Devuelve el tipo de sesión de host. Véase SessionType.</p> <p>Devuelve <code>{String}</code> El tipo de sesión del host.</p>
<code>setDeviceName()</code>	<p>Aporta un medio para modificar el nombre de dispositivo en una instancia de sesión.</p> <p>Parámetros <code>{String}</code> name Nombre del dispositivo que se utilizará al conectarse a un host.</p> <p>Genera <code>{Error}</code> Si se intenta configurar el nombre de dispositivo mientras la sesión está conectada.</p>

`getOIA()` Proporciona acceso a la instancia de [OIA](#) para esta sesión.
Devuelve
`{OIA}` OIA asociada a este sesión.

SessionType

Constantes utilizadas para identificar el tipo de host al que se está realizando la conexión. Consulte el objeto [Session](#).

Tipos de host disponibles:

IBM_3270

IBM_5250

VT

ALC

UTS

T27

StatusSet

Puede utilizar el objeto StatusSet para descifrar el estado de OIA. El objeto StatusSet devuelve valores definidos en el objeto [OIAStatus](#) y, cuando se utilizan juntos, se puede obtener información de estado del OIA.

Método	Descripción
<code>contains(status Flag)</code>	<p>Determina si el conjunto contiene el indicador de estado especificado de constantes de OIAStatus.</p> <p>Parámetros <code>{Number}</code> Estado de statusFlag que se comprobará.</p> <p>Devuelve <code>{Boolean}</code> "True" (verdadero) si el indicador de estado está presente en el conjunto.</p>
<code>isEmpty()</code>	<p>Determina si el conjunto de estados está vacío.</p> <p>Devuelve <code>{Boolean}</code> "True" (verdadero) si el conjunto está vacío.</p>

<code>size()</code>	Indica el número de indicadores de estado en el conjunto. Devuelve {Number} El número de estados.
<code>toArray()</code>	Convierte el conjunto de estados interno en una matriz. Devuelve {Object []} Matriz de indicadores de estado del conjunto.
<code>toString()</code>	Convierte el conjunto de estados interno en una cadena. Devuelve {String} Nombres delimitados por espacios de indicadores de estado en el conjunto.
<code>forEach(callback, thisArg)</code>	Función para iterar sobre cada elemento en el conjunto de estados. Parámetros {forEachCallback} Devolución de llamada para realizar la operación específica. Se llama con el nombre de cada estado en el conjunto.
<code>forEachCallback(string, thisArg)</code>	Una función de devolución de llamada suministrada por el usuario en la que se proporciona el comportamiento que se utilizará como parámetro de devolución de llamada a forEach. Parámetros {String} String El nombre de un estado en el conjunto de estado. {Object} thisArg Puntero opcional a un objeto de contexto.

Interfaz de usuario

El objeto de interfaz de usuario provee funciones para interactuar con el usuario, para preguntar por información básica y visualizarla. El objeto UI está disponible automáticamente en su macro como la variable "ui".

nota

Importante: Todas las funciones de la interfaz de usuario requieren la palabra clave "yield" delante de ellas. Esto permite bloquear la ejecución de la macro hasta que se cumplan las condiciones para la función UI.

[parameter] denota un parámetro opcional.

Método	Descripción
--------	-------------

<pre>prompt(message, [defaultAnswer], [mask])</pre>	<p>Se pregunta al usuario por información en la interfaz de usuario.</p> <p>Parámetros</p> <p>{String} Título del mensaje que se mostrará al usuario. Por defecto: cadena vacía.</p> <p>{String} Respuesta por defecto que se utilizará si el usuario la deja en blanco. Valor por defecto: cadena en blanco</p> <p>{Boolean} La máscara indica si se debe ocultar la solicitud (como con una contraseña).</p> <p>Devuelve</p> <p>{Promise} Se cumple si el usuario cierra la ventana del diálogo. Devuelve la entrada del usuario con "OK" o cero con "Cancel".</p>
<pre>message([message])</pre>	<p>Muestra un mensaje en la interfaz de usuario.</p> <p>Parámetros</p> <p>{String} Mensaje que se mostrará al usuario. Por defecto: cadena vacía.</p> <p>Devuelve</p> <p>{Promise} Se cumple cuando el usuario cierra la ventana del mensaje.</p>

Wait

Utilice el objeto wait para esperar una sesión particular o un estado de pantalla. Por ejemplo, puede esperar hasta que el cursor se encuentre en una posición particular o hasta que haya texto presente en una posición determinada antes de continuar con la ejecución de la macro.

Las funciones de espera se utilizan frecuentemente en combinación con funciones asíncronas como connect() y sendKeys().

nota

Todas las funciones tienen tiempo límite como parámetro opcional y tienen un valor de tiempo límite por defecto de 10 segundos (10 000 ms).

Importante: Todas las funciones de espera requieren la clave 'yield' enfrente de ellas. Esto permite bloquear la ejecución de la macro hasta que se cumplan las condiciones para la función de espera.

[parameter] denota un parámetro opcional.

Método	Descripción
<code>setDefaultTimeout(timeout)</code>	<p>Establece el valor de tiempo límite por defecto para todas las funciones.</p> <p>Parámetros <code>{Number}</code> Tiempo límite por defecto que se utilizará para todas las funciones de espera en milisegundos.</p> <p>Devuelve <code>{None}</code></p> <p>Genera <code>{RangeError}</code> Si el tiempo límite especificado es inferior a cero.</p>
<code>forConnect([timeout])</code>	<p>Espera una solicitud de conexión para completar.</p> <p>Parámetros <code>{Number}</code> Tiempo en milisegundos.</p> <p>Devuelve <code>{Promise}</code> Se cumple si la sesión ya está conectada o cuando se realiza la conexión. Rechazada si expira el tiempo de espera.</p>
<code>forDisconnect([timeout])</code>	<p>Espera una solicitud de desconexión para completar.</p> <p>Parámetros <code>{Number}</code> Tiempo límite en milisegundos.</p> <p>Devuelve <code>{Promise}</code> Se cumple si la sesión ya está desconectada o cuando al fin se desconecta. Rechazada si expira el tiempo de espera.</p>
<code>forFixedTime([timeout])</code>	<p>Espera de forma incondicional un tiempo fijo. Tiempo en milisegundos (ms).</p> <p>Parámetros <code>{Number}</code> Tiempo límite en milisegundos.</p> <p>Devuelve <code>{Promise}</code> Se cumple una vez transcurrido el tiempo.</p>

```
forScreenChange([timeout])
```

Espera a que cambie la pantalla de host. Esta función se devuelve cuando se detecta una actualización de pantalla. No ofrece garantías sobre el número de actualizaciones posteriores que pueden recibirse antes de que se complete la pantalla. Es aconsejable esperar repetidamente hasta que el contenido de la pantalla coincida con algunos criterios de detención conocidos.)

Parámetros

{Number} Tiempo límite en milisegundos.

Devuelve

{Promise} Se resuelve si la pantalla cambia. Rechazada si expira el tiempo de espera.

```
forCursorPosition([timeout])
```

Espera a que el cursor llegue a la posición especificada.

Parámetros

{Position} La posición que especifica la fila y la columna.

Devuelve

{Promise} Se cumple si el cursor ya se ha ubicado o si se ubica finalmente. Rechazada si expira el tiempo de espera.

```
forText(text, position, [timeout])
```

Espera a que el texto se encuentre en una posición específica de la pantalla.

Parámetros

{String} El texto previsto.

{Position} La posición que especifica la fila y la columna.

{Number} Tiempo límite en milisegundos.

Devuelve

{Promise} Se cumple si el texto ya se encuentra en la posición especificada o en cualquier otra ubicación. Rechazada si expira el tiempo de espera.

Genera

{RangeError} Si la posición no es válida.

```
forHostPrompt
(text,
column,
[timeout])
```

Espera a que un símbolo de sistema esté colocado en una columna específica de la pantalla.

Parámetros

{String} Solicitud de texto prevista.

{Number} Columna en la que se espera que se encuentre el cursor.

{Number} Tiempo límite en milisegundos.

Devuelve

{Promise} Se cumple si ya se dan las condiciones o cuando al fin se dan. Rechazada si expira el tiempo de espera.

Genera

{RangeError} Si la columna está fuera del rango.

```
forHostSettle
([settleTime]
,[timeout])
```

NOTA: `wait.forHostSettle` solo debe utilizarse cuando otras funciones de espera adicionales específicas son insuficientes.

Supervisa los datos entrantes de la pantalla; resuelve `settleTime` ms tras la última actualización **y** cuando el teclado está desbloqueado. Esta función es útil cuando los datos llegan en varios paquetes y desea asegurarse de que se ha recibido toda la pantalla antes de continuar.

Parámetros

{Number} Tiempo que se esperará después de la última actualización para asegurarse de que no se reciban datos de forma inesperada. El valor por defecto es 200 milisegundos.

{Number} tiempo de espera en milisegundos.

Devuelve

{Promise} Se cumple cuando ha transcurrido el tiempo de establecimiento tras la recepción de la última actualización de la pantalla y el teclado está desbloqueado.

Ejemplos de Macros

Para ayudarle a crear correctamente macros que se beneficien de todas las funciones del Editor de macros, dispone de estos ejemplos como punto de partida.

Interacción Básica con el Host

Este ejemplo muestra la interacción básica con el host, incluyendo:

Enviar datos al host

Esperar pantallas a mostrar

Utilizar la palabra clave `yield` para esperar funciones asíncronas

Leer texto de la pantalla

Mostrar información básica al usuario

Tratamiento de errores básicos

Todas las macros tienen disponibles los siguientes objetos de forma predeterminada:

1. **session:** punto de entrada principal para acceder al host. Puede conectar, desconectar y facilitar acceso al `PresentationSpace`.

El objeto `PresentationSpace` obtenido de la sesión representa la pantalla y proporciona muchas funciones comunes, como obtener y establecer la ubicación del cursor, enviar datos al host y leer de la pantalla.

2. **wait:** facilita una forma sencilla de esperar a varios estados del host antes de seguir enviando más datos o leer de la pantalla.
3. **UI:** ofrece funciones básicas de interfaz de usuario. Muestra datos al usuario o le pide información.


```

// Función Crear una nueva macro
var macro = createMacro(function*(){
  'use strict';

  // Todas las macros tienen disponibles los siguientes objetos de forma
  // predeterminada:
  // 1. session - Punto de entrada principal para acceder al host. Puede
  // conectar, desconectar y facilitar acceso al PresentationSpace.
  // El objeto PresentationSpace obtenido de la sesión representa la
  // pantalla y provee capacidades muy comunes como obtener y ajustar la
  // posición del cursor, enviar datos al host y leer de la pantalla.
  // 2. wait - Facilita una forma sencilla de esperar a varios estados del host
  // antes de seguir enviando más datos o leer de la pantalla.
  // 3. uiI - Provee capacidades básicas de interfaz de usuario. Mostrar datos
  // al usuario o pedirle información.

  // Declarar una variable para leer y visualizar algunos datos de pantalla.
  // La mejor práctica es declarar todas las variables cerca de la parte
  // superior de una función.
  var numberOfAccounts = 0;

  // Iniciar obteniendo el objeto PresentationSpace, que provee muchas
  // operaciones de pantalla comunes.
  var ps = session.getPresentationSpace();

  try {
    // Puede ajustar y obtener la posición del cursor
    ps.setCursorPosition(new Position(24, 2));

    // Utilizar la función sendKeys para enviar caracteres al host
    ps.sendKeys('cics');

    // SendKeys se utiliza también para enviar teclas de host como teclas PA
    // y PF.
    // Véase "Control Keys" en la documentación para todas las opciones
    // disponibles
    ps.sendKeys(ControlKey.ENTER);

    // Esperar a que el cursor se encuentre en la posición correcta.
    // El objeto wait provee varias funciones para esperar a que ocurran
    // determinados estados
    // de modo que usted pueda proceder o bien a enviar más teclas, o bien a
    // leer datos de la pantalla.
    yield wait.forCursor(new Position(24, 2));

    // Puede mezclar caracteres y teclas de control en una llamada sendKeys.

```

```

    ps.sendKeys('data' + ControlKey.TAB + ControlKey.TAB + 'more data' +
ControlKey.ENTER);

    // La palabra clave "yield" se debe utilizar enfrente de todas las
llamadas de función "wait" y "ui".
    // Le dice al navegador que detenga la ejecución de la macro hasta que la
// función wait (asíncrona) vuelva. Consulte la documentación para saber
qué funciones
    // requieren la palabra clave yield.
yield wait.forCursor(new Position(10, 26));
ps.sendKeys('accounts' + ControlKey.ENTER);

    // Puede esperar también a que aparezca un texto en ciertas áreas de la
pantalla
yield wait.forText('ACCOUNTS', new Position(3, 36)) ;
ps.sendKeys('1' + ControlKey.ENTER);

    // Todas las funciones wait excederán el tiempo de espera si no se
cumplen los criterios dentro de un límite de tiempo.
    // Puede incrementar tiempos de espera con un parámetro opcional en las
funciones wait (en milisegundos)
    // Todos los tiempos de espera se especifican en milisegundos y el valor
predeterminado es 10 segundos (10000 ms).
yield wait.forCursor(new Position(1, 1), 15000);
ps.sendKeys('A' + ControlKey.ENTER);

    // PS proporciona la función getText para leer texto de la pantalla
numberOfAccounts = ps.getText(new Position(12, 3), 5);

    // Utilizar el objeto ui para visualizar algunos datos de la pantalla
ui.message('Número de cuentas activas: ' + numberOfAccounts);

    // La try / catch permite capturar todos los errores y notificarlos a una
ubicación central
} catch (error) {
    // De nuevo, utilizamos el objeto ui para visualizar un mensaje que
indica que se ha producido un error
yield ui.message('Error: ' + error.message);
}
//Fin Macro Generada
});

// Ejecutar la macro y devolver los resultados al Ejecutor de macros
// La instrucción return es necesaria, ya que la aplicación aprovecha
// esto para saber si la macro se ha ejecutado correctamente y cuándo ha
finalizado
return macro();

```