WPS Communicate

guía del usuario y referencia

Versión: 4.1.2

www.worldprogramming.com

Tabla de contenidos

Información general....3

Instalación y configuración....4

Guía del usuario de cliente....7
Conceptos clave....7
Uso de WPS Communicate....7
Un programa simple....7
Uso de PROC DOWNLOAD....9
Uso de PROC UPLOAD....11
Ejecución de subprogramas en paralelo....13
Transferencia de las variables de macro....15
Referencia....16
Cómo leer los diagramas de sintaxis del ferrocarril....16
Instrucciones globales....18
Instrucciones del procesador de macros....28
Procedimientos....29
Opciones del sistema....37

Guía de administración del sistema....41
Autenticación en z/OS a través de Telnet....41
Ejemplo de script de autenticación Telnet....46
SSH (Secure Shell) desde un cliente Windows....47
Autenticación de contraseña (usando PuTTY) e inicio de sesión de WPS....49
Autenticación de clave pública....55
SSH (Secure Shell) desde un cliente UNIX....69
Autenticación de contraseña e inicio de sesión de WPS....69
Autenticación de clave pública....70
Inicio de sesión único de Kerberos....78

Variables de entorno....80

Avisos legales....81

Información general

WPS Communicate permite que diferentes partes seleccionadas de la lógica del programa SAS se ejecuten en diferentes hosts y permite la transferencia de datos y resultados entre ellos. Puede ser conveniente pensar en **WPS Communicate** como programas de habilitación para acercarse a los datos en los que operan, a diferencia del patrón más convencional en el que los datos se transfieren al programa.

En una sesión WPS normal, todo el procesamiento se realiza en un único host local, de manera predeterminada, de forma sincrónica. Sin embargo, puede ser ventajoso ejecutar diferentes partes de un programa en diferentes hosts. Por ejemplo, es posible que tenga una aplicación de informes que necesite extraer y resumir algunos datos antes de generar y distribuir informes. Podría tener sentido ejecutar las partes de extracción y resumen del programa en el host que almacena los datos, transfiriendo los datos resumidos de vuelta a la plataforma local para la generación y distribución de los informes.

Junto con el código de programa, los conjuntos de datos necesarios para el procesamiento remoto se pueden cargar y descargar desde un programa que se ejecuta localmente. Esto hace posible realizar un trabajo intensivo de base de datos en un servidor central, antes de volver a la plataforma local para el procesamiento de los resultados.

A partir de la versión 3.2, **WPS Communicate** tiene la capacidad de operar de forma asincrónica, por lo que el procesamiento inicial en múltiples hosts remotos ocurre en paralelo, permitiendo mejoras significativas de rendimiento para una gran clase de cargas de trabajo.

Nota:

Debido a que es necesario cambiar el código de programa para identificar las partes que se deben ejecutar de forma remota, **WPS Communicate** requiere que los usuarios estén familiarizados con el lenguaje de SAS.

Este manual se divide en las siguientes partes:

- *Instalación y configuración* (pág. 4).
- La *Guía del usuario de cliente* (pág. 7) que está dirigida a los usuarios de **WPS Communicate** en el día a día.
- La *Guía de administración del sistema* (pág. 41) que está dirigida a los técnicos informáticos que instalan y administran el sistema.

Instalación y configuración

Hay 3 elementos necesarios para una instalación de cliente/servidor de **WPS Communicate**: un medio de autenticación, el software WPS en sí y suficientes claves de licencia de WPS para abarcar la configuración.

Autenticación

WPS Communicate requiere el uso de una conexión SSH o Telnet entre los equipos servidor y cliente. Los métodos de autenticación disponibles, incluida la autenticación de clave pública (con y sin el uso de un agente de llavero) y cualquier software adicional requerido, se describen en el *Guía de administración del sistema* (pág. 41).

El protocolo recomendado depende del tipo de host al que se esté realizando una conexión:

- Telnet (consulte *Autenticación en z/OS a través de Telnet* (pág. 41))

  Nota:

  Para un host z/OS, Telnet (o TN3270) es el mecanismo de conexión más ampliamente utilizado y es el método recomendado para conectarse a hosts z/OS usando **WPS Communicate**. Es el único mecanismo compatible que permite el acceso directo a TSO (Time Sharing Option). Es posible conectarse a z/OS a través de SSH (Secure Shell), pero esto lo conecta con USS (UNIX System Services) en lugar de TSO, y esto no siempre está configurado en sistemas z/OS.

- SSH (Secure Shell). Esto no se proporciona con el software WPS. Sin embargo:

  1. Para un equipo servidor UNIX (Linux, Solaris o AIX), se puede utilizar el demonio SSH incorporado.
  2. Para un equipo servidor Windows, la instalación SSH de terceros está disponible por separado de Bitvise (consulte la *Guía de administración del sistema* (pág. 41) para detalles).

     Nota:

     Bitvise SSH es la única instalación SSH oficialmente compatible con World Programming para un equipo servidor Windows.

Software WPS

WPS se proporciona como un único archivo de instalación que contiene todas las funciones de WPS necesarias para su uso con **WPS Communicate**, incluyendo **WPS Workbench**, **Java Runtime Environment** y el componente **WPS Server** que se puede sujetar a licencia.

Nota:

Necesitará archivos de instalación de WPS que sean adecuados para los sistemas operativos de los equipos servidor y cliente. Por ejemplo, si tiene un equipo servidor Linux y equipos cliente Windows, necesitará el archivo de instalación para Linux, para el servidor, y los archivos de instalación apropiados para Windows (32 bits o 64 bits) para los clientes. Consulte la guía de instalación de la plataforma pertinente para detalles completos del proceso de instalación de WPS.

Claves de licencia de WPS

Para que una instalación de WPS pueda ejecutar un programa escrito en el lenguaje de SAS, el componente **WPS Server** debe activarse mediante la aplicación de una clave de licencia de WPS (proporcionada por separado al software WPS).

Nota:

Se requiere una instalación de WPS con licencia completa en cada cliente y en cada servidor host.

Resumen de instalación

A continuación, se ofrece un breve resumen de la secuencia de pasos necesarios para instalar y configurar una solución cliente/servidor de **WPS Communicate**.

Importante:

La persona que instala WPS y aplica las claves de licencia debe tener privilegios de administrador del sistema operativo en esas máquinas.

1. Fuera de WPS, configure y pruebe la conexión SSH o Telnet entre los equipos servidor y cliente, de acuerdo con las plataformas y los métodos de autenticación que están activos en su sitio (consulte la *Guía de administración del sistema* (pág. 41) para detalles).
2. Instale WPS completamente en el equipo servidor o en el mainframe z/OS.

   Nota:

   Asegúrese de que tiene su clave de licencia para la instalación del servidor o z/OS. Al iniciar WPS en un servidor, se le pedirá automáticamente que aplique la licencia. Para un mainframe z/OS, no se le pedirá y por lo tanto deberá seguir las instrucciones sobre la aplicación de una clave de licencia que se pueden encontrar en el documento separado "Guía de instalación y del usuario de WPS para z/OS".
3. Si aún no lo ha hecho, instale WPS en los equipos cliente individuales.

   Nota:

   Asegúrese de que tenga su clave de licencia de estación de trabajo, así que pueda aplicarse a cada cliente a su vez. Al iniciar WPS en los clientes, se le pedirá automáticamente que aplique la licencia.

Variables de entorno

La configuración de las variables de entorno no es necesaria antes de la sesión inicial de **WPS Communicate**, pero podría querer pedirle al administrador del sistema que las ajuste para sesiones posteriores. Consulte *Variables de entorno* (pág. 80).

Guía del usuario de cliente

Conceptos clave

Para transferir el flujo de control desde un programa que se ejecuta localmente a un servidor remoto, **WPS Communicate** ha introducido los siguientes dos pares de instrucciones: `SIGNON... SIGNOFF` y `RSUBMIT... ENDRSUBMIT`.

`SIGNON` y `SIGNOFF` deben encerrar las instrucciones `RSUBMIT` y `ENDRSUBMIT`:

```
SIGNON ...;
...
RSUBMIT;
...
ENDRSUBMIT;
...
SIGNOFF;
```

La instrucción `SIGNON...` es responsable de iniciar y autenticar su sesión con el servidor.

Entre las instrucciones `RSUBMIT` y `ENDRSUBMIT`, inserta el código de programa que desea ejecutar en la máquina remota.

Una vez que se haya ejecutado el código remoto, la instrucción `SIGNOFF` cierra la conexión y libera sus recursos.

Uso de WPS Communicate

Un programa simple

Esta sección contiene un ejemplo de un programa **WPS Communicate** muy simple que se ejecuta de forma remota en una plataforma Linux.

Antes de comenzar, asegúrese de que **WPS Workbench** esté instalado y en ejecución, y que haya iniciado sesión correctamente en el servidor remoto mediante un cliente SSH externo.

Este programa utiliza el inicio de sesión de contraseña simple para conectarse al servidor remoto. No es el método más seguro, ya que almacena un ID de usuario y una contraseña en texto sin formato en el código fuente, y se recomienda el uso de la autenticación `Public key` (es decir, `password-less`) para el uso a largo plazo (consulte *Guía de administración del sistema* (pág. 41)). Sin embargo, el inicio de sesión de contraseña sirve para verificar que **WPS Communicate** está funcionando en su configuración más básica.

1. Cree un nuevo programa en **WPS Workbench** de la manera siguiente:

```
SIGNON <servername> SSH
username="<username>"
password="<password>"
LAUNCHCMD="<path to WPS executable>";
RSUBMIT;
%PUT &SYSHOSTNAME;
%PUT 'Success with simple password sign-on';
ENDRSUBMIT;
SIGNOFF;
```

   Asegúrese de sustituir `<servername>` con el nombre de su servidor remoto y, de manera similar, reemplace `<username>` y `<password>` con su ID de usuario y contraseña reales en la máquina remota.

   Nota:

   La opción `LAUNCHCMD` debe apuntar a la ubicación del archivo ejecutable de WPS en el servidor remoto, por ejemplo `/home/installs/wps32/bin/wps -dmr`.

2. Ejecute el programa y examine el registro de salida:

```
367        ODS _ALL_ CLOSE;
368        FILENAME WPSWBHTM TEMP;
369        ODS HTML(ID=WBHTML) BODY=WPSWBHTM GPATH="C:\Users\techwriter\AppData
\Local\Temp\WPS Temporary
369      !  Data\_TD5876";
NOTE: Writing HTML(WBHTML) Body file WPSWBHTM
370        SIGNON DOCSERVER SSH
371        username="XXXX"
372        password=XXXXXXXXXX
373        LAUNCHCMD=<path to WPS executable>";
NOTE: Remote SSH signon to DOCSERVER starting
NOTE: Establishing tunnelled connection to DOCSERVER:55765
NOTE: (c) Copyright World Programming Limited 2002-2015.  All rights reserved.
NOTE: World Programming System 3.02 (03.02.00.00.011866)
      Licensed to World Programming Company Ltd
NOTE: This session is executing on the Linux platform and is running in 64-bit
 mode

NOTE: Remote signon to DOCSERVER complete
374        RSUBMIT;
NOTE: Remote submit to DOCSERVER starting
1          %PUT &SYSHOSTNAME;
harmony.teamwpc.local
2          %PUT 'Success with simple password sign-on';
'Success with simple password sign-on'
NOTE: Remote submit to DOCSERVER complete
375        SIGNOFF;
NOTE: Remote signoff from DOCSERVER starting
```

```
NOTE: Submitted statements took :
      real time : 0.130
      cpu time  : 0.028
NOTE: Remote signoff from DOCSERVER complete
376       quit; run;
377       ODS _ALL_ CLOSE;
```

- Las opciones `SSH`, `username`, `password` y `LAUNCHCMD` de la instrucción `SIGNON` proporcionan toda la información necesaria para iniciar sesión en el host remoto a través de SSH e iniciar el servidor de WPS remoto.
- Entre las instrucciones `RSUBMIT` y `ENDRSUBMIT`, se ejecutan las siguientes dos líneas de código en la máquina remota:

  ```
  %PUT &SYSHOSTNAME;
  %PUT 'Success with simple password sign-on';
  ```

  Nota:

  Las instrucciones `%PUT` escriben en el archivo de registro local, pero la variable de macro `&SYSHOSTNAME` se resuelve en la máquina remota (en `DOCSERVER` en este caso).

- Finalmente, la instrucción `SIGNOFF` elimina la conexión.

Uso de PROC DOWNLOAD

`PROC DOWNLOAD` está dirigido a transferir bibliotecas, conjuntos de datos o archivos desde un host remoto, utilizando respectivamente las opciones `INLIB`, `DATA` y `INFILE`, a la plataforma local, utilizando las opciones `OUTLIB`, `OUT` y `OUTFILE`. `PROC DOWNLOAD` se debe colocar dentro de un bloque de código `RSUBMIT... ENDRSUBMIT`.

Los siguientes son algunos ejemplos de sintaxis:

```
/* transfer a library */
PROC DOWNLOAD INLIB=remotelib OUTLIB=locallib; RUN;
```

```
/* transfer a single dataset */
PROC DOWNLOAD DATA=remotelib.dataset OUT=locallib.dataset; RUN;
```

```
/*transfer a file */
PROC DOWNLOAD INFILE=”remote_host_file_path” OUTFILE=”local_platform_file_path”;
 RUN;
```

Nota:

Para la sintaxis completa de este procedimiento, consulte la sección **Referencia**.

Ejemplo del uso de PROC DOWNLOAD

Esta sección contiene un ejemplo de uso de `PROC DOWNLOAD`.

1. Cree un nuevo programa en **WPS Workbench** copiando y pegando el código siguiente:

```
/*******************************************************************
Sign on to the remote host
*******************************************************************/
signon <servername> ssh
user='<username>'
password='<password>'
launchcmd='<path to WPS executable>';

/*******************************************************************
Create and populate a small dataset on the remote host
*******************************************************************/
rsubmit;
data communicatedemo;
input movie $ 1-46 year $ 48-51;
cards;
Avatar                                        2009
Titanic                                       1997
The Avengers                                  2012
Harry Potter and the Deathly Hallows - part 2 2011
Frozen                                        2013
;
run;

/*******************************************************************
Download the generated dataset to the local platform
*******************************************************************/
proc download
data=WORK.communicatedemo
out=WORK.communicatedemo;
run;

/*******************************************************************
Complete the remote program execution and close the connection
*******************************************************************/
endrsubmit;
signoff;
```

Asegúrese de sustituir `<servername>` por el nombre del servidor remoto, reemplace `<username>` y `<password>` según corresponda y configure `launchcmd` para que apunte a la ubicación del archivo ejecutable de WPS en el servidor remoto, por ejemplo `/home/installs/wps-3.2/bin/wps -dmr`. Este programa está destinado a crear un conjunto de datos en el host remoto, enumerando 5 de las películas de mayor recaudación de todos los tiempos, y descargarlo a su plataforma local.

2. Ejecute el programa y examine el registro de salida, un mensaje debajo de la invocación `PROC DOWNLOAD` explica la transferencia de datos:

```
NOTE: Dataset download in progress from WORK.communicatedemo to
 WORK.communicatedemo
NOTE: 333 bytes were transferred to dataset WORK.communicatedemo at 333000 bytes/
sec
NOTE: Dataset "WORK.communicatedemo" has 5 observation(s) and 2 variable(s)
```

   El programa se ejecuta, creando el conjunto de datos en la máquina remota y descargándolo a la plataforma local. Puede examinar el conjunto de datos `communicatedemo` en su biblioteca `WORK` local seleccionando **Servidor local ➤ Bibliotecas ➤ Work** en la pestaña **Explorador de servidores de WPS**.

Uso de PROC UPLOAD

`PROC UPLOAD` está dirigido a transferir bibliotecas, conjuntos de datos o archivos desde la plataforma local, utilizando respectivamente las opciones `INLIB`, `DATA` y `INFILE` a un host remoto, utilizando las opciones `OUTLIB`, `OUT` y `OUTFILE`. `PROC UPLOAD` se debe colocar dentro de un bloque de código `RSUBMIT... ENDRSUBMIT`.

Los siguientes son algunos ejemplos de sintaxis:

```
/* transfer a library */
PROC UPLOAD INLIB=locallib OUTLIB=remotelib; RUN;
```

```
/* transfer a single dataset */
PROC UPLOAD DATA=locallib.dataset OUT=remotelib.dataset; RUN;
```

```
/*transfer a file */
PROC UPLOAD INFILE=”local_host_file_path” OUTFILE=”remote_host_file_path”; RUN;
```

Nota:

Para la sintaxis completa de este procedimiento, consulte la sección **Referencia**.

Ejemplo del uso de PROC UPLOAD

Esta sección contiene un ejemplo del uso de `PROC UPLOAD`, donde el conjunto de datos que se ha descargado a través del ejemplo `PROC DOWNLOAD` se carga y guarda en un segundo host remoto.

1. Copie y pegue el siguiente código:

```

/*****************************************************************
Sign on to the remote host to upload the dataset
*****************************************************************/
signon <server2name> ssh
user='<username>'
password='<password>'
```

```
launchcmd='<path to WPS executable>';

/*************************************************************
Create a library for the dataset on the host
*************************************************************/
rsubmit;
libname rlib "/home/<username>/datasets";

/*************************************************************
Upload the dataset to the host and output its contents to the library
*************************************************************/
proc upload
data=WORK.communicatedemo
out=rlib.communicatedemo;
run;

/*************************************************************
Complete the remote program execution
*************************************************************/
endrsubmit;

/*************************************************************
Sign off the remote host and close the connection
*************************************************************/
signoff;
```

Asegúrese de sustituir `<servername>` por el nombre del servidor remoto, reemplace `<username>` y `<password>` según corresponda y configure `launchcmd` para que apunte a la ubicación del archivo ejecutable de WPS en el servidor remoto, por ejemplo `/home/installs/wps-3.2/bin/wps -dmr`.

Importante:

Una vez que se encuentra la instrucción `signoff`, todos los conjuntos de datos en la ubicación `WORK` del host remoto se eliminarán, porque `WORK` es una ubicación temporal. Si no desea que esto suceda, debe generar el conjunto de datos a una ubicación permanente, por ejemplo, utilizando una biblioteca tal como `rlib` en el código anterior.

2. Ejecute el programa y examine el registro de salida, un mensaje debajo de la invocación `PROC UPLOAD` explica la transferencia de datos:

```
NOTE: Dataset upload in progress from WORK.communicatedemo to
 WORK.communicatedemo
NOTE: 333 bytes were transferred to dataset WORK.communicatedemo at 333000 bytes/
sec
NOTE: Dataset "WORK.communicatedemo" has 5 observation(s) and 2 variable(s)
```

El programa se ejecuta y carga el conjunto de datos.

Importante:

Sólo puede examinar el conjunto de datos si lo ha guardado en una ubicación permanente (es decir, no en `WORK`), por ejemplo ejecutando **WPS Workbench** en el servidor remoto y ejecutando una instrucción de apertura de la biblioteca, tal como `LIBNAME rlib "/home/<username>/datasets"`. A continuación, aparecerá en la pestaña **Explorador de servidores de WPS** en **Bibliotecas**.

Ejecución de subprogramas en paralelo

WPS Communicate asincrónico

WPS Communicate puede ejecutar subprogramas de forma asincrónica, es decir, un subprograma no tiene que esperar a la finalización de otro antes de ejecutarse. Dependiendo de las cargas de trabajo relativas de los subprogramas separados, esto puede conducir a mejoras significativas de rendimiento: la suma de las duraciones de los subprogramas hasta la duración del subprograma más largo reduce la espera.

Si un subprograma remoto se ejecuta de forma sincrónica o asincrónica, es controlado por la opción `WAIT` de su instrucción de invocación `RSUBMIT` y una correspondiente instrucción `WAITFOR` que se utiliza para hacer que WPS espere la finalización de uno o más subprogramas remotos.

En el siguiente fragmento de código, los subprogramas se ejecutan en paralelo en el *host1* y *host2*. Cuando hayan terminado, el procesamiento continúa localmente. Dicho procesamiento podría, por ejemplo, ejecutar una combinación de los resultados de los dos subprogramas anteriores.

```
%let remote-id1 = host1
%let remote-id2 = host2
/*******************************************************************
Sign on to the remote machines
*******************************************************************/
signon <remote-id1> ssh
user='<user1>'
password='<password1>'
launchcmd='<wps-install-location>/bin/wps -dmr>';

signon <remote-id2> ssh
user='<user2>'
password='<password2>'
launchcmd='<wps-install-location/bin/wps -dmr>';

/*******************************************************************
Execute sub-programs
*******************************************************************/
rsubmit <remote-id1> wait=no;
/****
Run code on <remote-id1>
****/
endrsubmit;

rsubmit <remote-id2> wait=no;
/****
Run code on <remote-id2>
****/
endrsubmit;

/*******************************************************************
Wait for all remote processing to complete
*******************************************************************/
WAITFOR _ALL_ <remote-id1> <remote-id2>;

/*******************************************************************
Release connections to remote machines
```

```
*****************************************************************/
signoff <remote-id1>;
signoff <remote-id2>;

/*****************************************************************
Perform final local processing
*****************************************************************/
data _null_;
/****
Local data step processing
****/
run;
```

La opción `WAIT=NO` de ambas instrucciones `RSUBMIT` informa WPS que los subprogramas debe ejecutarse de forma asincrónica.

Nota:

La instrucción `WAITFOR _ALL_` hace que el programa principal suspenda la ejecución hasta que el procesamiento se complete en **todo** los `remote-ids` del servidor o hasta que el intervalo `TIMEOUT`, si se específica, se haya expirado. Si utiliza `WAITFOR _ANY_ <remote-id1> <remote-id2>`, o simplemente `WAITFOR <remote-id1> <remote-id2>`, en lugar de `WAITFOR _ALL_`, la ejecución del programa principal sólo suspenderá la ejecución hasta que se complete el procesamiento en uno de **cualquier** `remote-id` del servidor (o hasta que el intervalo `TIMEOUT`, si se especifica, se expire). El valor predeterminado es `_ANY_` en lugar de `_ALL_` si no se proporciona ningún argumento entre `WAITFOR` y los `remote-ids` del servidor.

Ejecución de subprogramas locales en paralelo

WPS Communicate ofrece una forma adicional de ejecutar en paralelo el procesamiento local al permitirle crear múltiples conexiones a su plataforma local como si fueran conexiones con hosts remotos. En lugar de proporcionar instrucciones de inicio de sesión completas que requieren autenticación, como se ha descrito anteriormente, basta utilizar:

```
signon local1 launchcmd="<local-wps-install-directory>\bin\wps.exe -dmr";
signon local2 launchcmd="<local-wps-install-directory>\bin\wps.exe -dmr";
```

o simplemente

```
signon local1 launchcmd="!sascmd -dmr"
signon local2 launchcmd="!sascmd -dmr"
```

Los argumentos `local1` y `local2` se convierten en alias para las conexiones a la plataforma local que se pueden utilizar exactamente de la misma manera que los nombres de host de los servidores conectados de forma remota. Aunque depende de los patrones de uso de recursos de los varios subprogramas, tal técnica puede conducir a mejoras en el rendimiento aunque no haya ningún aumento neto de la CPU disponible o el ancho de banda de E/S.

Transferencia de las variables de macro

Las variables de macro se pueden pasar entre la plataforma local y un host remoto, utilizando las instrucciones del procesador de macros `%SYSLPUT` y `%SYSRPUT`.

La instrucción `%SYSLPUT` crea una variable de macro en un host remoto con el que ha establecido una sesión de **WPS Communicate**. La llamada de macro debe colocarse después de la instrucción `SIGNON`, pero antes de la instrucción `RSUBMIT`.

La instrucción `%SYSRPUT` recupera una variable de macro de un host remoto al que existe una sesión **WPS Communicate** establecida, creando una variable de macro local idéntica. La llamada de macro se puede sólo colocar dentro de un bloque de código `RSUBMIT... ENDRSUBMIT` porque se está ejecutando en el host remoto y devuelve las variables a la plataforma local.

Los siguientes son algunos ejemplos de fragmentos de código que utilizan estas llamadas:

```
signon <servername> ssh
user='<username>'
password='<password>'
launchcmd='<path to WPS executable>';
/*****************************************************************
Send over a macro definition from the local to the remote platform
*****************************************************************/
%SYSLPUT LOCALOS=&SYSSCPL.;

/*****************************************************************
SUBMIT the following code to UNIX, between the RSUBMIT/ENDRSUBMIT block
*****************************************************************/
rsubmit;

%put "EXECUTING ON REMOTE OS &SYSSCPL. FROM LOCAL OS &LOCALOS.";
%put &SYSHOSTNAME;

%SYSRPUT REMOTEOS=&SYSSCPL.;

endrsubmit;

%put "EXECUTING ON LOCAL OS &SYSSCPL. FROM REMOTE OS &REMOTEOS.";

/*****************************************************************
SIGNOFF the UNIX platform, preventing further RSUBMITs
*****************************************************************/
signoff;

%put &SYSHOSTNAME;
```

Nota:

Para obtener la sintaxis completa de estas instrucciones de procesador de macros, consulte la sección **Referencia**.

Referencia

Las definiciones de los diagramas de sintaxis de ferrocarril son anotaciones que ayudan a explicar la sintaxis de los lenguajes de programación, y se usan en esta guía para describir la sintaxis del lenguaje.

Cómo leer los diagramas de sintaxis del ferrocarril

Los diagramas del ferrocarril son una notación de la sintaxis gráfica que acompaña estructuras lingüísticas significativas, tales como procedimientos, instrucciones y demás.

La descripción de cada concepto lingüístico comienza con su diagrama de sintaxis.

Introducción de texto

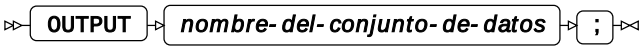
El texto que se debe introducir exactamente como está visualizado, se muestra en una fuente de máquina de escribir:

Este ejemplo describe un fragmento de sintaxis en el que la palabra clave `OUTPUT` termina con un carácter de punto y coma: `;`. La forma de diagrama de sintaxis es: 
.

Generalmente, las mayúsculas y minúsculas del texto no son significativas, pero en este aspecto, la convención es usar mayúsculas para palabras clave.

Elementos de los marcadores de posición

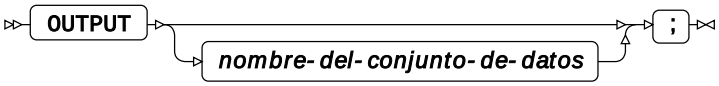
Los marcadores de posición que se deben sustituir con el texto pertinente y dependiente del contexto, se reproducen en una fuente minúscula y cursiva:



Aquí, la palabra clave `OUTPUT` se debe introducir literalmente, pero *nombre-del-conjunto-de-datos* se debe sustituir con algo apropiado al programa, en este caso, el nombre de un conjunto de datos al que agregar una observación.

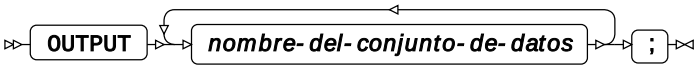
Opcionalidad

Cuando los elementos son opcionales, aparecen en una rama por debajo de la línea principal en diagramas del ferrocarril. La opcionalidad es representada por una ruta de acceso alternativa sin obstáculos a través del diagrama:



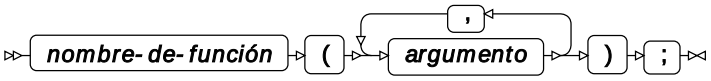
Repetición

En los diagramas de sintaxis, la repetición se representa con un bucle de retorno que opcionalmente especifica el separador que se debe colocar entre varias instancias.



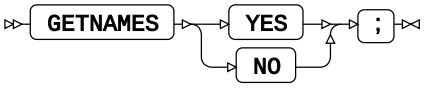
La palabra clave `OUTPUT` arriba se debe introducir literalmente y terminar con una o más repeticiones de *nombre-de-conjunto-de-datos*, en este caso, no se requiere ningún separador a excepción de un espacio.

El siguiente ejemplo muestra el uso de un separador.



Elecciones

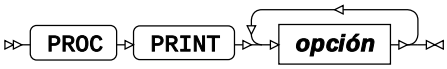
En los diagramas de sintaxis, la elección se muestra con varias ramas paralelas.



En el ejemplo de arriba, la palabra clave `GETNAMES` se debe introducir literalmente y, a continuación, la palabra clave `YES` o la palabra clave `NO`.

Fragmentos

Cuando la sintaxis es demasiado complicada para encajarse en una definición, podría dividirse en fragmentos:



opción

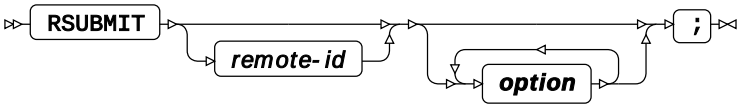
Arriba, la sintaxis completa se divide en fragmentos separados de diagramas de sintaxis. El primero indica que `PROC PRINT` debe terminarse con una o más instancias de una *opción*, cada una de las cuales debe adherir a la sintaxis proporcionada en el segundo diagrama.

Instrucciones globales

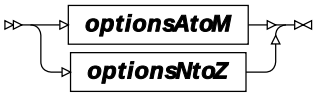
ENDRSUBMIT

Esta instrucción indica el final de un bloque de código que ha comenzado con una instrucción `RSUBMIT`.

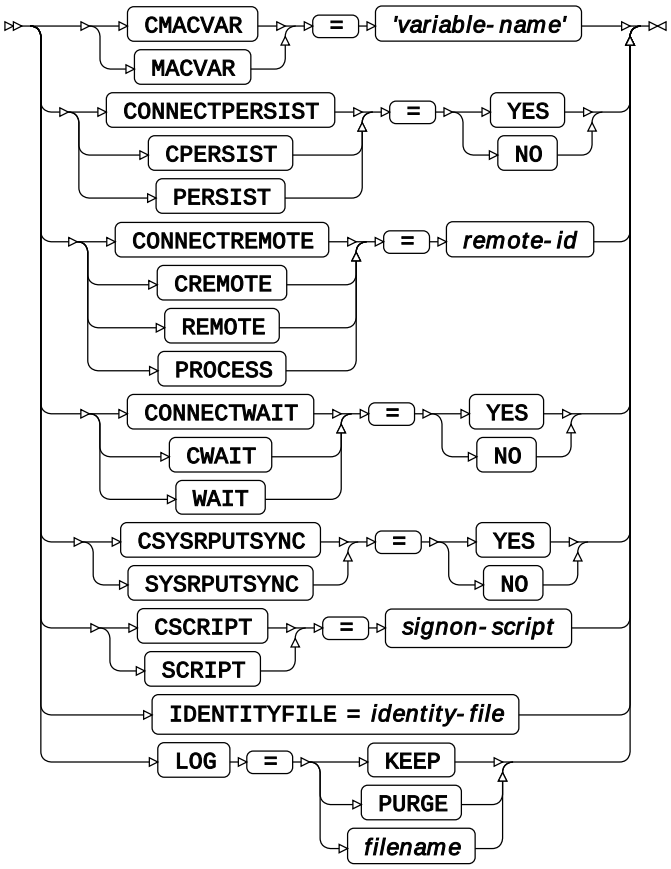
RSUBMIT



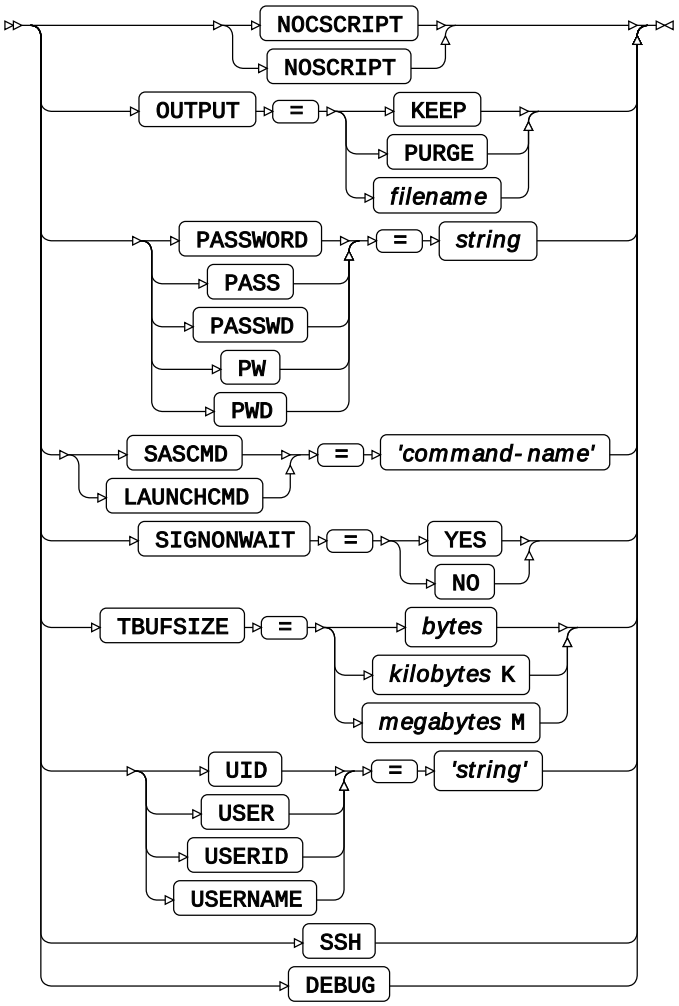
option



options A to M



options N to Z



Esta instrucción marca el principio de un bloque de código de programa que se enviará a un host (normalmente remoto) para la ejecución.

CMACVAR, MACVAR

Esta opción especifica una variable de macro cuyo valor está vinculado al estado de finalización del bloque `RSUBMIT` actual.

CONNECTPERSIST, CPERSIST, PERSIST

Esta opción significa si se produce o no un cierre de sesión automático después de `SIGNON` y `RSUBMIT`.

CONNECTREMOTE, CREMOTE, REMOTE, PROCESS

Esta opción identifica la máquina remota al que se establecerá una conexión, directamente o mediante el nombre de una variable de macro que contenga la dirección.

Nota:

Si se utiliza la opción CONNECTREMOTE con el nombre del host remoto proporcionado específicamente como una variable de macro, no debe colocarse ninguna Y comercial antes del nombre de la variable de macro. La sintaxis correcta se ilustra en el siguiente fragmento:

```
...
%LET HostName = RemoteHost;

options ssh_hostvalidation=none;
signon connectremote=HostName ssh /* Not &HostName */
user = <username>
password = <password>
launchcmd = '<location-of-wps-executable> -dmr';
...
```

CONNECTWAIT, CWAIT, WAIT

Esta opción determina si el bloque RSUBMIT se ejecutará en modo asincrónico o sincrónico, estableciéndolo en NO o YES respectivamente.

CSYSRPUTSYNC, SYSRPUTSYNC

Si se establece en YES, esta opción obliga a definir variables de macro cuando se ejecuta %SYSRPUT.

CSCCRIPT, SCRIPT

Esta opción identifica un script de inicio de sesión.

IDENTITYFILE

Esta opción especifica un archivo que contiene la información de autenticación, tal como las claves SSH.

NOSCRIPT, NOCSCRIPT

Esta opción indica que no se debe utilizar ningún script para iniciar la sesión.

LOG

Esta opción define si el registro del sistema debe mantenerse, purgarse o enviarse a un archivo específico.

OUTPUT

Esta opción define si la salida del subprograma debe mantenerse, purgarse o enviarse a un archivo específico.

PASSWORD, PASS, PASSWD, PW, PWD

Esta opción se utiliza para especificar una contraseña para la autorización remota.

SASCMD, LAUNCHCMD

Cuando está presente, esta opción se utiliza para especificar el comando necesario para iniciar WPS en la máquina remota.

SIGNONWAIT

Esta opción estipula que un `SIGNON` debe finalizar antes de permitir el procesamiento posterior.

TBUFSIZE

Esta opción especifica el tamaño del búfer de mensajes de WPS COMMUNICATE.

UID, USER, USERID, USERNAME

Cuando está presente, esta opción especifica el nombre de usuario.

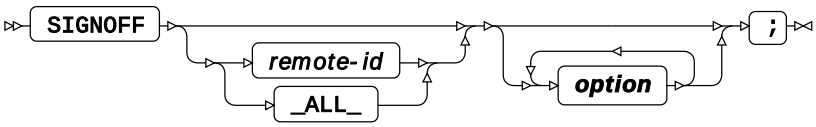
SSH

Esta opción especifica que la conexión utilizará el protocolo SSH cifrado.

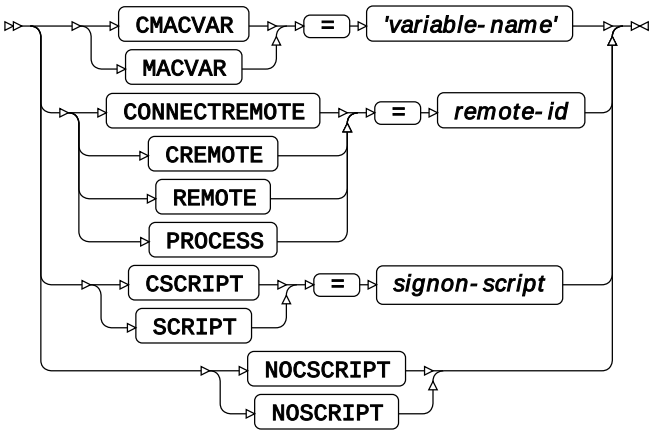
DEBUG

Esta opción especifica que los mensajes de depuración adicionales se escriben en el registro del sistema.

SIGNOFF



option



Esta instrucción cierra una conexión con un servidor remoto, después de la ejecución de un bloque de código ejecutado de forma remota.

CMACVAR, MACVAR

Esta opción especifica una variable de macro asociada con la sesión remota y cuyo valor está enlazado al estado de finalización de la instrucción `SIGNOFF` actual.

CONNECTREMOTE, CREMOTE, REMOTE, PROCESS

Esta opción asigna el nombre a la sesión remota desde la que desea cerrar la sesión.

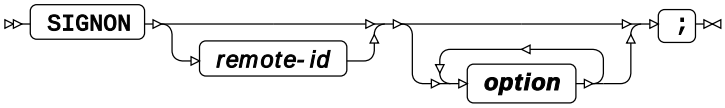
CSCRIPT, SCRIPT

Esta opción identifica un script que se ejecutará durante el cierre de sesión.

NOSCRIPT, NOCSCRIPT

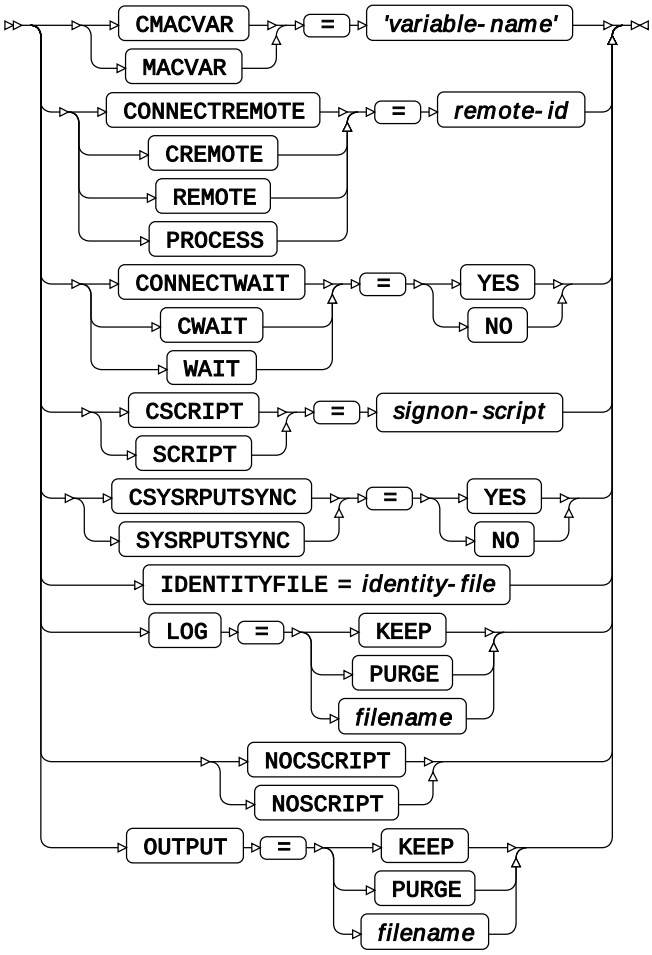
Esta opción indica que no debe implicar ningún script en el proceso de cierre de sesión.

SIGNON

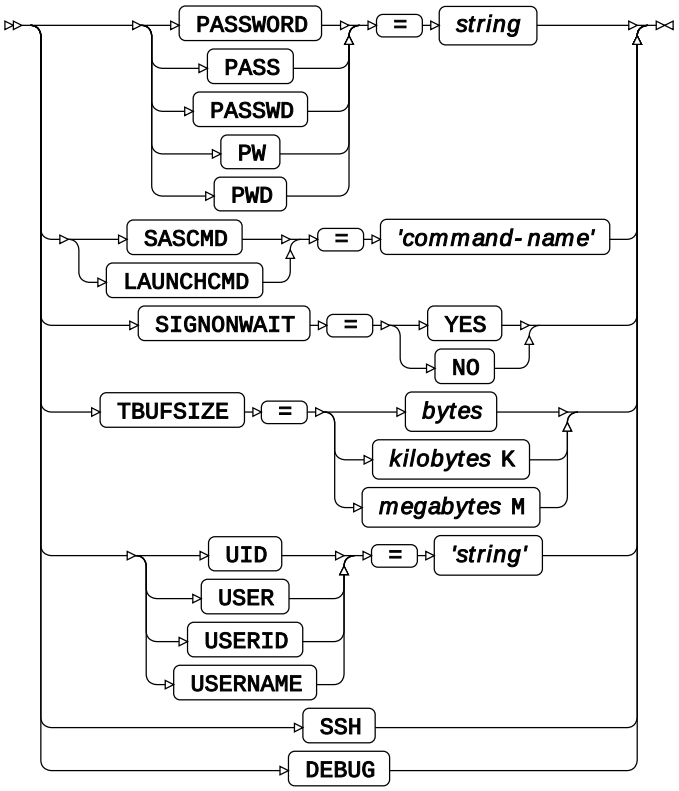


option

options A to O



options P to Z



Esta instrucción y sus opciones proporcionan la información necesaria para especificar dónde está ubicada la instalación de WPS remota, además de credenciales para conectarse e iniciar sesión en el servidor remoto, antes de invocar un bloque de código ejecutado de forma remota.

CMACVAR, MACVAR

Esta opción especifica una variable de macro asociada con la sesión remota y cuyo valor está enlazado al estado de finalización de la instrucción `SIGNON` actual.

CONNECTREMOTE, CREMOTE, REMOTE, PROCESS

Esta opción nombra la sesión remota.

Tenga en cuenta que si se utiliza la opción CONNECTREMOTE con el nombre del host remoto proporcionado específicamente como una variable de macro, (quizás contraintuitivamente) no debe colocarse ninguna Y comercial antes del nombre de la variable de macro. La sintaxis correcta se ilustra en el siguiente fragmento:

```
...
%LET HostName = RemoteHost;

options ssh_hostvalidation=none;
signon connectremote=HostName ssh /* Not &HostName */
user = <username>
password = <password>
launchcmd = '<location-of-wps-executable> -dmr';
...
```

CONNECTWAIT, CWAIT, WAIT

Esta opción determina si el bloque RSUBMIT se ejecutará en modo asincrónico o sincrónico, estableciéndolo en NO o YES respectivamente.

CSYSRPUTSYNC, SYSRPUTSYNC

Si se establece en YES, esta opción obliga a definir variables de macro cuando se ejecuta %SYSRPUT.

CSCCRIPT, SCRIPT

Esta opción identifica un script de inicio de sesión.

IDENTITYFILE

Esta opción especifica un archivo que contiene la información de autenticación, tal como las claves SSH.

NOSCRIPT, NOCSCRIPT

Esta opción indica que no se debe utilizar ningún script para iniciar sesión.

LOG

Esta opción define si el registro del sistema debe mantenerse, purgarse o enviarse a un archivo específico.

OUTPUT

Esta opción define si la salida del subprograma debe mantenerse, purgarse o enviarse a un archivo específico.

PASSWORD, PASS, PASSWD, PW, PWD

Esta opción se utiliza para especificar una contraseña para la autorización remota.

SASCMD, LAUNCHCMD

Cuando está presente, esta opción se utiliza para especificar el comando necesario para iniciar WPS en la máquina remota.

SIGNONWAIT

Esta opción estipula que un `SIGNON` debe finalizar antes de permitir el procesamiento posterior.

TBUFSIZE

Esta opción especifica el tamaño del búfer de mensajes de WPS COMMUNICATE.

UID, USER, USERID, USERNAME

Cuando está presente, esta opción especifica el nombre de usuario.

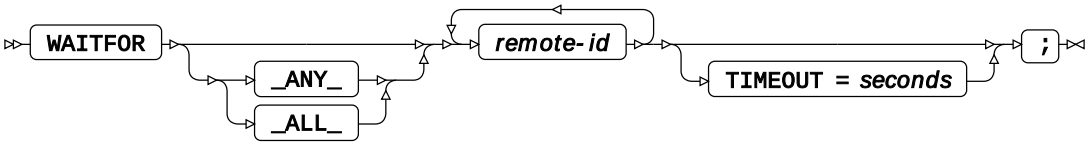
SSH

Esta opción especifica que la conexión utilizará el protocolo SSH cifrado.

DEBUG

Esta opción especifica que los mensajes de depuración adicionales se escriben en el registro del sistema.

WAITFOR



Dado que el diagrama anterior se aplica sólo a WPS Communicate, la instrucción `WAITFOR _ALL_` suspende la ejecución de la sesión actual hasta que se complete el procesamiento de **todos** los `remote-ids` del servidor o hasta que el intervalo `TIMEOUT`, si se especifica, haya expirado.

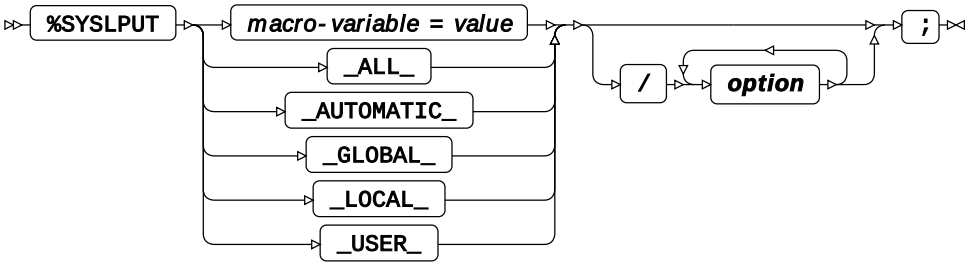
Si utiliza `WAITFOR _ANY_`, o simplemente `WAITFOR`, en lugar de `WAITFOR _ALL_`, la ejecución de la sesión sólo se suspenderá hasta que se complete el procesamiento en uno de los `remote-ids` del servidor (o hasta que el intervalo `TIMEOUT`, si se especifica, se expire).

Nota:

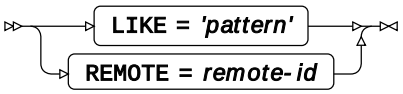
Como se implica arriba, el valor predeterminado es `_ANY_` en lugar de `_ALL_` si no se proporciona ningún argumento entre `WAITFOR` y los `remote-ids`.

Instrucciones del procesador de macros

%SYSLPUT

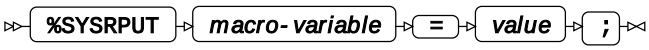


option



Esta instrucción crea una variable de macro en un host remoto con el que ha establecido una sesión de WPS Communicate. Debe colocarse fuera del correspondiente bloque `RSUBMIT`.

%SYSRPUT



Esta instrucción recupera una variable de macro de un host remoto al que hay una sesión de WPS Communicate establecida, creando una variable de macro local idéntica. Debe colocarse dentro del correspondiente bloque `RSUBMIT`.

Procedimientos

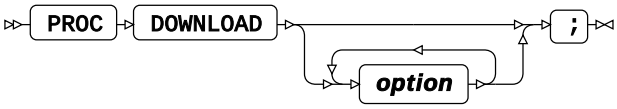
Procedimiento DOWNLOAD

Este procedimiento descarga uno o más archivos, bibliotecas o conjuntos de datos de un host remoto. Sólo se puede invocar desde el interior de un bloque `RSUBMIT`.

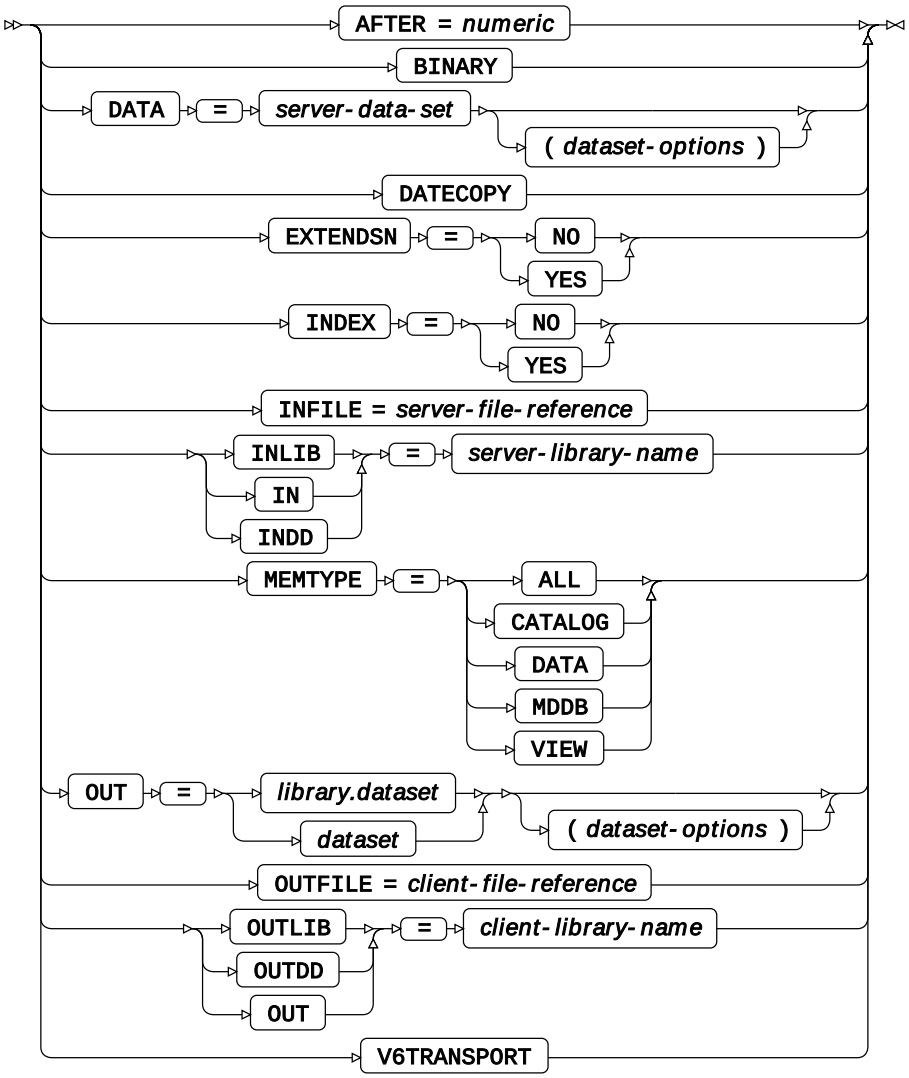
Instrucciones admitidas

- *PROC DOWNLOAD* (pág. 29)
- *EXCLUDE* (pág. 32)
- *SELECT* (pág. 32)
- *WHERE* (pág. 33)

PROC DOWNLOAD



option



AFTER

Especifica una fecha de modificación numérica, asegurándose de que sólo se descargan conjuntos de datos o bibliotecas modificados después de esta fecha. Esta opción no es válida para las descargas de archivos externos.

BINARY

Válida sólo al descargar archivos externos, esta opción especifica que la transferencia debe ser una copia binaria exacta.

DATA

Especifica el nombre de un conjunto de datos que se va a descargar.

DATECOPY

Cuando está presente, esta opción indica que la fecha y hora de creación de un conjunto de datos remoto debe conservarse cuando se descarga. Esta opción no es válida para las descargas de archivos externos.

EXTENDSN

Especifica si las variables numéricas cortas deben tener sus longitudes extendidas. Esta opción no es válida para descargas de archivos externos y podría considerarse si se están transfiriendo conjuntos de datos de un mainframe a un PC.

INDEX

Para conjuntos de datos remotos que tienen índices, esto indica si estos índices deben restablecerse en la máquina local después de la descarga. Esta opción no es válida para descargas de archivos externos.

INFILE

Especifica el nombre de un archivo externo remoto a descargar. Si esta opción está presente, también debe ser la opción `OUTFILE=`.

INLIB

Especifica el nombre de la biblioteca remota. Esta opción no es válida para las descargas de archivos externos.

OUT

Especifica el nombre del conjunto de datos local receptor. Esta opción no es válida para las descargas de archivos externos.

OUTFILE

Especifica el nombre del archivo local para recibir una descarga de archivos externos. Si esta opción está presente, también debe ser la opción `INFILE`.

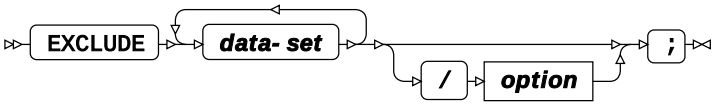
OUTLIB

Especifica el nombre de la biblioteca local en la que se descarga un conjunto de datos remoto. Esta opción no es válida para las descargas de archivos externos.

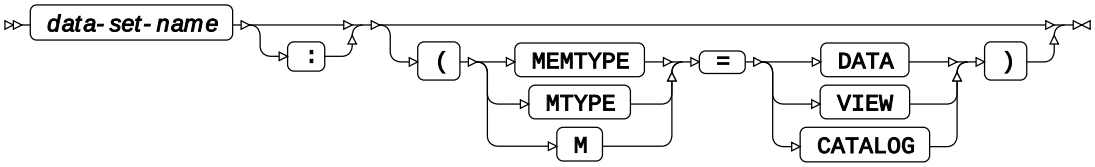
V6TRANSPORT

Esta es una opción de traducción al intercambiar los datos entre dos versiones diferentes.

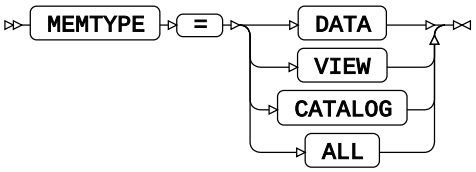
EXCLUDE



data-set



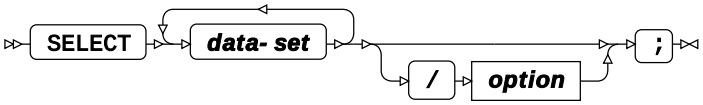
option



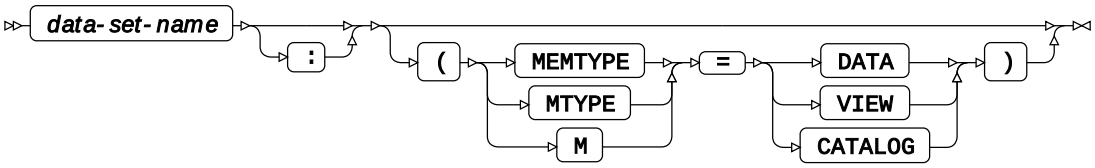
MEMTYPE

Esta opción especifica los tipos de miembro a descargar, vea el diagrama de sintaxis anterior. Esta opción no es válida para las descargas de archivos externos.

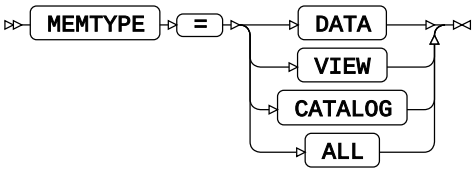
SELECT



data-set



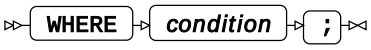
option



MEMTYPE

Vea la instrucción `EXCLUDE`.

WHERE



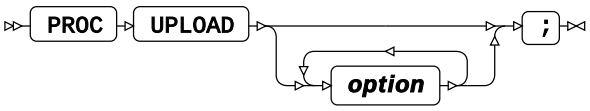
Procedimiento UPLOAD

Este procedimiento carga uno o más archivos, bibliotecas o conjuntos de datos a un host remoto. Sólo se puede invocar desde el interior de un bloque `RSUBMIT`.

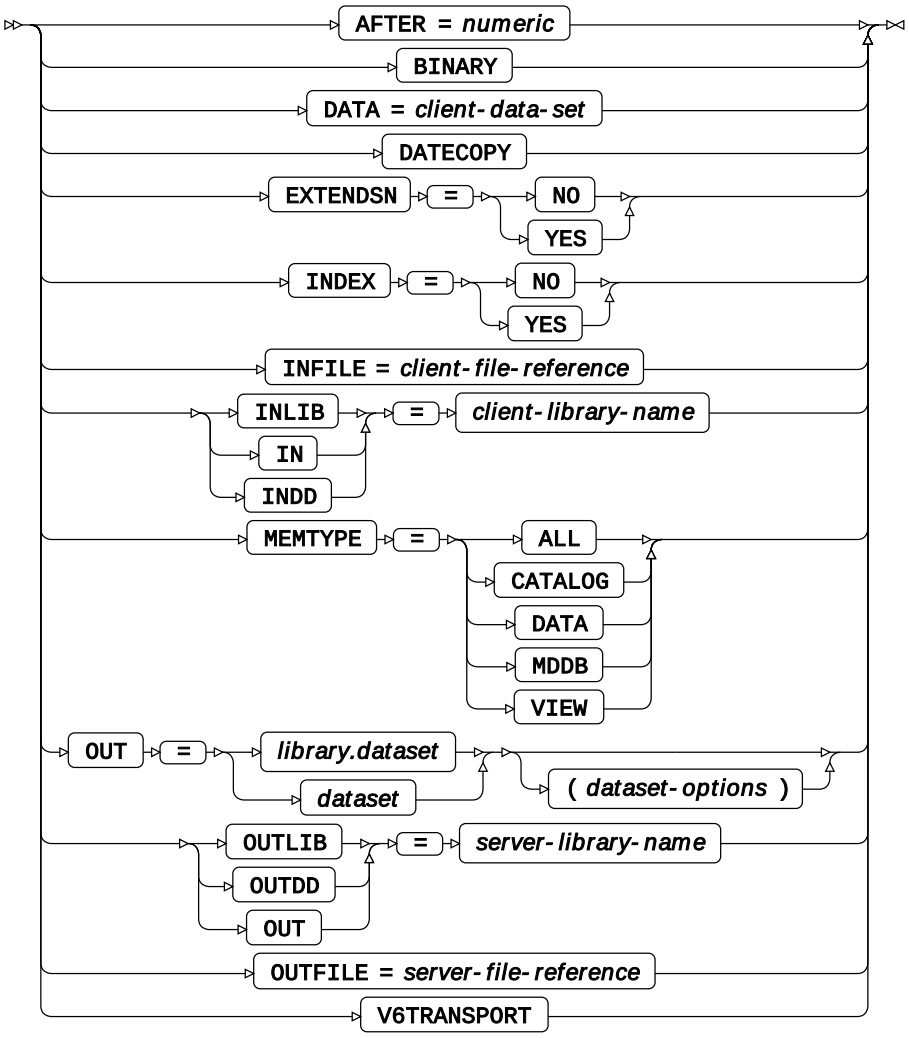
Instrucciones admitidas

- *PROC UPLOAD* (pág. 33)
- *EXCLUDE* (pág. 36)
- *SELECT* (pág. 36)
- *WHERE* (pág. 37)

PROC UPLOAD



option



AFTER

Especifica una fecha de modificación numérica, asegurándose de que solo se carguen conjuntos de datos o bibliotecas modificadas después de esta fecha. Esta opción no es válida para las cargas externas de archivos.

BINARY

Válida sólo al cargar archivos externos, esta opción especifica que la transferencia debe ser una copia binaria exacta.

DATA

Especifica el nombre de un conjunto de datos que se va a cargar.

DATECOPY

Cuando está presente, esta opción indica que la fecha y la hora de creación de un conjunto de datos local deben conservarse al cargarse. Esta opción no es válida para las cargas externas de archivos.

EXTENDSN

Especifica si las variables numéricas cortas deben tener sus longitudes extendidas. Esta opción no es válida para las cargas externas de archivos y podría considerarse si se transfieren conjuntos de datos a un mainframe desde un PC.

INDEX

Para conjuntos de datos locales que tienen índices, esto indica si estos índices deben restablecerse en la máquina remota después de la carga. Esta opción no es válida para las cargas de archivos externos.

INFILE

Especifica el nombre de un archivo externo local a cargar. Si esta opción está presente, también debe ser la opción `OUTFILE=`.

INLIB

Especifica el nombre de la biblioteca local. Esta opción no es válida para las cargas externas de archivos.

MEMTYPE

Esta opción especifica los tipos de miembro a cargar, vea el diagrama de sintaxis anterior. Esta opción no es válida para las cargas externas de archivos.

OUT

Especifica el nombre del conjunto de datos local receptor. Esta opción no es válida para las cargas externas de archivos.

OUTFILE

Especifica el nombre del archivo remoto para recibir una carga de archivos externos. Si esta opción está presente, también debe ser la opción `INFILE`.

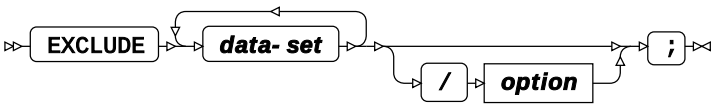
OUTLIB

Especifica el nombre de la biblioteca remota en la que se carga un conjunto de datos local. Esta opción no es válida para las cargas externas de archivos.

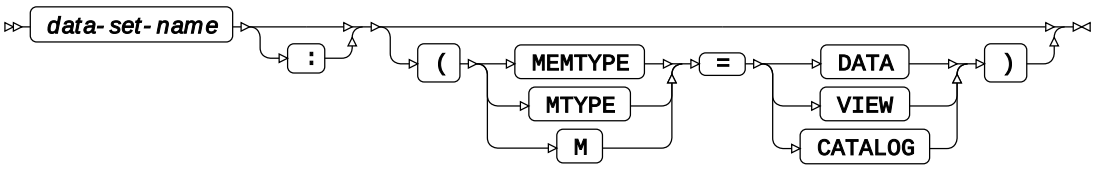
V6TRANSPORT

Esta es una opción de traducción al intercambiar los datos entre dos versiones diferentes.

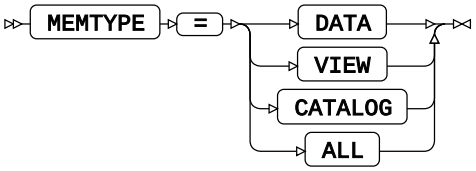
EXCLUDE



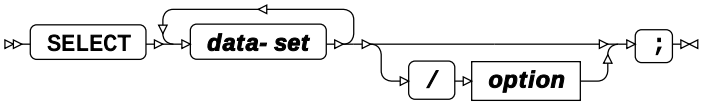
data-set



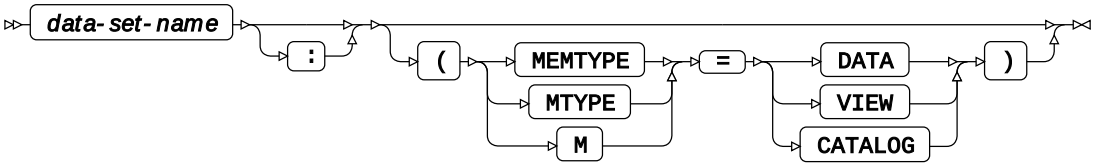
option



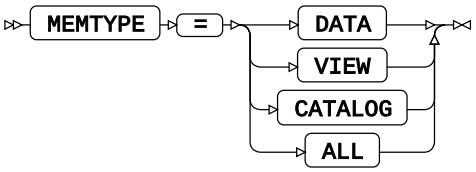
SELECT



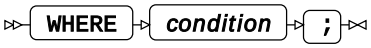
data-set



option

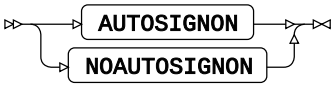


WHERE



Opciones del sistema

AUTOSIGNON



Válido en:	Instrucción `OPTIONS`, archivo de configuración y línea de comandos
Valor predeterminado:	NOAUTOSIGNON

Descripción

Cuando esta opción del sistema está activa, el envío remoto intentará iniciar la sesión automáticamente.

COMAMID

Válido en:	Instrucción `OPTIONS`, archivo de configuración y línea de comandos
Valor predeterminado:	TCP

Longitud máxima:	8

Descripción

Esta opción del sistema especifica el método de comunicación a utilizar para establecer comunicaciones remotas.

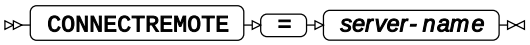
CONNECTPERSIST

Válido en:	Instrucción `OPTIONS`, archivo de configuración y línea de comandos
Valor predeterminado:	CONNECTPERSIST

Descripción

Cuando se establece, esta opción del sistema especifica que una conexión remota se mantendrá después de un bloque `RSUBMIT`. Esta opción del sistema es un alias de `CPERSIST`.

CONNECTREMOTE

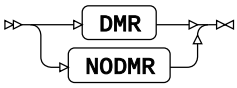


Válido en:	Instrucción `OPTIONS`, archivo de configuración y línea de comandos
Valor predeterminado:	*blank*
Longitud máxima:	1024

Descripción

Esta opción del sistema identifica un servidor remoto específico al que conectarse. Está en blanco (cadena vacía) de manera predeterminada.

DMR

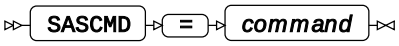


Válido en:	Sólo línea de comandos
Valor predeterminado:	NODMR

Descripción

Esta opción del sistema invoca una sesión de servidor WPS COMMUNICATE. Es inactivo de manera predeterminada y sólo se puede efectuar a través de la línea de comandos.

SASCMD

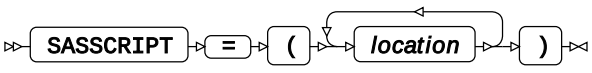


Válido en:	Instrucción `OPTIONS`, archivo de configuración y línea de comandos
Valor predeterminado:	""
Longitud máxima:	32767

Descripción

Esta opción del sistema especifica el comando que WPS COMMUNICATE utilizará para iniciar otra sesión de WPS local. Está en blanco (cadena vacía) de manera predeterminada.

SASSCRIPT



Válido en:	Instrucción `OPTIONS`, archivo de configuración y línea de comandos
Valor predeterminado:	""
Longitud máxima:	1024

Descripción

Esta opción del sistema especifica la ubicación de los scripts de inicio de sesión de WPS COMMUNICATE. Está en blanco (cadena vacía) de manera predeterminada.

Guía de administración del sistema

Esta parte de la guía está destinada a los administradores de sistemas que son responsables de la autenticación del servidor relativa a **WPS Communicate** y **WPS Link** y de la generación y implementación de las claves públicas y privadas requeridas.

Nota:

WPS Communicate y **WPS Link** son funciones separadas que pueden ejecutarse independientemente. Es decir, no se necesita uno para ejecutar al otro. Utilice **WPS Communicate** para ejecutar el código selectivo en hosts de servidor remoto o un mainframe de z/OS, y **WPS Link** para ejecutar programas enteros, a través de la GUI de Workbench, sólo en hosts de servidores remotos. Las dos funciones se describen juntas aquí, ya que ambas requieren medios de autenticación remota.

El siguiente es un resumen de los métodos de autenticación que se aplican tanto a **WPS Communicate** como a **WPS Link**.

Método de autenticación	WPS Communicate	WPS Link
Contraseña	Sí	Sí
Clave pública con la frase de contraseña y el agente de llavero	Sí	Sí
Clave pública con la frase de contraseña y ningún agente de llavero	No	Sí
Kerberos	Sí	Sí
Telnet en z/OS	Sí	No

Autenticación en z/OS a través de Telnet

WPS Communicate puede utilizar un inicio de sesión Telnet a un host z/OS remoto para iniciar un servidor de WPS a través de un `CLIST` proporcionado, denominado `TSOWPS`. No se necesita ninguna configuración de USS (UNX System Services), y la parte USS de WPS no se utiliza.

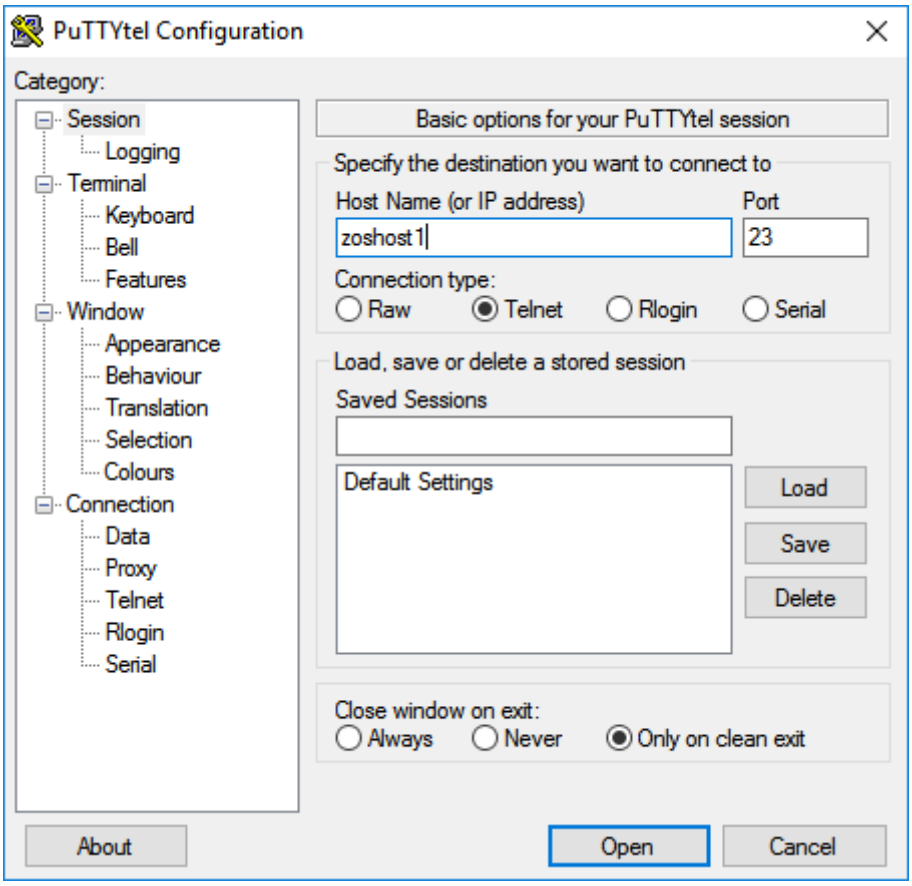
1. Primero, asegúrese de que el `TSOWPS CLIST` esté operativo para su instalación, ejecutando un normal inicio de sesión TN3270 con un emulador de terminal 3270 en su host z/OS e invocando el `TSOWPS CLIST` proporcionado. Normalmente, puede utilizar la opción 6 de ISPF (Interactive System Productivity Facility) y escriba `TSOWPS` en el aviso.

   Si recibe un mensaje similar al siguiente, es posible que necesite realizar cambios en la instalación de `TSOWPS CLIST` o WPS:

   ```
   WPS CANNOT BE INVOKED AS WPSPFX HAS NOT BEEN DEFINED
   ```

   Este mensaje le informa que el prefijo del conjunto de datos de instalación no está resuelto y que WPS no puede localizar sus varios componentes.

2. Capture los avisos de inicio de sesión de Telnet ejecutando un inicio de sesión manual en el host z/OS con un cliente Telnet, tal como el cliente PuTTYtel. Inicie este cliente e introduzca el nombre del host en el campo de entrada **Host Name (o IP address)**. Asegúrese de que el botón de radio **Telnet** está seleccionado:



   Los mensajes subsiguientes dependerán de cómo se haya configurado el host z/OS. El propósito de este paso es doble:

   - El primero es verificar que puede ejecutar una conexión Telnet al host.
   - La segunda es capturar los avisos y las respuestas necesarias.

El registro siguiente captura una secuencia típica de avisos al ejecutar un inicio de sesión de Telnet en un host z/OS. Nuestro objetivo es entender esta secuencia y luego usarla para ayudar a escribir un script que automatice el proceso.

Nota:

Las respuestas introducidas por el usuario no se han repetido de esta salida por el motivo de que la secuencia exacta de entradas y respuestas depende a menudo de la instalación. De ello se deduce que el script de inicio de sesión manualmente codificado puede necesitar ajustes en consecuencia.

```
IKJ56700A ENTER USERID -
IKJ56714A ENTER CURRENT PASSWORD FOR XXXX-
ICH70001I XXXX LAST ACCESS AT 13:59:30 ON THURSDAY, JANUARY 15, 2015
IKJ56496I DEFAULT ACCOUNT NUMBERS COULD NOT BE OBTAINED - ENTER ACCOUNT NUMBER
IKJ56481I THE PROCEDURE NAME DBSPROCA IS A DEFAULT NAME - YOU MAY CHANGE IT
IKJ56455I XXXX LOGON IN PROGRESS AT 14:05:08 ON JANUARY 15, 2015
IKJ56951I NO BROADCAST MESSAGES
*************************************************************
* APPLICATION DEVELOPER'S CONTROLLED DISTRIBUTION (ADCD) *
*************************************************************
* *
* USER.CLIST(ISPFCL) PRODUCES THIS MESSAGE *
* USER.* DATASETS CONTAIN SYSTEM CUSTOMIZATION BY WP *
* ADCD.* DATASETS CONTAIN SYSTEM CUSTOMIZATION *
* SMP/E DATASETS CAN BE LOCATED FROM 3.4 WITH DSNAME **.CSI *
* *
*************************************************************
READY
```

3. Escriba el script de inicio de sesión de Telnet.

Después de haber ejecutado un inicio de sesión manual, es necesario escribir un script para automatizar el proceso, informado por la secuencia de desafíos y respuestas observadas durante el inicio de sesión manual. En el siguiente comentario, cada línea de script se describe con una breve narración.

Nota:

Se incluye un script de muestra completo y 'limpio' (para el corte/pegado y la modificación potenciales) en *Ejemplo de script de autenticación Telnet* (pág. 46).

```
TRACE ON;
```

La instrucción `TRACE ON` envía las instrucciones al registro tal como se han ejecutado por WPS. Esto es útil para la depuración, pero probablemente se debe desactivar para la producción.

```
ECHO ON;
```

La instrucción `ECHO ON` hace que todas las respuestas recibidas desde el servidor de Telnet se repitan en el registro, de nuevo, esto es útil para los propósitos de depuración.

```
LOG "NOTE: Signon script entered.";
```

Esta instrucción imprime un mensaje en el registro para indicar que se está procesando el script. Las instrucciones `LOG` pueden utilizarse generosamente en el script de inicio de sesión para demostrar el progreso.

```
IF signoff THEN GOTO signoff;
```

Esta línea es una parte estándar de la mayoría de los scripts de inicio de sesión. Una variable especial `signoff` se establece si el script se ejecuta como consecuencia de la instrucción `SIGNOFF` en lugar de la instrucción `SIGNON`. Esto permite que un script sobrelleve ambas situaciones. Aquí, el control se ramifica a la etiqueta `signoff` si se detecta un cierre de sesión.

Nota:

No hay ningún significado en el nombre de la etiqueta; es simplemente convencional llamarlo `signoff`.

```
WAITFOR "ENTER USERID -", 5 seconds : fail;
TYPE "&USER" ENTER;
```

Aquí, se invoca una instrucción `WAITFOR`, para esperar a que la línea dada se reciba desde el servidor de Telnet. La instrucción hace que el script espere hasta que se reciba una línea que contenga el texto dado; en este caso, `ENTER USERID -` en algún lugar de su contenido. Si no recibe el mensaje indicado en 5 segundos, se ramificará a la siguiente etiqueta `fail`. Si se recibe la respuesta, el procesamiento continúa en la línea siguiente que simula el usuario que escribe. Aquí, la respuesta es el nombre de usuario, que se proporciona al script de inicio de sesión mediante una variable de macro `&USER`.

```
WAITFOR "ENTER CURRENT PASSWORD", 5 seconds : fail;
TYPE "&PASSWORD" ENTER;
```

El siguiente mensaje esperado es la solicitud de contraseña, y de nuevo se proporciona a través de una variable de macro.

Nota:

No es necesario capturar y coincidir con el aviso completo; el comando `WAITFOR` simplemente busca una línea que contenga el texto dado en algún lugar dentro de él.

```
WAITFOR "ENTER ACCOUNT NUMBER", 5 seconds : fail;
TYPE "WPS" ENTER;
WAITFOR "YOU MAY CHANGE IT", 5 seconds : fail;
TYPE ENTER;
```

Luego, aparece un mensaje pidiendo un `ACCOUNT NUMBER`, al cual la respuesta dependerá de la instalación. Aquí, es `WPS`. A continuación, puede aparecer un aviso acerca de un nombre de procedimiento, al que una respuesta `ENTRAR` vacía podría ser una respuesta satisfactoria. Dependiendo de la configuración de z/OS, una o más respuestas adicionales se pueden también considerar y proporcionar usando las mismas técnicas; consulte los intercambios que ocurrieron durante el inicio de sesión manual inicial para determinar si hay que tener en cuenta más conversaciones de este tipo.

```
WAITFOR "READY", 5 seconds : fail;
LOG 'NOTE: Logged onto z/OS. Starting remote WPS now...';
```

El siguiente aviso que esperar es el aviso READY, que indica que la autenticación está completa y que la sesión del TSO está lista para recibir comandos. Esto ilustra una función de la instrucción WAITFOR; se reciben varias líneas antes de que llegue el aviso READY, pero éstas se analizan y se omiten. No es necesario incluir una instrucción WAITFOR para cada línea de salida que genera el proceso de inicio de sesión.

```
TYPE "altlib activate application " lf;
WAITFOR 'ENTER Application library', 5 seconds :fail;
TYPE "CLIST" lf;
WAITFOR "ENTER a single dataset", 5 seconds :fail;
TYPE "'WPS.V310.B31754.CLIST'" lf;
WAITFOR "READY", 5 seconds : fail;
type "TSOWPS OPTIONS('DMR WPSCOMPROTOCOL=WPS') TRACE"  enter;
```

Aquí, se invocan los comandos necesarios para iniciar WPS. La biblioteca de WPS CLIST se agrega temporalmente a la ruta de búsqueda CLIST mediante el comando ALTLIB; el nombre de biblioteca específico depende de la versión de WPS y del número de compilación. A continuación, la TSOWPS CLI se invoca explícitamente. Esto ya se puede mover a una biblioteca de CLIST del usuario, en cuyo caso el comando de ALTLIB no sería necesario. Para invocar WPS como un servidor para usar con **WPS Communicate**, se requiere la opción DMR y mientras WPSCOMPROTOCOL toma el valor predeterminado de WPS en la mayoría de las instalaciones, lo mejor es ser explícitos. Finalmente, la configuración de TRACE es opcional y puede omitirse si la salida de CLIST es demasiado detallada.

```
WAITFOR "SESSION ESTABLISHED", 5 seconds : fail;
LOG 'NOTE: WPS Communicate conversation established.';
STOP;
```

Después de iniciar el proceso de WPS, el script espera la línea de salida indicando que el servidor de WPS está en ejecución y está esperando la conexión secundaria. Esta línea siempre contiene la cadena SESSION ESTABLISHED y la ruta de acceso correcta a través del script de inicio de sesión siempre debe terminar con una instrucción WAITFOR. Una vez recibida esta línea, la instrucción STOP termina el procesamiento del script de inicio de sesión, devolviendo el control a WPS.

```
signoff:
WAITFOR 'READY', 5 seconds : fail;
TYPE 'LOGOFF' ENTER;
WAITFOR "LOGGED OFF", 5 seconds : fail2;
LOG 'NOTE: WPS Communicate conversation terminated.';
STOP;
```

Esta sección contiene las líneas que se ejecutan cuando se produce un SIGNOFF. El proceso de WPS se habrá señalado para terminar, pero es necesario esperar una pausa para que termine y para que se reciba el mensaje TSO READY. Una vez que se haya recibido, se simula un comando LOGOFF escrito, esperando el mensaje de confirmación antes de terminar el script con una instrucción STOP.

```
fail:
LOG "ERROR: Expected prompt not received";
TYPE "LOGOFF" enter;

fail2:
ABORT;
```

Las etiquetas `fail` y `fail2` son etiquetas que anuncian el código que se activa cuando no se encuentran las respuestas esperadas, incluso al intentar cerrar la sesión. Señalan las condiciones de error que, si se alcanzan, necesitan de una mayor investigación.

4. Pruebe el script de inicio de sesión.

   Después de escribir un script de inicio de sesión básico y guardarlo en una ubicación conocida, debe probarlo estableciendo una conexión con él. Esto requiere un simple programa de WPS tal como:

```
filename rlink '<path to signon script>';

%let HOST=zoshost1;
%let USER=XXXX;
%let PASSWORD=XXXX;

signon HOST;
rsubmit;
%PUT &SYSHOSTNAME;
endrsubmit;
signoff;
```

   Deberá sustituir la instrucción `filename rlink` con la ruta completa del script de inicio de sesión y proporcionar el nombre de host, junto con un nombre de usuario y una contraseña adecuados para iniciar la sesión en el host z/OS. Estos deben ser los mismos que se utilizaron anteriormente durante el inicio de sesión manual.

   Si todo va bien, se debe crear un registro de WPS sin errores. Habrán algunas salida de depuración adicional debido a las instrucciones `ECHO` y `TRACE` en el script de inicio de sesión, pero si el inicio de sesión se ha realizado correctamente, se pueden comentar estas instrucciones, poniendo `/* */` alrededor de ellos, o eliminado, para hacer la salida menos detallada. La variable de la macro `&SYSHOSTNAME` se resolverá con el host remoto y su valor se escribirá en el registro de WPS local, lo que demuestra que la conexión y el inicio de sesión se han finalizado correctamente.

Ejemplo de script de autenticación Telnet

```
TRACE ON;

ECHO ON;

LOG "NOTE: Signon script entered.";

IF signoff THEN GOTO signoff;

WAITFOR "ENTER USERID -", 5 seconds : fail;
TYPE "&USER" ENTER;

WAITFOR "ENTER CURRENT PASSWORD", 5 seconds : fail;
TYPE "&PASSWORD" ENTER;

WAITFOR "ENTER ACCOUNT NUMBER", 5 seconds : fail;
TYPE "WPS" ENTER;
```

```
WAITFOR "YOU MAY CHANGE IT", 5 seconds : fail;
TYPE ENTER;

WAITFOR "READY", 5 seconds : fail;
LOG 'NOTE: Logged onto z/OS... Starting remote WPS now.';

TYPE "altlib activate application " lf;
WAITFOR 'ENTER Application library', 5 seconds :fail;
TYPE "CLIST" lf;
WAITFOR "ENTER a single dataset", 5 seconds :fail;
TYPE "'WPS.V310.B31754.CLIST'" lf;
WAITFOR "READY", 5 seconds : fail;
type "TSOWPS OPTIONS('DMR WPSCOMPROTOCOL=WPS') TRACE "  enter;

WAITFOR "SESSION ESTABLISHED", 5 seconds : fail;
LOG 'NOTE: WPS Communicate conversation established.';
STOP;

signoff:
WAITFOR 'READY', 5 seconds : fail;
TYPE 'LOGOFF' ENTER;
WAITFOR "LOGGED OFF", 5 seconds : fail2;
LOG 'NOTE: WPS Communicate conversation terminated.';
STOP;

fail:
LOG "ERROR: Expected prompt not received";
TYPE "LOGOFF" enter;

fail2:
ABORT;
```

SSH (Secure Shell) desde un cliente Windows

Esta sección abarca el uso de SSH con **WPS Communicate** y **WPS Link** para crear y mantener las conexiones entre los equipos servidor y cliente.

Nota:

Las diferencias de uso entre **WPS Communicate** y **WPS Link** se destacan cuando es apropiado.

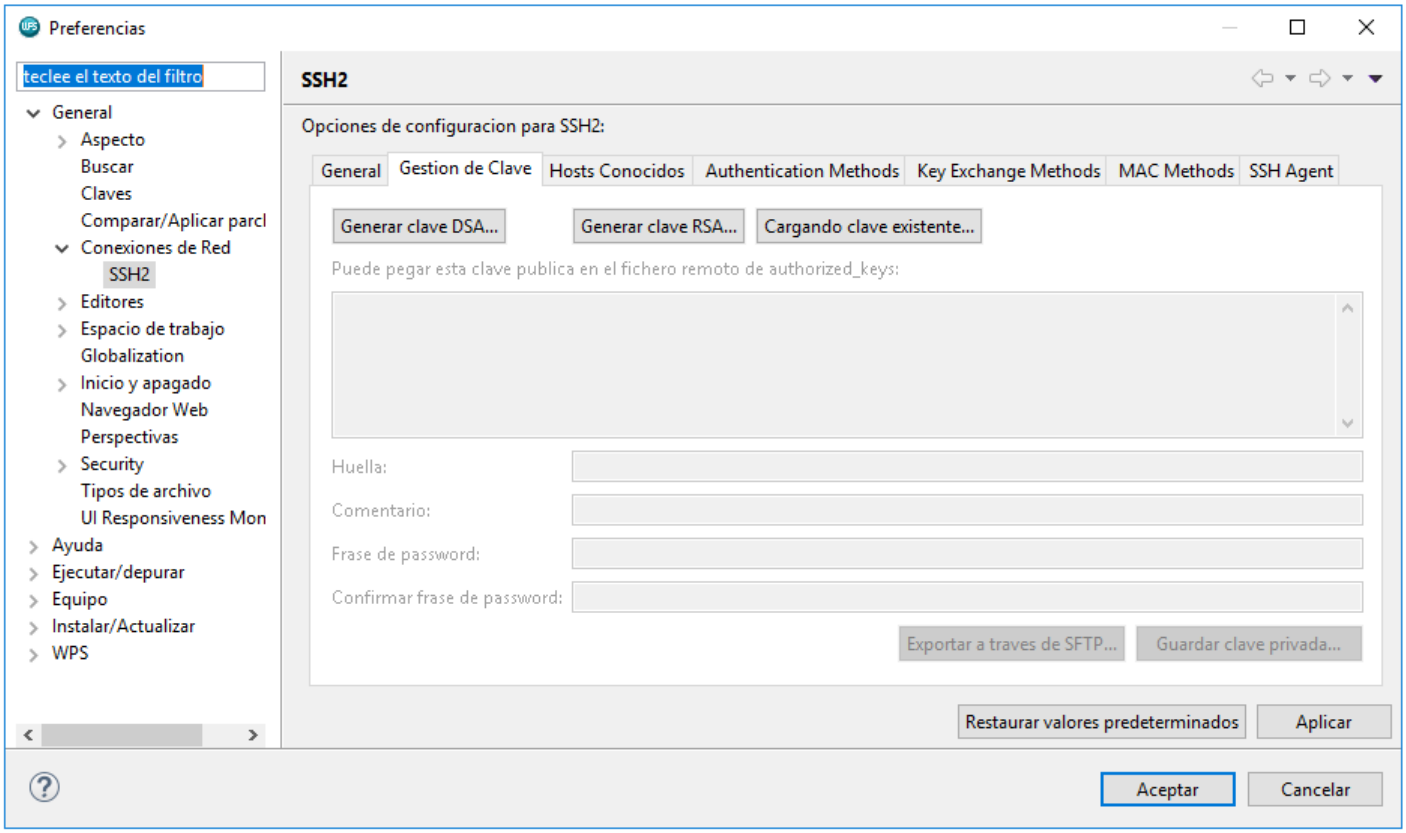
Antes de acceder a un host remoto mediante **WPS Communicate** o **WPS Link**, es importante asegurarse de que puede acceder al host remoto manualmente a través de un cliente SSH externo, tal como PuTTY. Esto demuestra que puede al menos conectarse a la máquina mediante el protocolo SSH y que su ID de usuario y contraseña son válidos.

Nota:

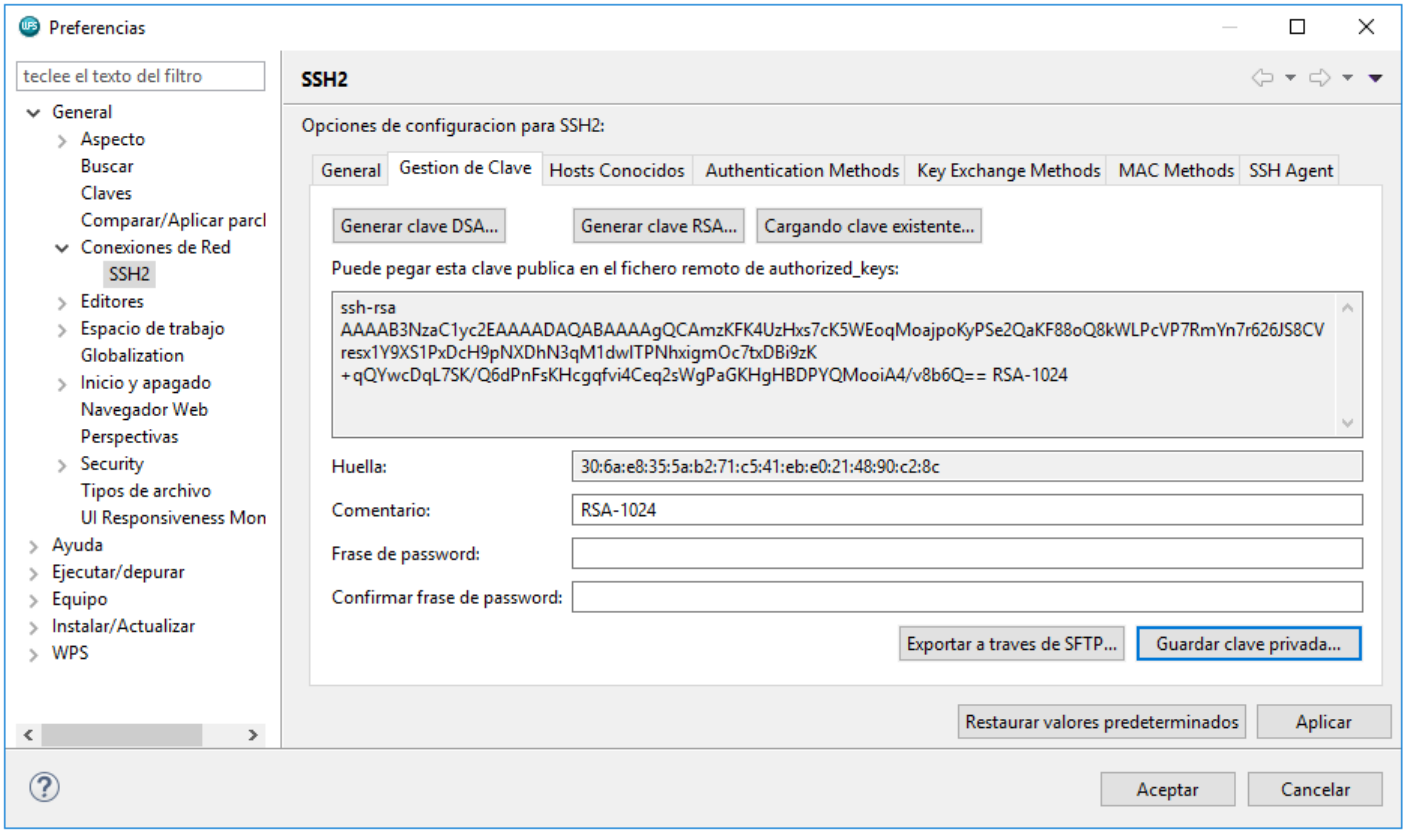
Si tiene la intención de usar *Autenticación de clave pública* (pág. 55) y las teclas no se han generado en el servidor, puede desear descargar PuTTYgen (consulte *Generación de claves mediante PuTTYgen* (pág. 56)). Si tiene la intención de utilizar claves públicas con una **frase de contraseña** y está utilizando **WPS Communicate**, también tendrá que descargar un agente de llavero tal como Pageant (consulte *Autenticación de la frase de contraseña mediante Pageant* (pág. 68)). El uso de dicho agente para la **frase de contraseña** no es necesario con **WPS Link**, aunque pueda ser deseable si se está conectando a varios servidores.

Si está utilizando **WPS Link** y ya tiene una clave privada y desea aplicarla, continúe de la manera siguiente:

1. En el menú principal de WPS Workbench, seleccione **Ventana ➤ Preferencias** y, en el recuadro izquierdo del diálogo **Preferencias** posterior, expanda los nodos **General ➤ Conexiones de red ➤ SSH2**.
2. Seleccione la pestaña **Administración de claves** del diálogo **Preferencias**:



3. Haga clic en **Cargar clave existente...**.
4. Examine la clave privada requerida y selecciónela, para visualizar la pantalla que se muestra en el siguiente ejemplo:

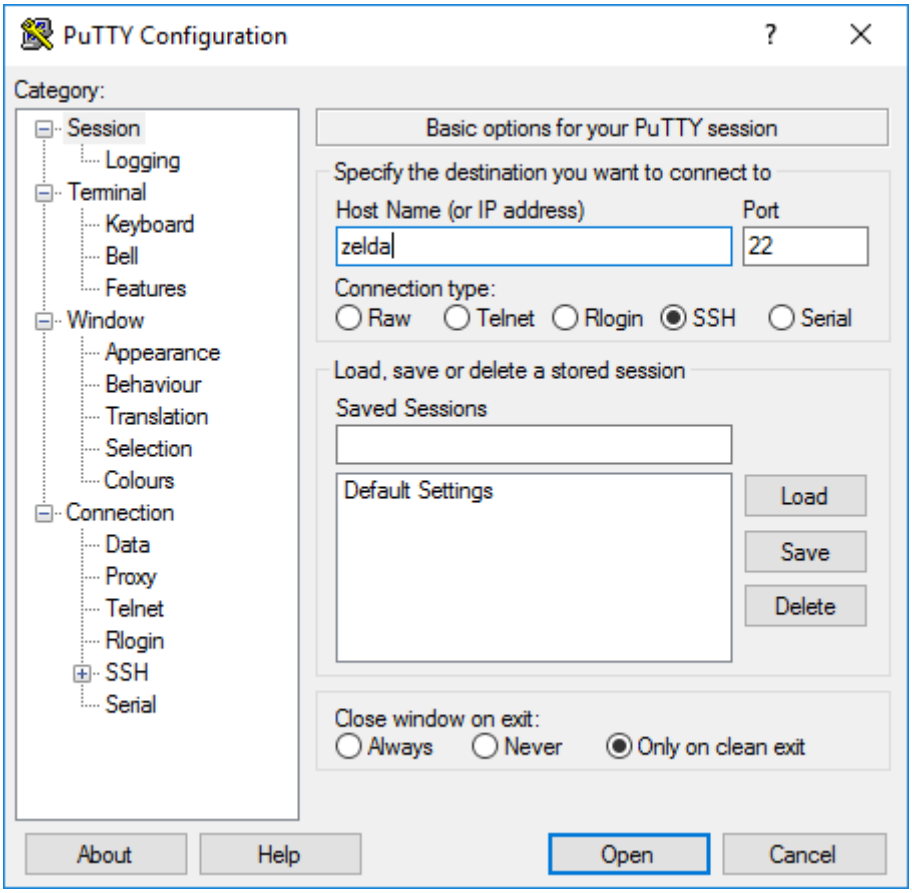


5. Si desea aplicar una frase de contraseña a su archivo de claves privadas, complete los campos **Frase de contraseña** y **Confirmar frase de contraseña**.
6. Haga clic en **Aceptar** para guardar los cambios y cerrar la ventana **Preferencias**.

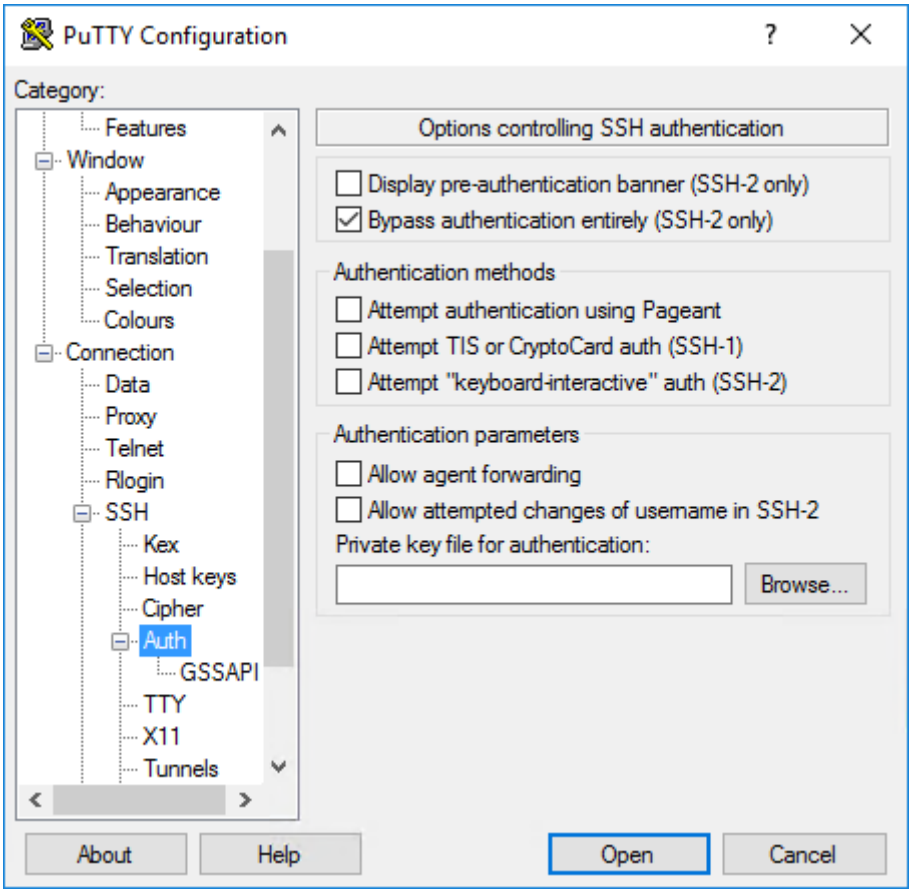
Autenticación de contraseña (usando PuTTY) e inicio de sesión de WPS

El inicio de sesión SSH manual proporciona la oportunidad de ejecutar la validación de la clave de host. Para mayor seguridad, WPS ejecuta la validación de la clave de host durante el inicio de sesión de SSH. Sin embargo, WPS no tiene ningún mecanismo para interactuar con el usuario para aceptar nuevas claves de host, o para preguntar acerca de un cambio aparente de clave. En su lugar, WPS se basa en la aceptación de la clave de host que ya un cliente SSH externo ha realizado, y validará la clave de host que recibe contra la misma base de datos utilizada por el cliente SSH externo. En los clientes Windows, WPS utilizará de manera predeterminada la base de datos de claves de host PuTTY almacenada en el registro de Windows, por lo que es necesario iniciar sesión en el host remoto utilizando el cliente PuTTY SSH para validar la clave de host y agregarlo a la base de datos de claves de host antes de intentar realizar una conexión con WPS.

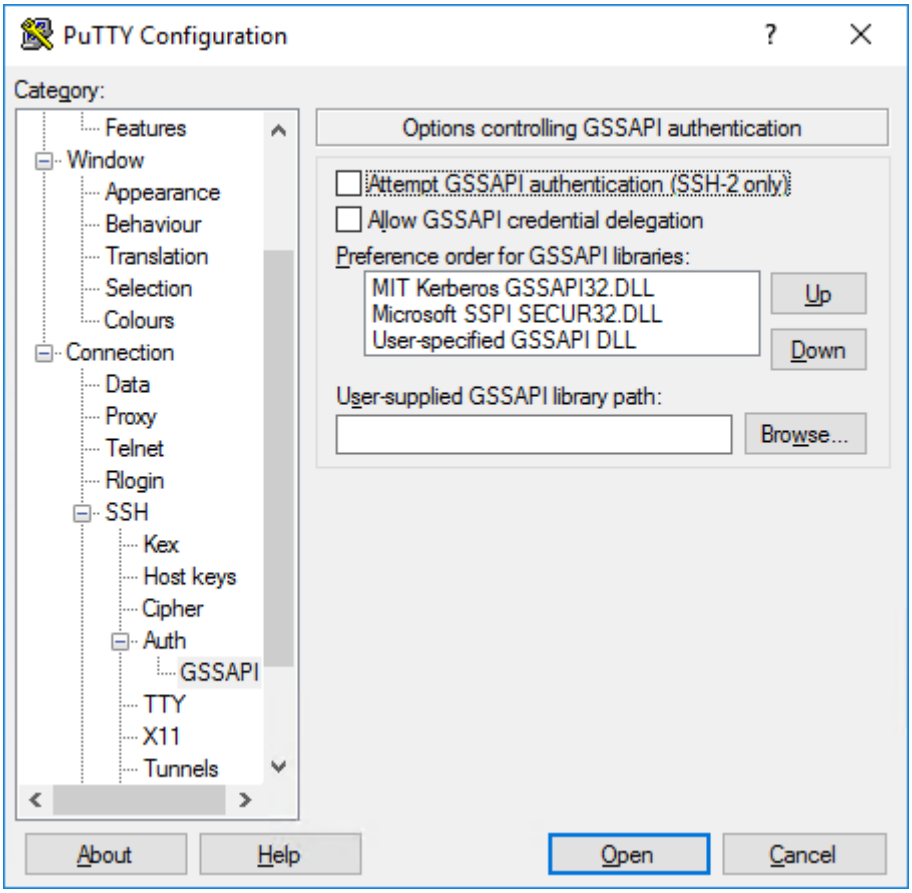
1. Inicie el cliente PuTTY e introduzca el nombre de host en el campo de entrada principal **Host Name (or IP address)**:



2. Expanda la página de configuración de **SSH ➤ Auth** desde la lista de categorías de la izquierda y asegúrese de que no esté seleccionado nada en **Authentication methods** y de que el campo **Private key file for authentication** esté vacío:



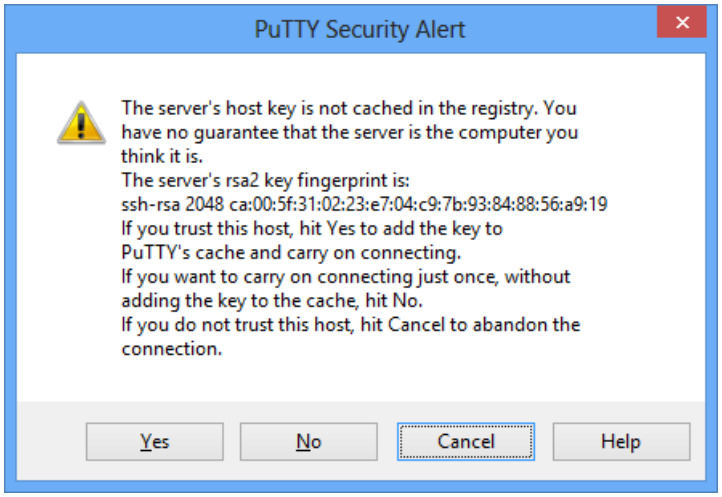
3. Seleccione la página **GSSAPI** y asegúrese de que **Attempt GSSAPI authentication** no esté seleccionado:



Estos controles garantizan que sólo se utiliza la autenticación de contraseña y que no hemos seleccionado inadvertidamente un mecanismo de autenticación más implicado.

4. Haga clic en **Open** y cuando se le pide, escriba su contraseña y presione `Entrar`.

   Si es la primera vez que se ha iniciado la sesión a este host específico, se mostrará una alerta del siguiente tipo:



En este punto debe confirmar que ésta es la huella digital correcta para el host y, suponiendo que sea así, haga clic en **Yes** para aceptar la clave de modo permanente. Esto permitirá posteriormente que WPS ejecute la validación de la clave de host utilizando la misma clave almacenada en caché. Si todo ha funcionado, se conectará a través de una sesión de terminal a su host remoto. La clave del host se habrá validado y almacenado en el registro de Windows, que es donde los componentes de WPS lo buscarán. Ahora puede desconectarse del conocimiento de que al lanzar WPS, podrá acceder al mismo servidor remoto, extrayendo automáticamente la clave del host validada desde el registro de Windows para ejecutar la validación.

Nota:

No confunda esta validación de clave de host con la autenticación de clave pública: son dos cosas completamente separadas. La validación de la clave del host simplemente le da la oportunidad de confirmar que el host al que se está conectando es, de hecho, el host al que desea conectarse.

5. Si está utilizando **WPS Link**, cree la conexión con el host requerida y el servidor de host remoto a través de **WPS Workbench**. Si está utilizando **WPS Communicate**, inicie sesión en WPS a través de la instrucción `SIGNON`, para lo cual debe especificar la opción de la instrucción IDENTITYFILE o la opción del sistema SSH_IDENTITYFILE, por ejemplo:

```
SIGNON <servername> SSH
USERNAME="<username>"
password="<password>"
LAUNCHCMD="/home/installs/wps-3.2/bin/wps -dmr ";

RSUBMIT;
%PUT &SYSHOSTNAME;
ENDRSUBMIT;
SIGNOFF;
```

Alternativamente:

```
OPTIONS SSH_IDENTITYFILE="C:\Users\techwriter\.ssh\wpscommunicate.ppk";
SIGNON <servername> SSH
password="<password>"
LAUNCHCMD="/home/installs/wps-3.2/bin/wps -dmr";

RSUBMIT;
%PUT &SYSHOSTNAME;
ENDRSUBMIT;
SIGNOFF;
```

Nota:

No puede utilizar ni **IDENTITYFILE** ni **SSH_IDENTITYFILE** si está utilizando *Autenticación de la frase de contraseña mediante Pageant* (pág. 68).

Iniciar sintaxis de comando

Si se está conectando a un servidor SSH de Windows, se requiere una ruta de acceso del comando de inicio de estilo Windows. En este ejemplo, las comillas son obligatorias porque la ruta incluye espacios:

```
'C:\Program Files\World Programming\WPS\3\bin\wps' -dmr
```

Si se está conectando a un servidor UNIX/Linux, la ruta podría ser:

```
/home/installs/wps-3.2/bin/wps -dmr
```

Mejora de la autenticación de nombre de usuario/ contraseña

Hasta ahora, los ejemplos de inicio de sesión a través de WPS mediante SSH se han basado en una instrucción `SIGNON` que contiene directamente el nombre de usuario y la contraseña. Esto no es ideal, ya que significa almacenar estos detalles confidenciales en un archivo de origen.

Actualmente, WPS no admite la solicitud de credenciales. Sin embargo, la contraseña se puede ofuscar usando `PROC PWENCODE`.

Nota:

Esto no es un método de cifrado de alta seguridad. WPS actualmente admite el mecanismo BASE64 para la ofuscación.

Para utilizar `PROC PWENCODE` con una instrucción `SIGNON`:

1. Ejecute el siguiente programa para codificar su contraseña:

   ```
   proc pwencode in="<password>" out=log;
   run;
   ```

   Esto produce algo similar al siguiente en el registro:

   ```
   43         proc pwencode in=<password> out=log;
   44         run;
   {sas001}dG9wc2VjcmV0
   NOTE: Procedure pwencode step took :
         real time : 0.004
         cpu time  : 0.000
   ```

   Nota:

   Ejecute este programa como una tarea desconectada.

2. Copie y pegue la contraseña codificada en su programa `SIGNON`:

   ```
   SIGNON docserver SSH
   username="<username>"
   password="{sas001}dG9wc2VjcmV0"
   LAUNCHCMD="/home/installs/wps-3.2/bin/wps -dmr";
   ```

Autenticación de clave pública

Este método es más seguro que el uso de una contraseña simple, y a veces se llama autenticación *sin contraseña*.

Con SSH, la autenticación mediante claves no sólo mejora la seguridad en general, sino también, en el caso de **WPS Communicate**, evita tener nombres de usuario y contraseñas comprometidas con el código fuente (incluso en forma ofuscada o cifrada).

Este método de autenticación se basa en un par de claves criptográficas, en el que la clave privada reside (y nunca abandona) el equipo cliente, y la clave pública está instalada en el servidor SSH al que el cliente necesita conectarse o, en el caso de Windows, en **Bitvise SSH Server** (de WPS 3.2 en adelante). El protocolo SSH utiliza el par de claves para establecer la identidad del cliente y ejecutar la autenticación.

Se describen dos métodos mediante los cuales se pueden generar las claves en un cliente Windows:

- *Generación de claves mediante PuTTYgen* (page 56)
- *Generación de claves mediante WPS Workbench* (page 59)

Después de la generación de las claves públicas, deben colocarse en el servidor remoto de acuerdo con *Implementación de claves públicas en el servidor SSH remoto* (page 61) o, en el caso de un servidor Windows, *Implementación de claves públicas en Bitvise SSH Server* (page 63).

Si desea conectarse a varios servidores, sin tener que recordar o introducir su contraseña para cada sistema, también debe usar *Autenticación de la frase de contraseña mediante Pageant* (page 68).

La validez de los pares de claves debe comprobarse de acuerdo con *Verificación del acceso al host remoto (mediante PuTTY) y el inicio de sesión de WPS* (page 66).

Note:

La autenticación de clave pública se puede utilizar con **WPS Communicate** y **WPS Link**. Sin embargo, para **WPS Communicate**, si no está utilizando un agente de llavero tal como Pageant, no puede haber una **frase de contraseña** en el archivo de claves privadas, ya que actualmente no existe un mecanismo interactivo para solicitarlo durante la autenticación de WPS.

Note:

Debe asegurarse de que la autenticación de clave pública no está deshabilitada en el equipo cliente.

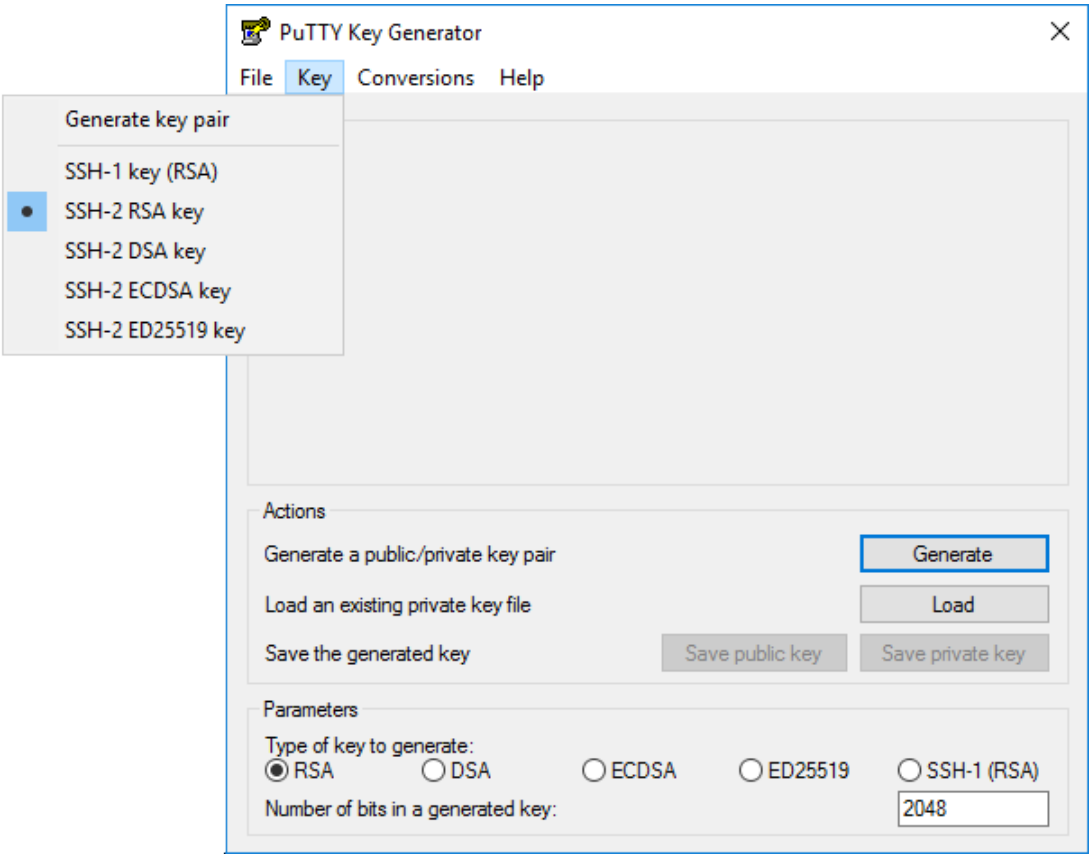
Generación de claves mediante PuTTYgen

Para generar un par de claves, que es la combinación de la clave privada y la clave pública para el cifrado asimétrico, continúe de la manera siguiente:

Nota:

Debe tener en cuenta que, como alternativa, también puede utilizar WPS Workbench para generar pares de claves (consulte *Generación de claves mediante WPS Workbench* (pág. 59)).

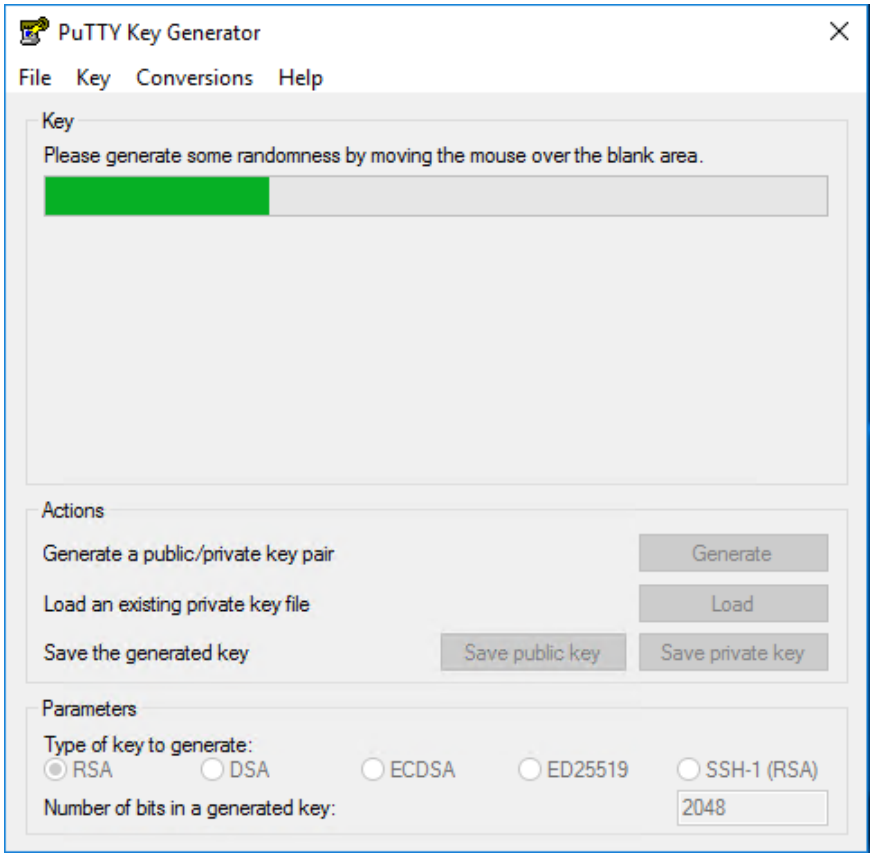
1. Inicie la herramienta PuTTYgen (disponible a través de las mismas fuentes que PuTTY).



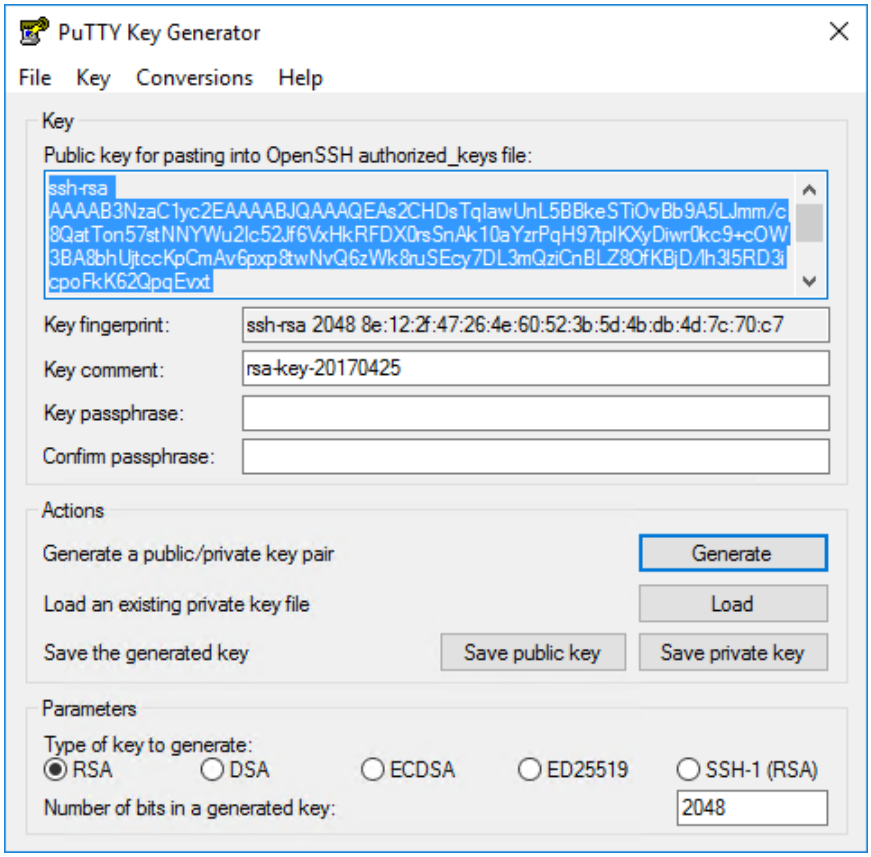
Importante:

Se recomienda utilizar el parámetro **SSH-2 RSA** con una mínima longitud de clave de **2048**.

2. Haga clic en el botón **Generate** y mueva el ratón dentro del área que se indican para generar algo de aleatoriedad, esto actuará como una semilla para su par de claves:



3. El sistema genera un par de claves:



4. Si desea aplicar una frase de contraseña a su archivo de claves privadas, complete los campos **Frase de contraseña de clave** y **Confirmar frase de contraseña**. Tendrá que hacerlo si va a usar *Autenticación de la frase de contraseña mediante Pageant* (pág. 68).

Nota:

Si está utilizando **WPS Communicate**, no introduzca una **frase de contraseña** a menos que vaya a utilizar *Autenticación de la frase de contraseña mediante Pageant* (pág. 68). Si está usando **WPS Link**, puede introducir una **frase de contraseña** y usarla con o sin *Autenticación de la frase de contraseña mediante Pageant* (pág. 68).

---

5. Haga clic en **Guardar clave privada**. Si no ha introducido una **frase de contraseña**, se le pedirá que confirme que desea guardar la clave sin una **frase de contraseña**. El archivo resultante está en el formato nativo de PuTTY (`*.PPK`) y, cuando se le pide que guarde el archivo en una carpeta, debe asegurarse de que esté almacenado en la carpeta `.ssh` en su perfil de usuario.

Nota:

Asegúrese de que los permisos de su archivo de claves privadas son tales que sólo usted puede leerlo. Este archivo es esencialmente su contraseña, así que es importante que nadie más pueda acceder al archivo.

---

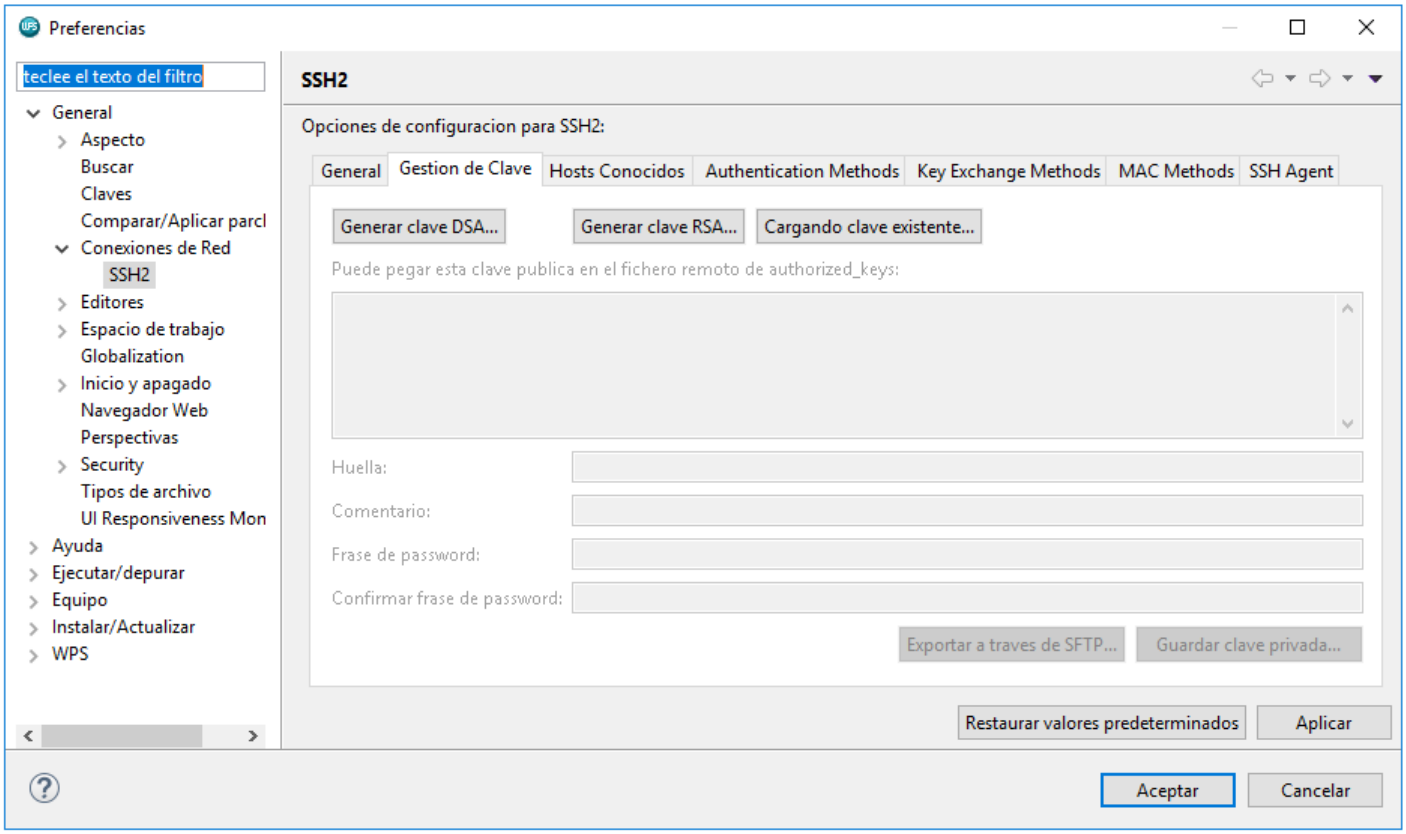
6. La clave pública que se resalta en el paso *3* (pág. 57) contiene la información necesaria para permitir que un usuario verifique que otra parte está en posesión de la clave privada correspondiente. La clave pública no necesita mantenerse segura, pero debe guardarla copiándola en su área de pegado y pegándola en un archivo de texto sin formato, o seleccionando **Guardar clave pública** para guardarla en la carpeta `.ssh` en su perfil de usuario. Luego continúe como en *Implementación de claves públicas en el servidor SSH remoto* (pág. 61) o *Implementación de claves públicas en Bitvise SSH Server* (pág. 63).

Nota:

Si va a usar **WPS Link** junto con **WPS Communicate**, para evitar la necesidad de dos pares de claves separados, debe tener una estrategia coherente, es decir evitar una **frase de contraseña** en ambos casos, o bien crear una sola **frase de contraseña** y asociarla con un solo par de claves a través de *Autenticación de la frase de contraseña mediante Pageant* (pág. 68).

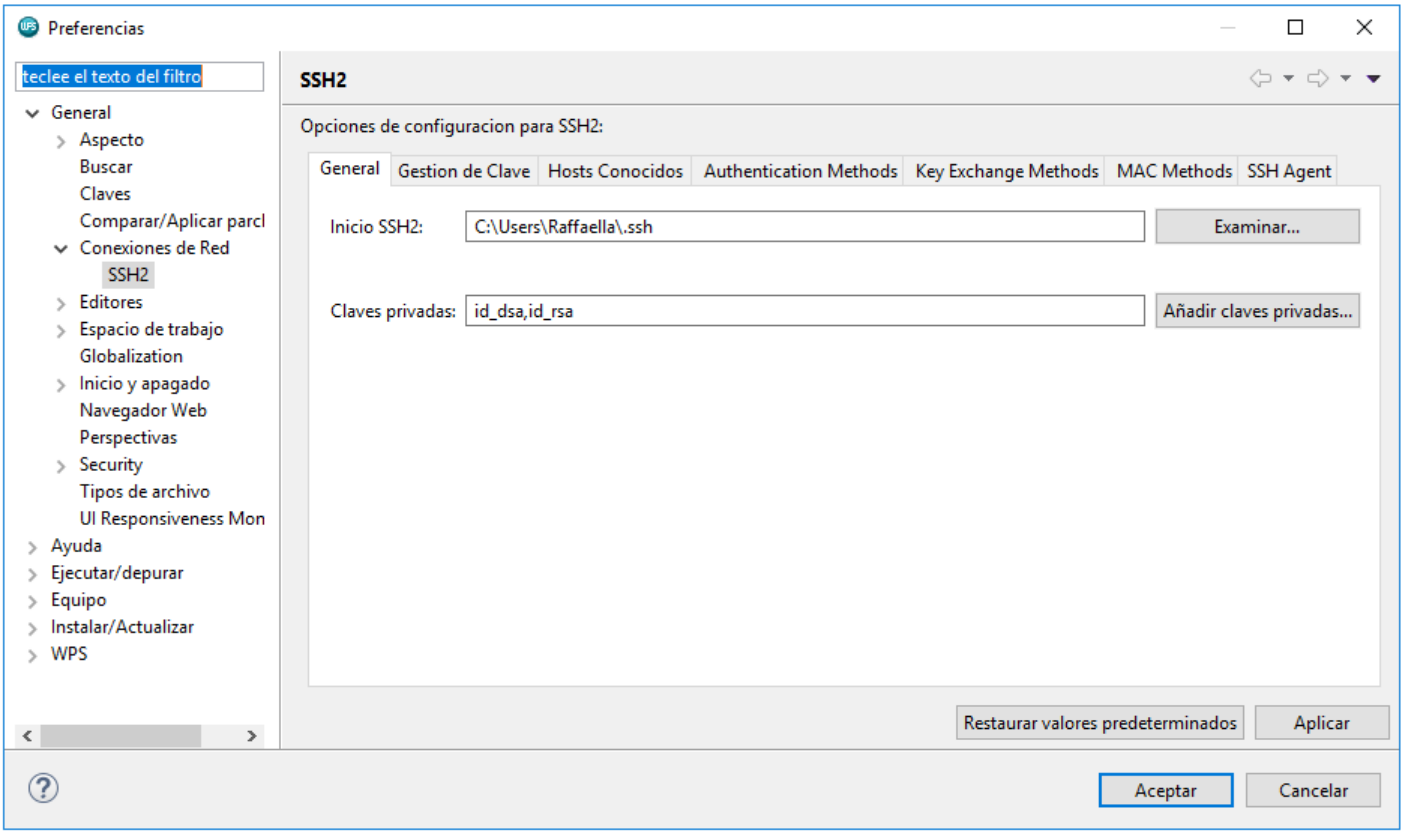
Generación de claves mediante WPS Workbench

1. En el menú principal de WPS Workbench, seleccione **Ventana ➤ Preferencias** y, en el recuadro izquierdo del diálogo **Preferencias** posterior, expanda los nodos **General ➤ Conexiones de red ➤ SSH2**.
2. Seleccione la pestaña **Administración de claves** del diálogo **Preferencias**:



3. Haga clic en **Generar clave RSA...**.

   Se genera un par de claves: la clave pública se visualiza en un cuadro de texto en el centro del diálogo, por ejemplo:



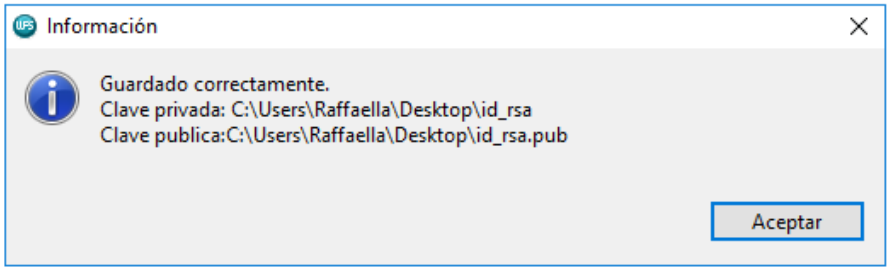
4. Si desea aplicar una frase de contraseña a su archivo de claves privadas, complete los campos **Frase de contraseña** y **Confirmar frase de contraseña**. Tendrá que hacerlo si va a usar *Autenticación de la frase de contraseña mediante Pageant* (pág. 68).

   Nota:

   Si está utilizando **WPS Communicate**, no introduzca una **frase de contraseña** a menos que vaya a utilizar *Autenticación de la frase de contraseña mediante Pageant* (pág. 68). Si está usando **WPS Link**, puede introducir una **frase de contraseña** y usarla con o sin *Autenticación de la frase de contraseña mediante Pageant* (pág. 68).

5. Haga clic en **Guardar clave privada**. Si no ha introducido una **frase de contraseña**, se le pedirá que confirme que desea guardar la clave sin una **frase de contraseña**. Cuando se le pida que guarde el archivo de clave resultante en una carpeta, asegúrese de que esté almacenado en la carpeta `.ssh` de su perfil de usuario. Si no desea utilizar el nombre predeterminado de `id_rsa`, confiere al archivo un nombre más significativo.

   WPS Workbench muestra un diálogo de información que confirma que ha guardado su archivo de claves privadas, junto con el archivo de claves públicas correspondiente. Le da al archivo de claves públicas el mismo prefijo que su archivo de claves privadas, pero le anexa `.pub`, por ejemplo:



Nota:

Asegúrese de que los permisos de su archivo de claves privadas son tales que sólo usted puede leerlo. Este archivo es esencialmente su contraseña, así que es importante que nadie más pueda acceder al archivo.

6. Haga clic en **Aceptar** para cerrar el diálogo **Información**.
7. Haga clic en **Aceptar** para guardar los cambios y cerrar la ventana **Preferencias**.
8. La clave pública que se muestra en la pantalla (para copiar y pegar si es necesario) y se guarda en un archivo, contiene la información necesaria para permitir a un usuario verificar que otra parte está en posesión de la clave privada correspondiente. Luego continúe como en *Implementación de claves públicas en el servidor SSH remoto* (pág. 61) o *Implementación de claves públicas en Bitvise SSH Server* (pág. 63).

Nota:

Si va a usar **WPS Link** junto con **WPS Communicate**, para evitar la necesidad de dos pares de claves separados, debe tener una estrategia coherente, es decir evitar una **frase de contraseña** en ambos casos, o bien crear una sola **frase de contraseña** y asociarla con un solo par de claves a través de *Autenticación de la frase de contraseña mediante Pageant* (pág. 68).

Implementación de claves públicas en el servidor SSH remoto

1. Inicie sesión en la máquina remota.
2. Una vez conectado, debe configurar el servidor para que acepte su clave pública para la autenticación, así que cambie al directorio `.ssh` y abra el archivo `authorized_keys`.

   Si esta es la primera clave pública que se va a poner en el archivo, es posible que necesite crear el directorio y el archivo primero, por ejemplo, ejecutando los siguientes comandos:

   ```
   mkdir -p .ssh
   touch ~/.ssh/authorized_keys
   ```

3. Establezca los permisos correctos, por ejemplo:

   ```
   chmod 600 ~/.ssh/authorized_keys
   ```

Nota:

También debe asegurarse de que los directorios `$HOME` y `.ssh` tengan los permisos adecuados tanto para el servidor como para su operación específica.

4. Ahora puede agregar la clave pública al archivo `authorized_keys`, como en el siguiente ejemplo:

```
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

Si actualmente tiene acceso SSH basado en contraseña, configurado en su servidor y tiene la utilidad `ssh-copy-id` instalada, puede simplemente transferir su clave pública escribiendo:

```
ssh-copy-id username@remote_host
```

A continuación, se le solicitará la contraseña de la cuenta de usuario en el sistema remoto. Después de escribir la contraseña, el contenido de la clave `~/.ssh/id_rsa.pub` se anexará al final del archivo `~/.ssh/authorized_keys` de la cuenta de usuario. A continuación, puede iniciar sesión en esa cuenta sin una contraseña:

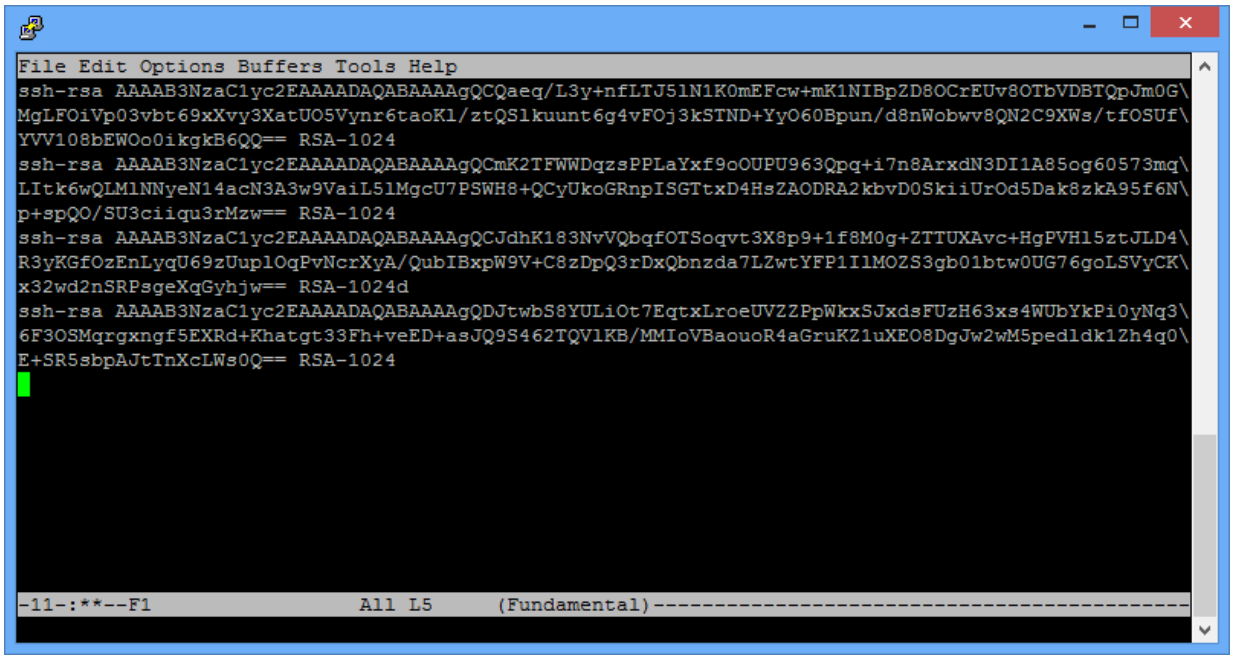
```
ssh username@remote_host
```

Alternativamente, puede copiar y pegar la clave pública desde PuTTYgen o **WPS Workbench** en el archivo `authorized_keys`, asegurándose de que termina en una sola línea.

5. Compruebe el contenido de `~/.ssh/authorized_keys` para asegurarse de que su clave pública se haya agregado correctamente, introduciendo lo siguiente en la línea de comandos:

```
more ~/.ssh/authorized_keys
```

El contenido de un archivo `~/.ssh/authorized_keys` típico podría parecerse a:



Nota:

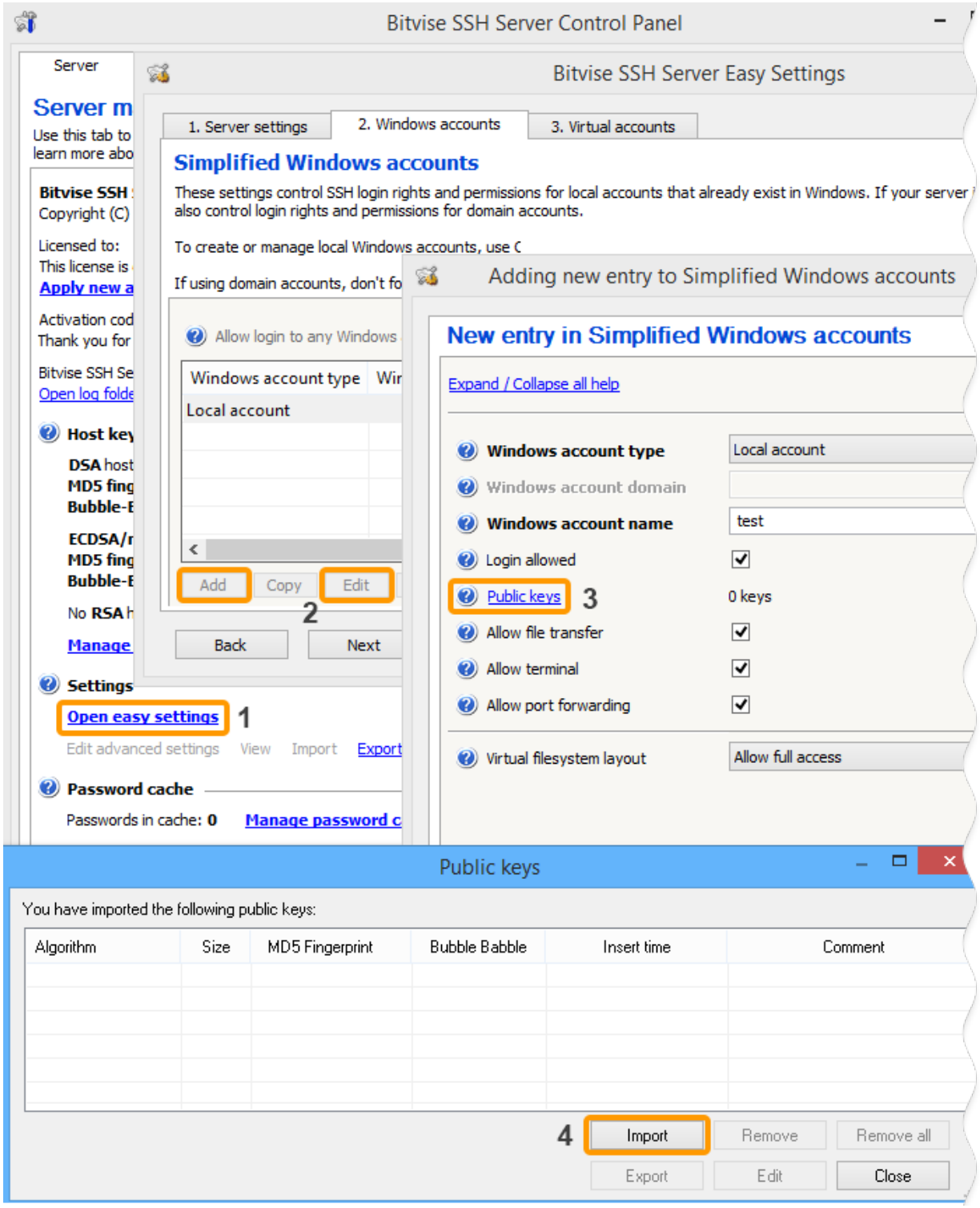
Si observa cuidadosamente, puede ver que el archivo anterior contiene cuatro claves públicas, cada una empieza con `ssh-rsa` y termina con una expresión similar a `RSA-1024`.

Implementación de claves públicas en Bitvise SSH Server

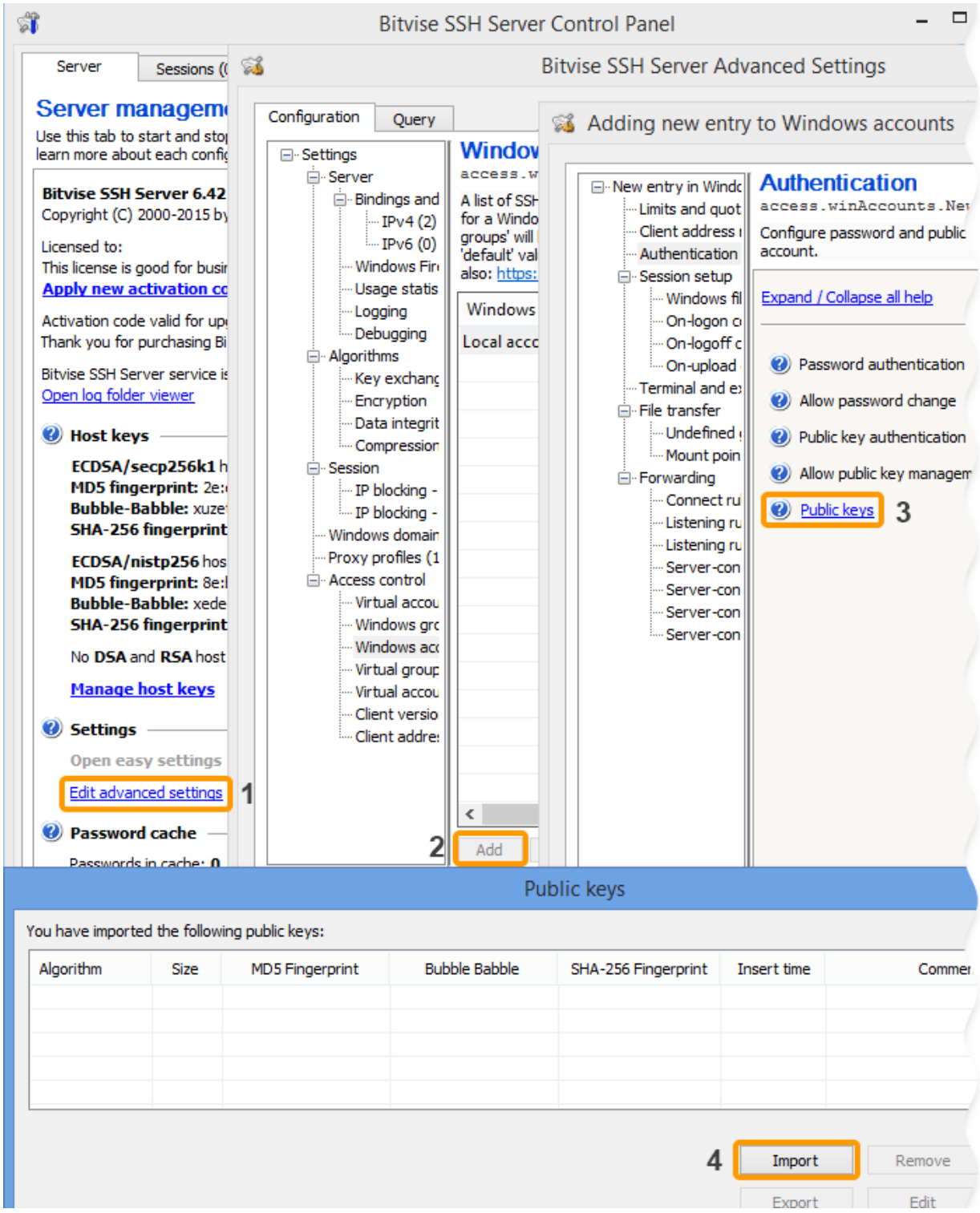
Si es novato en Bitvise SSH Server (consulte la documentación de Bitvise para detalles de configuración específicos), le recomendamos que primero se asegure de que pueda establecer una conexión SSH en funcionamiento antes de cambiar cualquier configuración en el servidor. Si no puede conectarse al servidor SSH usando su configuración predeterminada, esto es más probable debido a un problema de red o firewall que deberá resolver antes de poder conectarse. En su configuración predeterminada, Bitvise SSH Server acepta conexiones en el número de puerto utilizado a menudo para servidores SSH, 22. Este es el único puerto que debe abrir en su firewall para conectarse al servidor SSH. Si utiliza el reenvío de puerto para tunelizar otras aplicaciones a través de SSH, **no** debe abrir ningún puerto adicional para las conexiones de túnel. Todas las conexiones de túnel se reenvían a través de la sesión SSH, establecida mediante el puerto 22.

1. Al conectarse a Bitvise SSH Server con un cliente SSH por primera vez, inicie sesión con el nombre de usuario y la contraseña de una cuenta de Windows que existe en la máquina donde se ejecuta el servidor SSH. Para iniciar sesión en una cuenta de dominio de Windows, especifíquela en el formato `domain\account`.

   Puede utilizar cualquier cliente SSH para iniciar sesión en Bitvise SSH Server, siempre y cuando admita el protocolo SSH versión 2.
2. Después de asegurarse de que la clave pública se haya guardado en un archivo, transfiérala a la máquina donde está instalado Bitvise SSH Server o a la máquina desde la que administra el servidor SSH de forma remota utilizando Bitvise SSH Client.
3. Abra el **SSH Server Control Panel** y, a continuación, para importar la clave pública en la configuración de la cuenta del usuario SSH, utilice **Open easy settings**:



o **Edit advanced settings**:

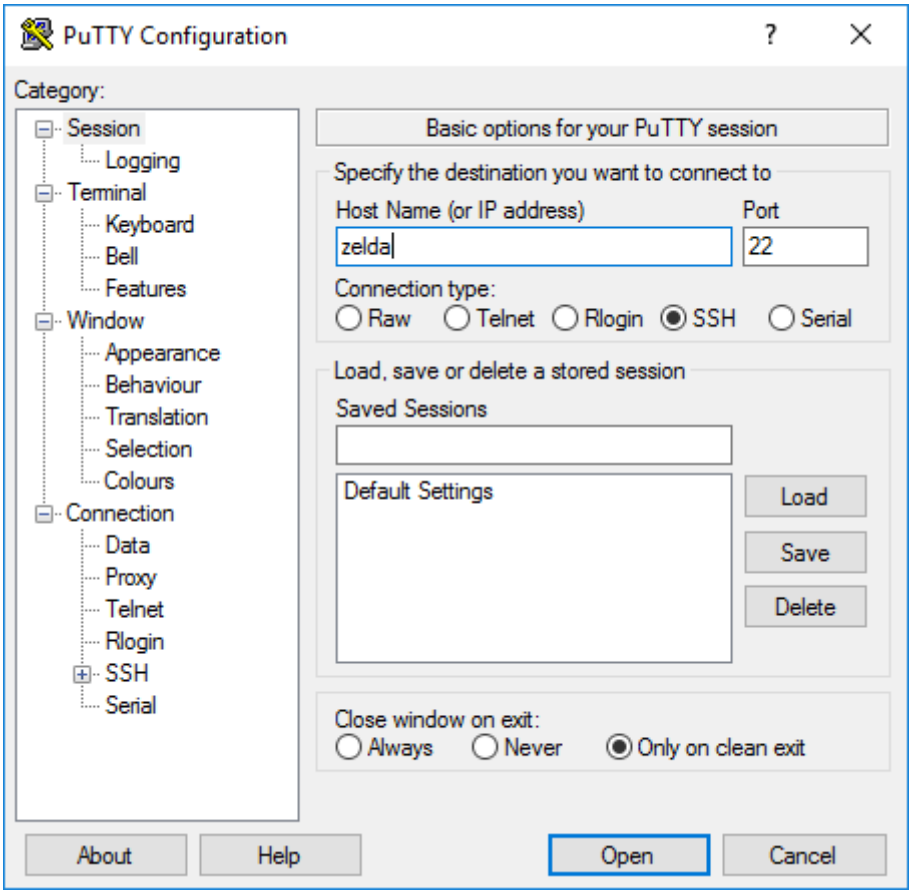

Bitvise SSH Server Control Panel
Server
Sessions
Server managem
Use this tab to start and sto
learn more about each confi
Bitvise SSH Server 6.42
Copyright (C) 2000-2015 by
Licensed to:
This license is good for busir
Apply new activation co
Activation code valid for up
Thank you for purchasing Bi
Bitvise SSH Server service is
Open log folder viewer
Host keys
ECDSA/secp256k1 h
MD5 fingerprint: 2e:
Bubble-Babble: xuze
SHA-256 fingerprint
ECDSA/nistp256 hos
MD5 fingerprint: 8e:
Bubble-Babble: xede
SHA-256 fingerprint
No DSA and RSA host
Manage host keys
Settings
Open easy settings
Edit advanced settings
1
Password cache
Passwords in cache: 0
Bitvise SSH Server Advanced Settings
Configuration
Query
Settings
Server
Bindings and
IPv4 (2)
IPv6 (0)
Windows Fir
Usage statis
Logging
Debugging
Algorithms
Key exchang
Encryption
Data integrit
Compressior
Session
IP blocking -
IP blocking -
Windows domair
Proxy profiles (1
Access control
Virtual accou
Windows gro
Windows acc
Virtual group
Virtual accou
Client versio
Client addres
Windov
access.w
A list of SSH
for a Windo
groups' will
'default' val
also: https:
Windows
Local acco
2
Add
Adding new entry to Windows accounts
New entry in Windc
Limits and quot
Client address
Authentication
Session setup
Windows fil
On-logon c
On-logoff c
On-upload
Terminal and e
File transfer
Undefined
Mount poin
Forwarding
Connect ru
Listening ru
Listening ru
Server-con
Server-con
Server-con
Server-con
Authentication
access.winAccounts.Ne
Configure password and public
account.
Expand / Collapse all help
Password authentication
Allow password change
Public key authentication
Allow public key manager
Public keys
3
Public keys
You have imported the following public keys:
Algorithm
Size
MD5 Fingerprint
Bubble Babble
SHA-256 Fingerprint
Insert time
Commer
4
Import
Remove
Export
Edit

Nota:

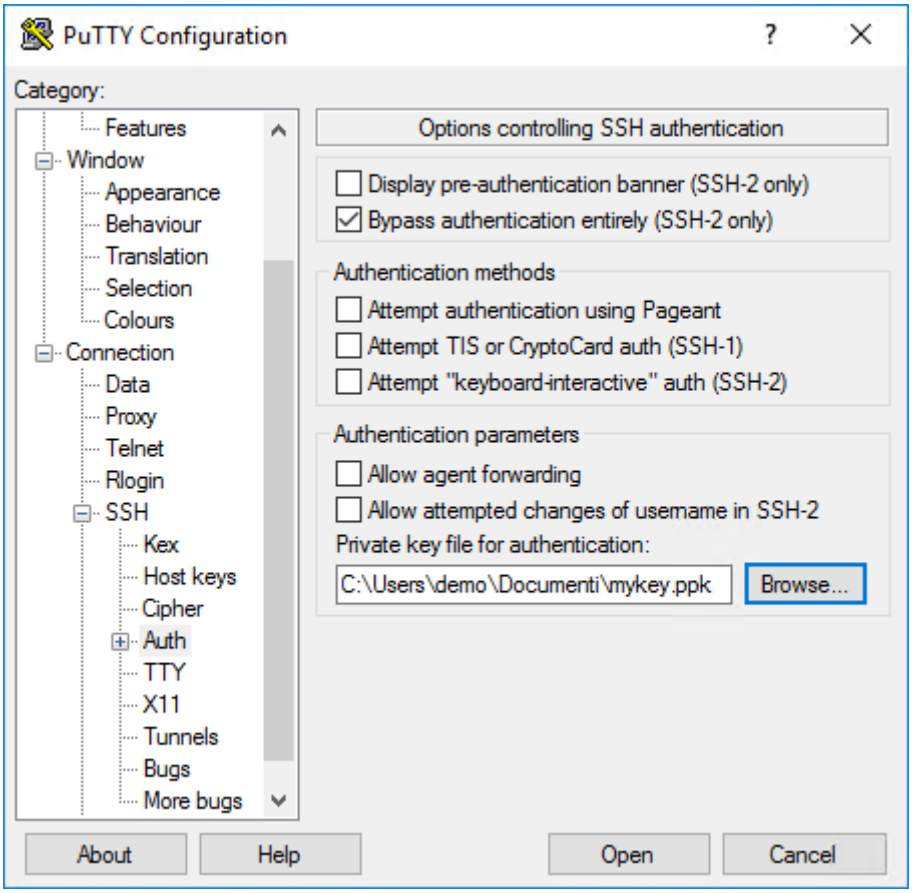
Para las cuentas de Windows, Bitvise SSH Server también admite la sincronización con `~/.ssh/authorized_keys`, siempre que esta función esté habilitada en **Advanced SSH Server settings**, bajo **Access control**. Si esta función está habilitada, Bitvise SSH Server comprobará la existencia del archivo `authorized_keys` cuando el usuario cierre la sesión. Si el archivo existe, Bitvise SSH Server reemplazará todas las claves públicas configuradas para el usuario con las claves encontradas en este archivo.

Verificación del acceso al host remoto (mediante PuTTY) y el inicio de sesión de WPS

1. Verifique la autenticación mediante PuTTY de la manera siguiente.
   a. Inicie el cliente PuTTY e introduzca el nombre del host en el campo **Nombre de host**:



   b. Seleccione la página de configuración **SSH ➤ Auth** desde la lista de categorías de la izquierda y asegúrese de que no esté seleccionado nada en **Authentication methods** (a menos que esté verificando *Autenticación de la frase de contraseña mediante Pageant* (pág. 68), en cuyo caso se debe seleccionar **Attempt authentication using Pageant**).



c. En el campo **Private key file for authentication**, introduzca el nombre del archivo de claves privadas que ha generado mediante *Generación de claves mediante PuTTYgen* (pág. 56) o *Generación de claves mediante WPS Workbench* (pág. 59).

d. Haga clic en **Abrir**, se autenticará a través de su combinación de pares de claves y se abrirá una ventana de consola. La ventana visualiza una notificación acerca del método de autenticación.

Nota:

Si se le pide una contraseña, se ha producido un error con la autenticación de la clave pública.

---

2. Si la autenticación de la clave pública se finaliza correctamente, si está utilizando **WPS Link**, cree la conexión con el host requerida y el servidor del host remoto a través de **WPS Workbench**. Si está utilizando **WPS Communicate**, inicie sesión en WPS usando su clave privada, a través de la instrucción `SIGNON`, para lo cual debe especificar la opción de la instrucción **IDENTITYFILE** o la opción del sistema **SSH_IDENTITYFILE**, por ejemplo:

```
SIGNON <servername> SSH
USERNAME="<username>"
IDENTITYFILE="C:\Users\techwriter\.ssh\wpscommunicate.ppk"
LAUNCHCMD="/home/installs/wps-3.2/bin/wps -dmr ";
```

Alternativamente:

```
OPTIONS SSH_IDENTITYFILE="C:\Users\techwriter\.ssh\wpscommunicate.ppk";
SIGNON <servername> SSH
username="<username>"
LAUNCHCMD="/home/installs/wps-3.2/bin/wps -dmr";
```

Nota:

No puede utilizar ni **IDENTITYFILE** ni **SSH_IDENTITYFILE** si está utilizando *Autenticación de la frase de contraseña mediante Pageant* (pág. 68).

Autenticación de la frase de contraseña mediante Pageant

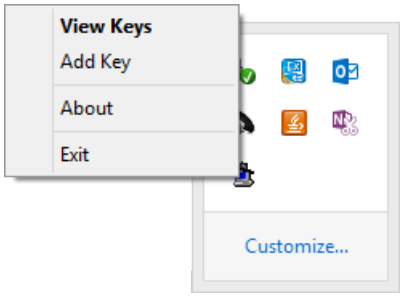
WPS Communicate no admite la lectura de archivos de claves privadas que se han guardado en forma cifrada con una **frase de contraseña**. Sin embargo, todavía es posible utilizar pares de claves privadas de esta forma si utiliza un agente de llavero. Supondremos que está usando Pageant en Windows.

Un agente de llavero es una utilidad que se ejecuta en el equipo cliente y almacena el archivo de claves públicas descifrado. El cliente SSH (o WPS al ejecutar un inicio de sesión SSH) contacta al agente para que una clave pública se use cuando se conecta a un host determinado. Aunque todavía se requiere una contraseña para descifrar la clave privada, esto sólo se requiere una vez, cuando el agente de llavero primero abre la clave privada.

Este procedimiento supone que ya ha generado un par de claves con una **frase de contraseña** e ha implementado la clave pública en el servidor SSH remoto (para UNIX/Linux) o en Bitvise SSH Server (para Windows).

Continúe de la manera siguiente:

1. Ejecute la herramienta Pageant, haga clic con el botón secundario en el icono de la **bandeja del sistema** y elija **View Keys** o **Add Keys** para agregar su archivo de claves privadas.



   En este momento se le solicitará la **frase de contraseña** para el archivo de claves privadas.

2. Pageant abre y descifra, permitiendo a los clientes (tal como el programa normal PuTTY o WPS) solicitar la lista de identidad cargada para la autenticación.

Nota:
WPS detecta automáticamente si se está ejecutando Pageant.

SSH (Secure Shell) desde un cliente UNIX

Antes de acceder a un host remoto mediante **WPS Communicate** o **WPS Link**, es importante asegurarse de que puede acceder al host remoto manualmente a través de SSH. Esto demuestra que puede al menos conectarse a la máquina mediante el protocolo SSH y que su ID de usuario y contraseña son válidos.

Nota:
Si tiene la intención de usar *Autenticación de clave pública* (pág. 70) y las teclas no se han generado en el servidor, entonces puede que desee usar **ssh-keygen** (consulte *Generación de claves mediante ssh-keygen* (pág. 71)). Si tiene la intención de utilizar claves públicas con una **frase de contraseña** y está utilizando **WPS Communicate**, también tendrá que descargar un agente de llavero tal como **ssh-agent** (consulte *Autenticación de la frase de contraseña mediante ssh-agent* (pág. 77)). El uso de dicho agente para la **frase de contraseña** no es necesario con **WPS Link**, aunque pueda ser deseable si se está conectando a varios servidores.

Autenticación de contraseña e inicio de sesión de WPS

WPS utilizará de manera predeterminada la base de datos de claves del host OpenSSH almacenada en el archivo `~/.ssh/known_hosts`. Es necesario iniciar sesión en el host remoto utilizando la línea de comandos del cliente de OpenSSH para validar y aceptar la clave de host y asegurarse de que se agregue al archivo `known_hosts` antes de intentar establecer una conexión con WPS. Con OpenSSH, también es posible que un administrador del sistema agregue claves manualmente al archivo `/etc/ssh/ssh_known_hosts`, en vez del archivo `~/.ssh/known_hosts`.

Para iniciar sesión mediante SSH en un host remoto y asegurarse de que se utilice la autenticación de contraseña:

1. Introduzca el siguiente comando:

```
ssh -o PreferredAuthentications=password <hostname>
```

2. Compruebe que recibe una solicitud de contraseña:

```
<user>@<hostname>'s password:
```

Nota:

Es importante asegurarse de que se está utilizando la autenticación de contraseña y que, por ejemplo, el host no está configurado para aceptar sólo el inicio de sesión de clave pública.

3. Si está utilizando **WPS Link**, cree la conexión con el host requerida y el servidor de host remoto a través de **WPS Workbench**. Si está utilizando **WPS Communicate**, inicie sesión en WPS a través de la instrucción `SIGNON`, para lo cual debe especificar la opción de la instrucción **IDENTITYFILE** o la opción del sistema **SSH_IDENTITYFILE**, por ejemplo:

```
SIGNON <servername> SSH
USERNAME="<username>"
password="<password>"
LAUNCHCMD="/home/installs/wps-3.2/bin/wps -dmr ";

RSUBMIT;
%PUT &SYSHOSTNAME;
ENDRSUBMIT;
SIGNOFF;
```

Alternativamente:

```
OPTIONS SSH_IDENTITYFILE="C:\Users\techwriter\.ssh\wpscommunicate.ppk";
SIGNON <servername> SSH
password="<password>"
LAUNCHCMD="/home/installs/wps-3.2/bin/wps -dmr";

RSUBMIT;
%PUT &SYSHOSTNAME;
ENDRSUBMIT;
SIGNOFF;
```

Nota:

No puede utilizar ni **IDENTITYFILE** ni **SSH_IDENTITYFILE** si está utilizando *Autenticación de la frase de contraseña mediante ssh-agent* (pág. 77).

Autenticación de clave pública

El método descrito por el cual las claves se pueden generar en un cliente UNIX, es *Generación de claves mediante ssh-keygen* (pág. 71).

Después de la generación de las claves públicas, deben colocarse en el servidor remoto de acuerdo con *Implementación de claves públicas en el servidor SSH remoto* (pág. 61) o, en el caso de un servidor Windows, *Implementación de claves públicas en Bitvise SSH Server* (pág. 63).

Si desea conectarse a varios servidores, sin tener que recordar o introducir su contraseña para cada sistema, también debe usar *Autenticación de la frase de contraseña mediante ssh-agent* (pág. 77).

La validez de los pares de claves debe comprobarse de acuerdo con *Verificación del acceso al host remoto y el inicio de sesión de WPS* (pág. 76).

Nota:

La autenticación de clave pública se puede utilizar con **WPS Communicate** y **WPS Link**. Sin embargo, para **WPS Communicate**, si no está utilizando un agente de llavero tal como **ssh-agent**, no puede haber una **frase de contraseña** en el archivo de claves privadas, ya que actualmente no existe un mecanismo interactivo para solicitarlo durante la autenticación de WPS.

Nota:

Debe asegurarse de que la autenticación de clave pública no está deshabilitada en el equipo cliente.

Generación de claves mediante ssh-keygen

El programa **ssh-keygen** le permite crear claves RSA para su uso por la versión 2 del protocolo SSH. El tipo de clave a generar se especifica mediante la opción `-t`. Si se invoca sin argumentos, **ssh-keygen** generará una clave RSA para su uso en las conexiones del protocolo SSH 2.

1. Genere el par de claves. En el siguiente ejemplo, hemos iniciado sesión en `hostA` como `wplusr`:

```
ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/wplusr/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/wplusr/.ssh/id_rsa.
Your public key has been saved in /home/wplusr/.ssh/id_rsa.pub.
The key fingerprint is:
f6:61:a8:27:35:cf:4c:6d:13:22:70:cf:4c:c8:a0:23 wplusr@hostA
```

   Nota:

   No introduzca una **frase de contraseña** si está usando **WPS Communicate**. Debe utilizar un agente de llavero para esto (consulte *Autenticación de la frase de contraseña mediante ssh-agent* (pág. 77)). En el ejemplo anterior, la clave privada se ha guardado en `.ssh/id_rsa` (este archivo es de sólo lectura y sólo para usted, y nadie más debe ver el contenido de ese archivo, ya que se utiliza para descifrar toda la correspondencia cifrada con la clave pública) y la clave pública en `.ssh/id_rsa.pub`. Este es el archivo que debe agregarse al archivo `~/.ssh/authorized_keys` en la máquina remota.

2. Para implementar la clave pública, continúe como en *Implementación de claves públicas en el servidor SSH remoto* (pág. 61) o *Implementación de claves públicas en Bitvise SSH Server* (pág. 63).

Implementación de claves públicas en el servidor SSH remoto

1. Inicie sesión en la máquina remota.

2. Una vez conectado, debe configurar el servidor para que acepte su clave pública para la autenticación, así que cambie al directorio `.ssh` y abra el archivo `authorized_keys`.

   Si esta es la primera clave pública que se va a poner en el archivo, es posible que necesite crear el directorio y el archivo primero, por ejemplo, ejecutando los siguientes comandos:

   ```
   mkdir -p .ssh
   touch ~/.ssh/authorized_keys
   ```

3. Establezca los permisos correctos, por ejemplo:

   ```
   chmod 600 ~/.ssh/authorized_keys
   ```

   Nota:

   También debe asegurarse de que los directorios `$HOME` y `.ssh` tengan los permisos adecuados tanto para el servidor como para su operación específica.

4. Ahora puede agregar la clave pública al archivo `authorized_keys`, como en el siguiente ejemplo:

   ```
   cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
   ```

   Si actualmente tiene acceso SSH basado en contraseña, configurado en su servidor y tiene la utilidad `ssh-copy-id` instalada, puede simplemente transferir su clave pública escribiendo:

   ```
   ssh-copy-id username@remote_host
   ```

   A continuación, se le solicitará la contraseña de la cuenta de usuario en el sistema remoto. Después de escribir la contraseña, el contenido de la clave `~/.ssh/id_rsa.pub` se anexará al final del archivo `~/.ssh/authorized_keys` de la cuenta de usuario. A continuación, puede iniciar sesión en esa cuenta sin una contraseña:

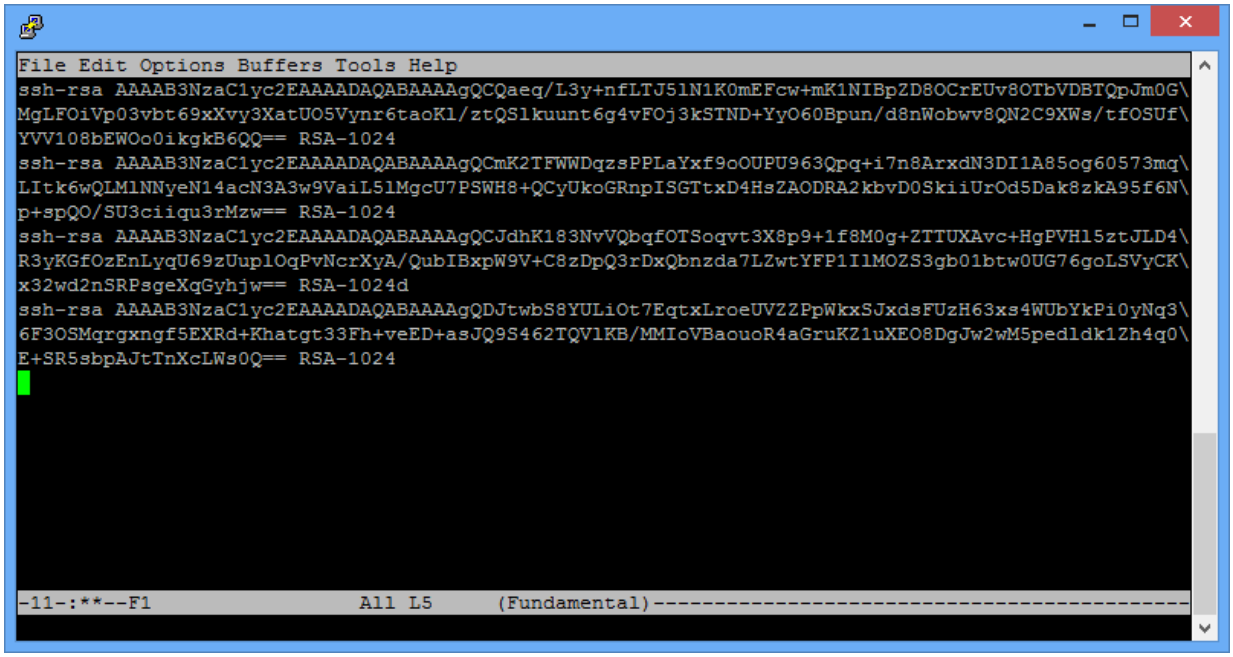
   ```
   ssh username@remote_host
   ```

   Alternativamente, puede copiar y pegar la clave pública desde PuTTYgen o **WPS Workbench** en el archivo `authorized_keys`, asegurándose de que termina en una sola línea.

5. Compruebe el contenido de `~/.ssh/authorized_keys` para asegurarse de que su clave pública se haya agregado correctamente, introduciendo lo siguiente en la línea de comandos:

   ```
   more ~/.ssh/authorized_keys
   ```

   El contenido de un archivo `~/.ssh/authorized_keys` típico podría parecerse a:



Nota:

Si observa cuidadosamente, puede ver que el archivo anterior contiene cuatro claves públicas, cada una empieza con `ssh-rsa` y termina con una expresión similar a `RSA-1024`.

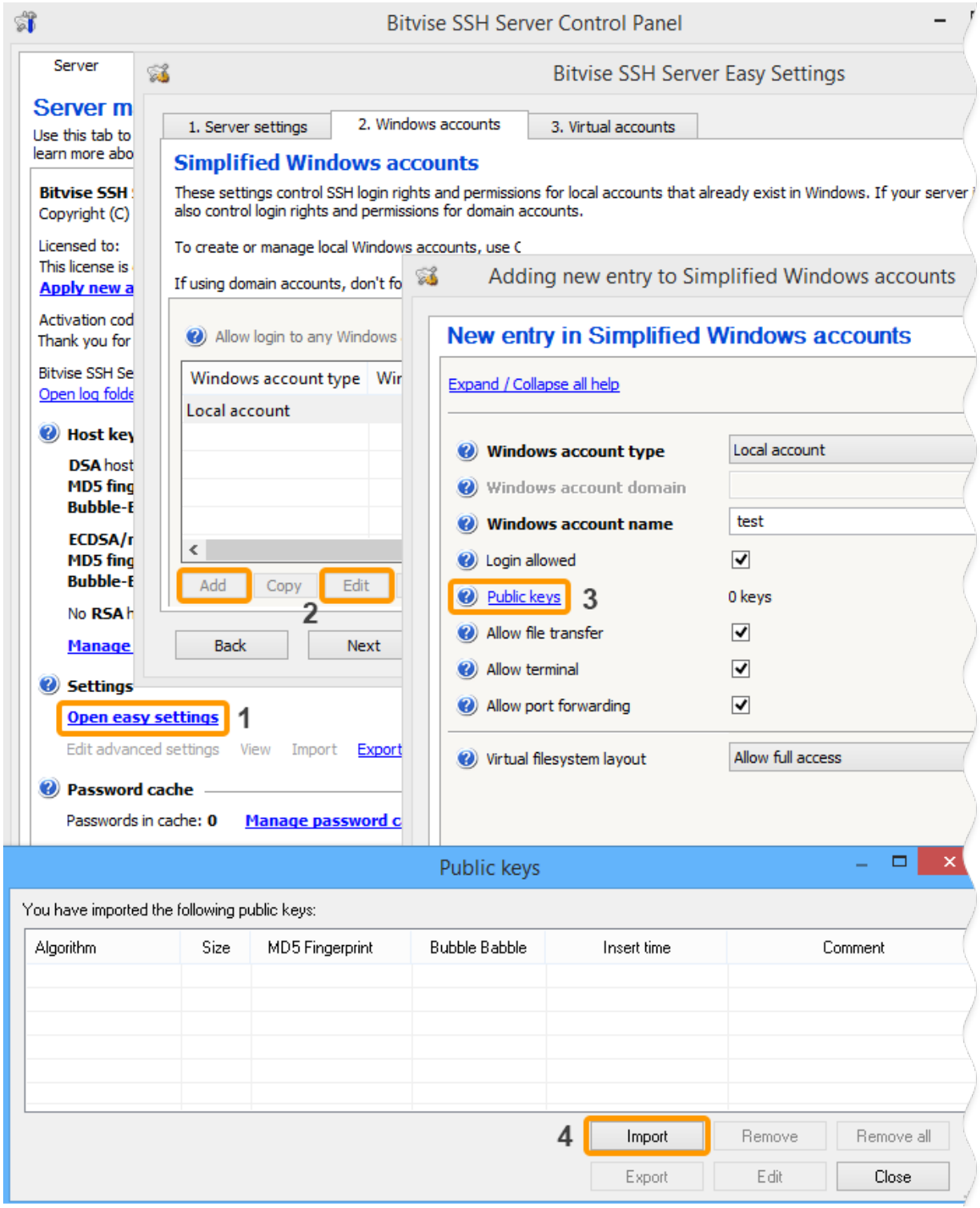
Implementación de claves públicas en Bitvise SSH Server

Si es novato en Bitvise SSH Server (consulte la documentación de Bitvise para detalles de configuración específicos), le recomendamos que primero se asegure de que pueda establecer una conexión SSH en funcionamiento antes de cambiar cualquier configuración en el servidor. Si no puede conectarse al servidor SSH usando su configuración predeterminada, esto es más probable debido a un problema de red o firewall que deberá resolver antes de poder conectarse. En su configuración predeterminada, Bitvise SSH Server acepta conexiones en el número de puerto utilizado a menudo para servidores SSH, 22. Este es el único puerto que debe abrir en su firewall para conectarse al servidor SSH. Si utiliza el reenvío de puerto para tunelizar otras aplicaciones a través de SSH, **no** debe abrir ningún puerto adicional para las conexiones de túnel. Todas las conexiones de túnel se reenvían a través de la sesión SSH, establecida mediante el puerto 22.

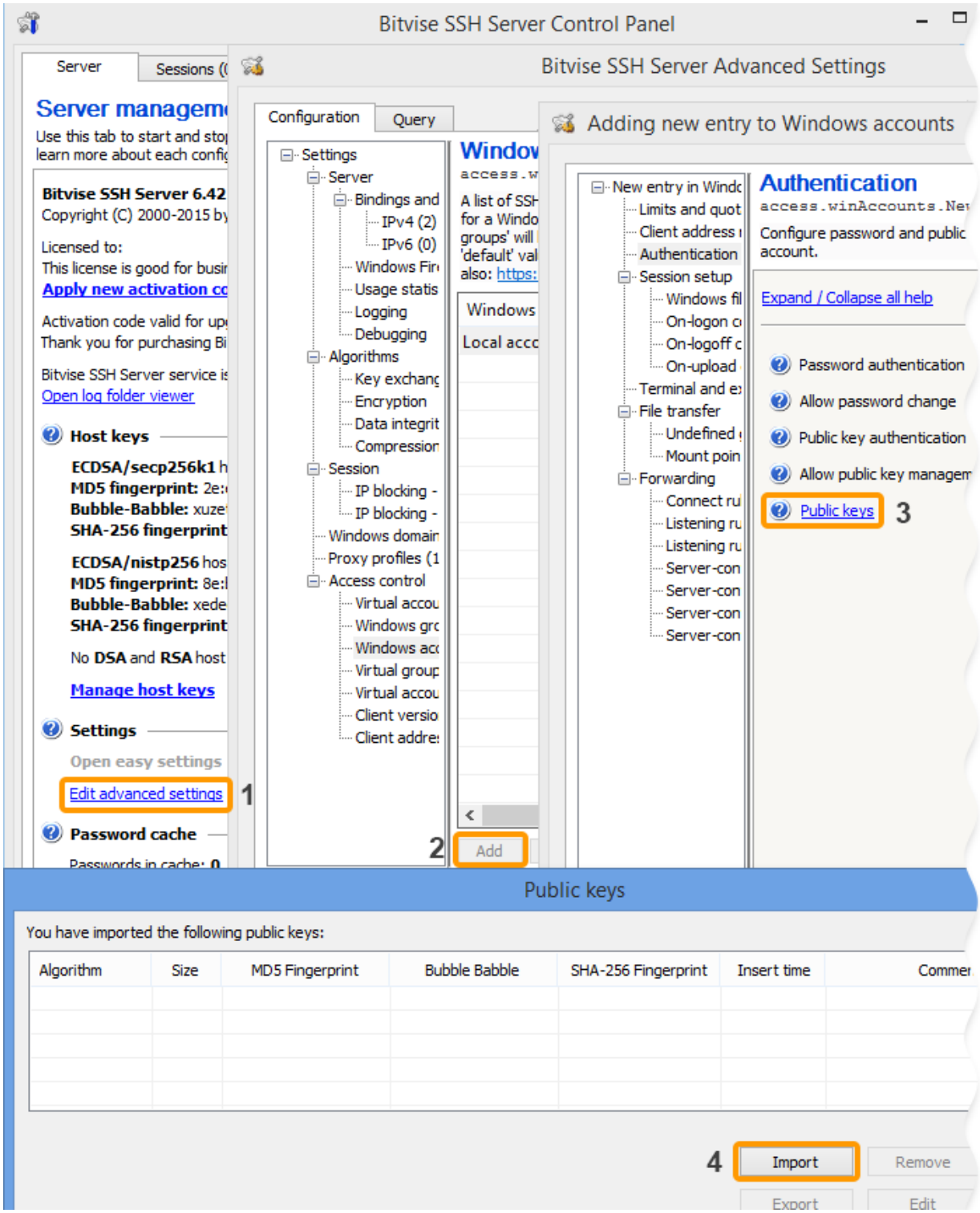
1. Al conectarse a Bitvise SSH Server con un cliente SSH por primera vez, inicie sesión con el nombre de usuario y la contraseña de una cuenta de Windows que existe en la máquina donde se ejecuta el servidor SSH. Para iniciar sesión en una cuenta de dominio de Windows, especifíquela en el formato `domain\account`.

   Puede utilizar cualquier cliente SSH para iniciar sesión en Bitvise SSH Server, siempre y cuando admita el protocolo SSH versión 2.

2. Después de asegurarse de que la clave pública se haya guardado en un archivo, transfiérala a la máquina donde está instalado Bitvise SSH Server o a la máquina desde la que administra el servidor SSH de forma remota utilizando Bitvise SSH Client.
3. Abra el **SSH Server Control Panel** y, a continuación, para importar la clave pública en la configuración de la cuenta del usuario SSH, utilice **Open easy settings**:



o **Edit advanced settings**:



Nota:

Para las cuentas de Windows, Bitvise SSH Server también admite la sincronización con `~/.ssh/authorized_keys`, siempre que esta función esté habilitada en **Advanced SSH Server settings**, bajo **Access control**. Si esta función está habilitada, Bitvise SSH Server comprobará la existencia del archivo `authorized_keys` cuando el usuario cierre la sesión. Si el archivo existe, Bitvise SSH Server reemplazará todas las claves públicas configuradas para el usuario con las claves encontradas en este archivo.

Verificación del acceso al host remoto y el inicio de sesión de WPS

1. Compruebe que el inicio sesión del servidor remoto puede llevarse a cabo, por ejemplo:

```
jsmith@local-host$ ssh jsmith@remote-host
Last login: Wed Oct 21 17:22:33 2015 from 192.168.1.2
[Note: SSH did not ask for password.]

jsmith@remote-host$ [Note: You are on remote-host here]
```

Nota:

Si se le pide una contraseña, se ha producido un error con la autenticación de la clave pública.

2. Si la autenticación de la clave pública se finaliza correctamente, si está utilizando **WPS Link**, cree la conexión con el host requerida y el servidor del host remoto a través de **WPS Workbench**. Si está utilizando **WPS Communicate**, inicie sesión en WPS usando su clave privada, a través de la instrucción `SIGNON`, para lo cual debe especificar la opción de la instrucción **IDENTITYFILE** o la opción del sistema **SSH_IDENTITYFILE**, por ejemplo:

```
SIGNON <servername> SSH
USERNAME="<username>"
IDENTITYFILE="C:\Users\techwriter\.ssh\wpscommunicate.ppk"
LAUNCHCMD="/home/installs/wps-3.2/bin/wps -dmr ";
```

Alternativamente:

```
OPTIONS SSH_IDENTITYFILE="C:\Users\techwriter\.ssh\wpscommunicate.ppk";
SIGNON <servername> SSH
username="<username>"
LAUNCHCMD="/home/installs/wps-3.2/bin/wps -dmr";
```

Nota:

No puede utilizar ni **IDENTITYFILE** ni **SSH_IDENTITYFILE** si está utilizando *Autenticación de la frase de contraseña mediante ssh-agent* (pág. 77).

Autenticación de la frase de contraseña mediante ssh-agent

WPS Communicate no admite la lectura de archivos de claves privadas que se han guardado en forma cifrada con una **frase de contraseña**. Sin embargo, todavía es posible utilizar pares de claves privadas de esta forma si utiliza un agente de llavero. Supondremos que está utilizando el **ssh-agent** de OpenSSH para UNIX/Linux.

El programa **ssh-agent** se ejecuta en el sistema cliente, actuando como un almacén temporal de claves privadas en forma descifrada. Cuando el cliente SSH se autentica con un host remoto, puede acceder a la clave privada del agente sin necesidad de solicitar al usuario.

Al iniciar, **ssh-agent** no tiene ninguna tecla. Éstos se cargan desde el disco mediante el comando **ssh-add**, momento en el que el usuario introduce la frase de contraseña de cada clave para descifrarla. El agente puede almacenar varias teclas simultáneamente, de las cuales el sistema elegirá automáticamente la clave correcta para el servidor remoto.

El siguiente procedimiento supone que ya haya generado un par de claves con una **frase de contraseña** e implementado la clave pública en el servidor SSH remoto (para UNIX/Linux) o en Bitvise SSH Server (para Windows).

Continúe de la manera siguiente:

1. Inicie **ssh-agent** escribiendo lo siguiente en su sesión de terminal local:

   ```
   eval $(ssh-agent)
   ```

   Esto iniciará el programa del agente y lo colocará en segundo plano.

   Nota:

   El comando `eval` le comunica al shell que ejecute la salida de **ssh-agent** como comandos shell. A partir de entonces, los procesos ejecutados por esta shell heredan sus variables de entorno y tienen acceso al agente.

2. Ahora, debe agregar su clave privada al agente, así que pueda administrar su clave:

   ```
   ssh-add
   ```

   Nota:

   Cuando se ejecuta sin argumentos, **ssh-add** agrega automáticamente los archivos `~/.ssh/id_rsa`, `~/.ssh/id_dsa`, `~/.ssh/id_ecdsa`, `~/.ssh/id_ed25519` y `~/.ssh/identity`.

3. Se le solicitará que introduzca su **frase de contraseña**:

   ```
   Enter passphrase for /home/demo/.ssh/id_rsa:
   Identity added: /home/demo/.ssh/id_rsa (/home/demo/.ssh/id_rsa)
   ```

Nota:

Si desea poder conectarse sin una contraseña a un servidor desde el otro servidor, tendrá que reenviar su información de clave SSH. Esto le permitirá autenticar a otro servidor a través del servidor al que está conectado, usando las credenciales en su host local. Para iniciarlo, **ssh-agent** debe estar en ejecución y su clave privada se debe haber agregado al agente (véase arriba).
A continuación, debe conectarse a su primer servidor utilizando la opción `-A`. Esto reenvía sus credenciales al servidor para esta sesión:

```
ssh -A username@remote_host
```

Desde aquí, puede ejecutar SSH en cualquier otro host que su clave SSH está autorizada a acceder. Se conectará como si su clave SSH privada estuviera ubicada en este servidor.

Nota:

WPS detecta automáticamente si se está ejecutando o no **ssh-agent**.

Inicio de sesión único de Kerberos

Puede configurar el inicio de sesión único de Kerberos así que pueda iniciar sesión desde WPS a un servidor remoto sin proporcionar directamente una contraseña o un archivo de identidad de SSH.

Esto normalmente requiere modificaciones en la configuración del demonio SSH para acomodar Kerberos, cuyos detalles están fuera del ámbito de este documento. Deberá discutir las modificaciones necesarias con el administrador del sistema del host remoto, ya que la configuración de la autenticación Kerberos es un trabajo para un administrador de sistema con experiencia.

La configuración Kerberos también se requiere en el equipo cliente.

Antes de intentar realizar un inicio de sesión de Kerberos mediante WPS, debe poder realizar una conexión de Kerberos a un host remoto mediante un cliente SSH convencional.

Una vez que puede realizar un inicio de sesión de Kerberos en un host remoto utilizando un cliente SSH externo, puede realizar el mismo inicio de sesión desde WPS. No es necesario especificar ninguna información de autenticación con la instrucción `SIGNON`, pero debe asegurarse de que no se utilice ni la opción de la instrucción **IDENTITYFILE** ni la opción del sistema **SSH_IDENTITYFILE**.

El ejemplo de código de inicio de sesión podría parecerse a:

```
SIGNON <servername> SSH
LAUNCHCMD="/home/installs/wps-3.2/bin/wps -dmr";
```

Clientes de Linux

El comando `kinit` realizará un inicio de sesión de Kerberos y le solicitará su contraseña de Kerberos. Esta acción adquiere un vale que se almacena en caché en un periodo durante el cual el vale se puede pasar de forma segura a otros hosts como prueba de identidad, permitiendo por tanto la autenticación a esos hosts, sin necesidad de proporcionar información de autenticación adicional:

```
kinit
ssh -o PreferredAuthentications=gssapi-with-mic <hostname>
```

Especificar la opción `PreferredAuthentications=gssapi-with-mic` garantiza que SSH sólo intente la autenticación GSSAPI y, por ejemplo, no intente utilizar la autenticación de clave pública que de otra forma podría llevarse a cabo sin solicitar una contraseña.

Clientes de Windows

Es posible configurar el inicio de sesión único de Kerberos desde WPS en Windows, pero existen algunas restricciones:

- La primera es que **no puede** ser un administrador local en su máquina. Si lo es, Windows no emitirá el vale de concesión de vales de Kerberos necesario para que WPS ejecute la autenticación Kerberos.
- El segundo es que necesita establecer una clave de registro para permitir que Windows da el vale de concesión de vales, incluso si no está en el grupo de administradores locales. El valor de `allowtgtsessionkey` bajo la siguiente clave debe establecerse en 1:

  ```
  HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\Kerberos\Parameters
  ```

  Nota:

  Es posible que necesite agregar este valor al registro si aún no está presente.

  Nota:

  Las restricciones anteriores no se aplican a PuTTY, por lo que si puede iniciar sesión utilizando PuTTY pero no WPS, es probable que una de las restricciones anteriores sea la causa.

Variables de entorno

Es posible que sea necesario configurar variables de entorno para proporcionar el acceso a software de terceros que se usan con una instalación de WPS específica.

El uso de WPS con aplicaciones de terceros, tales como servidores de bases de datos, requiere que la información de entorno esté disponible en el inicio de WPS, tal como:

- `LD_LIBRARY_PATH` o `LIBPATH` en sistemas Linux o Unix que apuntan a bibliotecas de cliente
- `ODBCSYSINI` que apunta a las bibliotecas de cliente unixODBC
- `PATH`, por ejemplo que apunta a las bibliotecas cliente en Windows o los directorios de aplicaciones en sistemas Linux o Unix.

El método recomendado por WPS para establecer las variables de entorno se describe en `install-EN`.

Una vez que el administrador del sistema haya configurado las variables de entorno requeridas, puede probar si una sesión de comunicación está disponible mediante el siguiente programa de lenguaje SAS:

```
options ssh_hostvalidation=none;
%let host=host port;
SIGNON host ssh user="user" password="password"
  launchcmd="installation_path/bin/wps -dmr";

RSUBMIT;
  PROC OPTIONS;
  RUN;
ENDRSUBMIT;

SIGNOFF;
```

Cuando el servidor de destino que hospeda WPS se identifica utilizando la variable de macro `HOST`, y las credenciales de usuario para ese servidor son argumentos para el comando `SIGNON`. La opción de contraseña para el comando `SIGNON` sólo es necesaria si está utilizando la autenticación de contraseña SSH; si utiliza la autenticación basada en claves, solo se requiere un identificador de usuario.

Este ejemplo es para una sesión de comunicación que conecta un servidor Linux; si está utilizando un servidor Windows, la ruta `launchcmd` requerirá la modificación. Cuando se ejecuta, este programa se conecta al servidor identificado, invoca WPS en ese servidor, muestra las opciones establecidas para el servidor de WPS remoto y luego se desconecta del servidor.

Avisos legales

Marcas comerciales

WPS e World Programming son marcas registradas o comerciales de World Programming Limited en la Unión Europea y en otros países. (r) o ® indican una marca comunitaria.

SAS y todos los otros nombres de productos o servicios de SAS Institute Inc. son marcas registradas o comerciales de SAS Institute Inc. en los EE.UU. y en otros países. ® indica la registración en los EE.UU.

Todas las otras marcas comerciales mencionadas pertenecen a sus respectivos propietarios.

Avisos generales

World Programming Limited no está asociada de ninguna manera con SAS Institute Inc.

WPS no es SAS System.

Las expresiones "SAS", "lenguaje SAS" y "lenguaje de SAS" utilizadas en este documento, se usan en referencia al lenguaje de programación, llamado a menudo en una de dichas maneras.

Las expresiones "programa", "programa SAS" y "programa en el lenguaje SAS" utilizadas en este documento, se usan en referencia a los programas escritos en el lenguaje SAS, que también se conocen como "scripts", "scripts SAS" o "scripts en el lenguaje SAS".

Las expresiones "IML", "lenguaje IML", "sintaxis IML", "Interactive Matrix Language" y "lenguaje de IML" utilizadas en este documento, se usan en referencia al lenguaje de programación, llamado a menudo en una de dichas maneras.

WPS incluye software desarrollado por terceros. Se puede encontrar más información en el archivo THANKS o acknowledgments.txt, incluidos en la instalación de WPS.