WPS Hub Installation, Administration and Use Guide

Install, understand, administer and use WPS Hub

Version: 4.1.0

www.worldprogramming.com

Contents

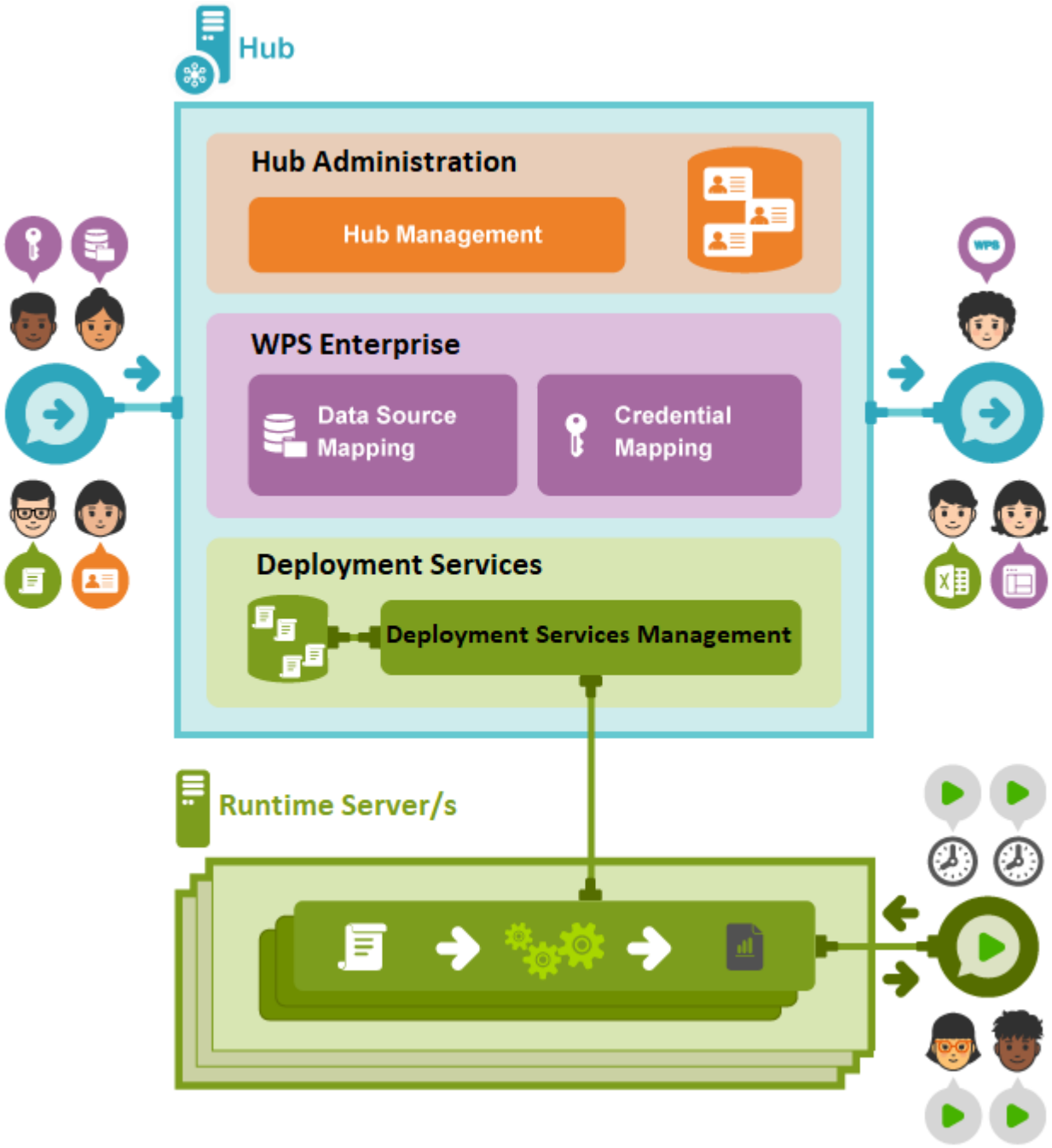
Hub Overview

WPS Hub is an enterprise management tool with two core functions: *WPS Enterprise* and *Deployment Services*. Together with these two core functions, there are numerous *Hub Administration* functions, allowing control over who can access Hub at what level.

The three main areas of WPS Hub are:

- *WPS Enterprise* holds access details and credentials for data sources, removing the need to hard code them into language statements.
- *Deployment Services* permits SAS language and R language programs to be run by external users over HTTP, for example from a web browser, or using a utility such as the WPS Hub MS Excel Plugin.
- *Hub Administration* includes: management of Hub users and their access, management of Deployment Services hosts and host credentials, and a sophisticated access control facility. There are also audit logs available for all Hub functions, and system logs and data for diagnostics and support.

An overview of the functional structure of WPS Hub is shown in the following diagram:



Components of WPS Hub, as shown in the diagram, are as follows:

- *Hub*

  Hub is an enterprise management tool with two core functions: WPS Enterprise and WPS Deployment Services, as well as Hub Administration functions. Hub is available for both Windows and Linux and can be managed either locally or remotely using a web interface known as the *WPS Hub web portal*.

- *Hub Administration*

Hub Administration functions include:

- *Users*

  WPS Hub requires a username and password to grant access to any of its services. These credentials can either be created and stored within Hub for use only with Hub, or they can be imported LDAP users. If the latter option is chosen, 'single sign on' can be implemented. Either type of user can be included in various WPS Hub *user groups*, which allow different levels of access to Hub to be specified.

- *Hub Management*

  The WPS Hub web portal can be used to manage all aspects of Hub, including access control, the definition of Deployment Services hosts and their access credentials, as well as access to an audit log of all Hub activity.

- *WPS Enterprise*

  WPS Enterprise provides centralised definitions for data sources, which are then accessible from Workbench. *Data source mapping* allows libraries to be defined centrally within Hub and then referenced directly from Workbench (replacing Libname statements with their hardcoded library definitions). *Credential mapping* stores access credentials for those data sources.

- *Deployment Services*

  WPS Deployment Services enables SAS language and R language programs to be run by external users over HTTP. WPS Hub stores these *programs* either in its own database, or via a link to a Git repository (referred to in Hub as a *program package*). Programs can then be deployed into *environments*, which are conceptual containers for programs that can be accessed externally, with optional authentication and security restrictions for each environment if required. Each environment contains *runtime servers* to run these deployed programs using WPS processing *engines*.

- *Deployment Services* Management

  Deployment Services allows large numbers of remote users to run SAS language and R language programs over HTTP.

  Deployment Services Management within Hub includes:

  - *Programs*

    Programs can be created, stored and managed using Hub Deployment Services. Alternatively, for more complex programming implementations, such as those with multiple source files or multiple resources, *program packages* can be defined. Hub uses the Git version control system to manage program packages, either by maintaining its own internal Hub Git repository, known as an *internal program package*, or by referencing an external Git repository, known as an *external program package*.

  - *Deployment Management*

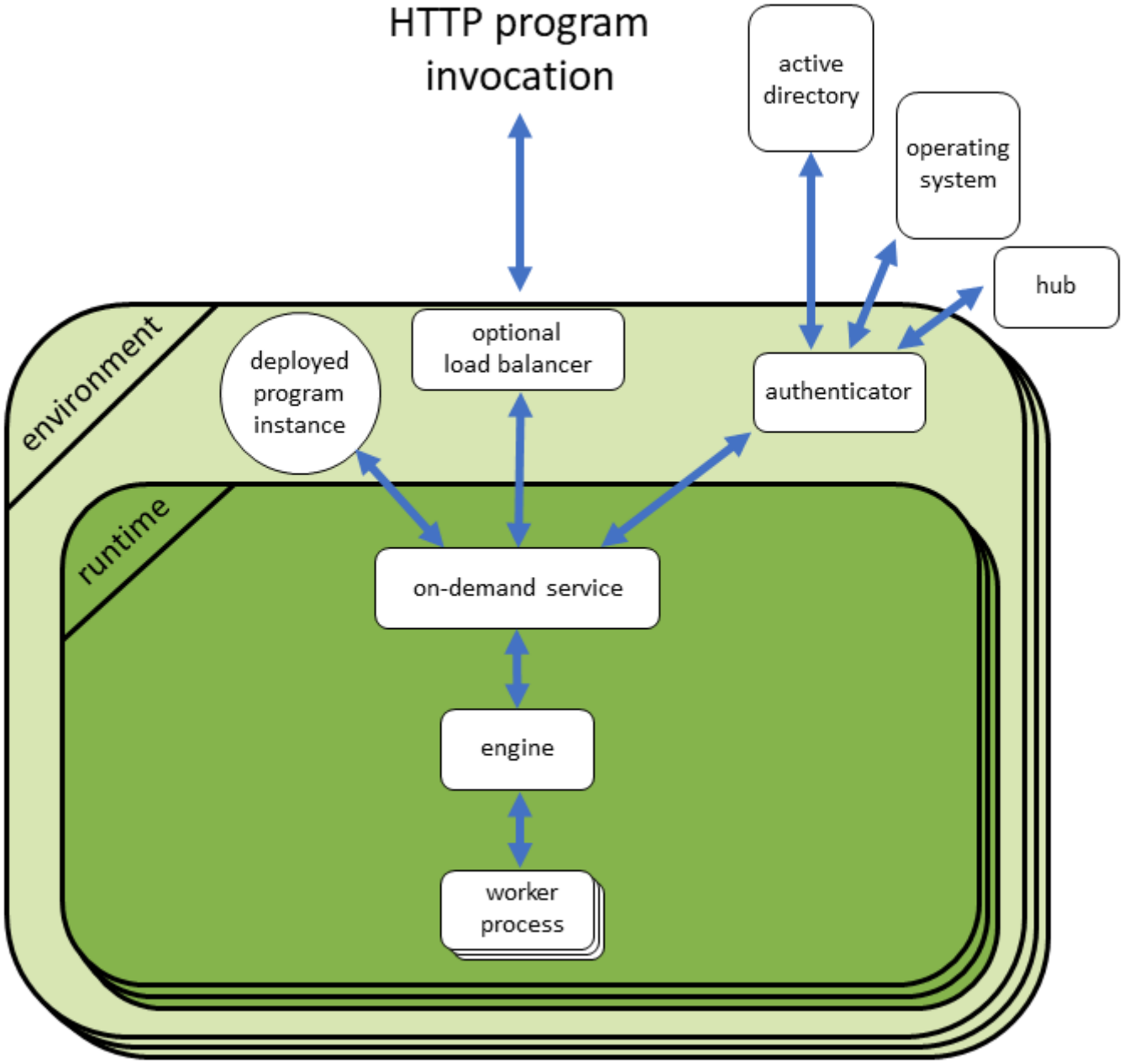
    For Deployment Services programs to be run externally, they need to be deployed into *environments*. To faciliate this, a number of Deployment Services entities need to be set up, as described below. Deployment management functions within Hub allow these Deployment Services entities to be created and managed.

- *Runtime Hosts*, *engines* and *worker processes*

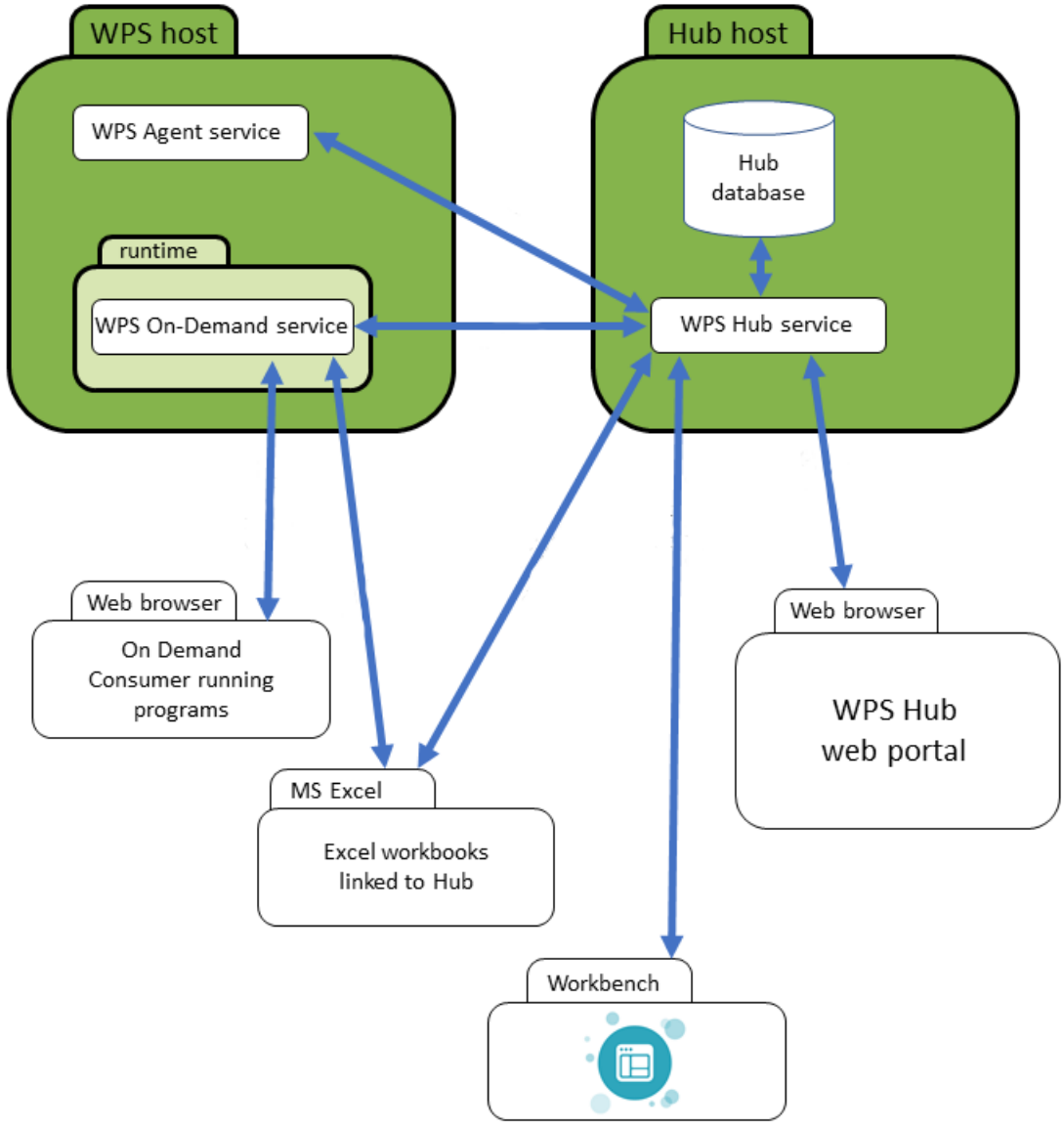
  Runtime hosts are the machines (physical or virtual) within environments that host *runtimes*, which in turn run Deployment Services programs. Conceptually, runtimes exist within environments (not pictured), which can also be defined from within Hub. Runtimes contain one or more *engines*, which use *worker processes* to run programs from the Hub directory when requested.

  Runtimes can be called either directly by external users, or via an externally provided load balancer. In either case, Deployment Services consumers will access programs over HTTP via a URL and will be unaware of the mechanism by which the deployed program is being run.

The concepts of Deployment Services are shown in the following diagram.



The architecture and connectivity of a typical WPS Hub installation are shown in the following diagram:



As can be seen in the diagram, Hub is based around two installations, which may be on the same or different hosts. The Hub host contains a Hub database or a link to a database which stores the data required by Hub: programs (or links to Git repositories of programs), users, and other data. The WPS Host is used by Deployment Services and will run SAS language and R language programs when requested via HTTP/HTTPS. All Hub operations are marshalled by services installed on these hosts.

Installation and configuration

This section describes how to install and configure WPS Hub.

Note:

If you are upgrading or reinstalling Hub, skip straight to the section *Upgrading or reinstalling WPS Hub* (page 38).

To install and configure Hub:

1. Download WPS Hub from the WPL Website.
2. Install WPS Hub.
3. Start the WPS Hub service.
4. Connect to WPS Hub using SSH.
5. Apply the WPS Hub licence key.
6. Configure the WPS Hub database.

Follow the sections in this guide to complete the above steps.

Installation prerequisites

Before installing WPS Hub:

- Ensure that you have administrator privileges on the host.
- If you want to use HTTPS to secure communications with the WPS Hub server (optional, but strongly recommended for production systems), you will need to create or obtain an internally or externally signed digital certificate (see *Configuring WPS Hub to use HTTPS* (page 25)).
- If you want to use LDAP to import users and groups from Active Directory, you need to know the LDAP configuration settings for your system (see *LDAP import of users and groups* (page 30)). If you do not know these settings, contact your IT department.
- If you want to use domain usernames and passwords for logging in to WPS Hub (single sign on) for a Linux client, you might need to install some additional packages on the WPS Hub server and also on the web browsers (see *Single sign on* (page 33)).
- If you plan to use any other installation configuration, contact WPL support for advice.

System requirements

WPS Hub can only run on certain systems and can only be accessed using particular operating systems.

WPS Hub server

To install and configure WPS Hub, you will require one of the following operating systems:

- Windows 8 or later, or Windows Server 2012 or later.
- Linux: The latest RHEL or Debian version of Linux is recommended, but WPS Hub also works with any RPM or Debian-based Linux distribution.

Databases

WPS Hub requires a database to operate. This database can either the default internal database supplied with WPS Hub, or you can use one of four supported external databases.

Supported DBs are:

- MariaDB, version 10 or later
- MySQL, version 8 or later
- PostgreSQL, version 10 or later

Web clients

To access the WPS web portal, you need one of the following supported browsers:

- Windows clients:
  - Google Chrome version 60.0.3112 or later
  - Mozilla Firefox version 58.0.1 or later
- Linux Clients:
  - Google Chrome version 60.0.3112 or later
  - Mozilla Firefox version 58.0.1 or later

Downloading and installing WPS Hub

WPS Hub can be downloaded as an installation package from the WPL website. Packages are available for both Windows and Linux. WPS Hub is a self contained installation that does not require a WPS installation on the host computer.

Downloading WPS Hub

WPS Hub installation packages for supported operating systems can be downloaded from the World Programming website.

Before you can download WPS Hub, WPL must provide access to the WPS Hub tab in the **Downloads** area of the WPL website. To request access, contact WPL customer support (`support@worldprogramming.com`).

To download the WPS Hub installation package:

1. Go to https://www.worldprogramming.com/support/downloads. Log in to your account and select **Downloads** from the top of the screen. By default, you see the WPS Analytics product downloads. Click **WPS Hub** to view your WPS Hub downloads. If you do not see the WPS Hub tab, contact WPL Support to request access.
2. Choose the appropriate download package platform and click **Download**:
   - `MSI` package for Windows.
   - `DEB` package for Debian-based Linux.
   - `RPM` package for Red Hat-based Linux systems (for example, RHEL, CentOS, Fedora).
     - For installations that use the System V service framework, choose the RPM package *with* RHEL6 in the filename.
     - For installations that use the newer Systemd service framework, choose the RPM package *without* RHEL6 in the filename.

Installing WPS Hub for Windows

WPS Hub for Windows is installed using a click through wizard, invoked from an MSI installation package.

To install WPS Hub you need access to an account with administrator privileges.

To install WPS Hub on Windows:

1. Either log in as an administrator to the computer that you want to install WPS Hub on, or proceed to the next step and run the MSI package as an administrator.

2. Run the downloaded MSI file to start the installer.
3. Click **Next** to start the installation process.
4. Accept the end-user licence agreement (you cannot install WPS Hub if you do not do this) then follow the on-screen instructions to complete the installation.

WPS Hub has now been installed. Now apply your WPS Hub license key (see *Applying the WPS Hub Licence Key* (page 17)).

Windows installation locations

On Windows installations, files are installed in these locations:

Location	Contents
`C:\ProgramData\World Programming\WPS Hub \4\etc`	Application configuration files,
`C:\ProgramData\World Programming\WPS Hub \4\data`	Application data files (Hub database, log files etc).

Installing WPS Hub for Linux

The installation instructions are slightly different for RPM-based Linux installations and Debian-based Linux installations. Follow the instructions that apply to your installation.

RPM-based Linux installation

WPS Hub for RPM based Linux is installed from the command line using the `install` command.

To install WPS Hub, you need access to an account with administrator privileges.

To install WPS Hub on RPM-based Linux (for example, RHEL, CentOS, Fedora):

1. From a terminal on the installation location, run one of the following commands, depending on the package manager supported by your Linux installation:

   ```
   sudo dnf install <package_name>.rpm
   ```

   or

   ```
   sudo yum install <package_name>.rpm
   ```

2. At the `Is this ok` prompt, type `y` and press Enter.

WPS Hub installs. Now apply your WPS Hub license key (see *Applying the WPS Hub Licence Key* (page 17))..

Debian-based Linux installation

WPS Hub for Debian based Linux is installed from the command line using the `dpkg` command.

To install WPS Hub, you need access to an account with administrator privileges.

To install WPS Hub on Debian-based Linux:

1. From a terminal, enter:

```
sudo dpkg -i <package_name>.deb
```

2. At the `Is this ok` prompt, type `y` and press enter.

WPS Hub installs. Now apply your WPS Hub license key (see *Applying the WPS Hub Licence Key* (page 17)).

Linux installation locations

On Linux installations, files are installed in these locations:

Location	Contents
`/var/worldprogramming/wpshub-4/etc`	Application configuration files.
`/var/worldprogramming/wpshub-4/data`	Data files (Hub database, log files etc).
`/opt/worldprogramming/wpshub-4`	Main executables and libraries.

WPS Hub Services

A WPS Hub service (called `wpshub`) supports the Hub installation. At this stage, directly after running the installation package, this service needs to be started..

Starting services

To start a service:

1. Take the following action, depending on your operating system:

- Windows: open **Services**, then locate the service you wish to start (**WPS Hub**, **WPS On Demand Runtime**, or **WPS Host Agent**)and click the **start** icon ( ).
- For Linux where **Systemd** is supported, log in to the server where WPS Hub is installed and run the following command:

  ```
  sudo systemctl start service
  ```

  where *service* is *wpshub*, *wpsondmd*, or *wpsagent*.
- For Linux where **Systemd** is not supported, log in to the server where WPS Hub is installed and run the following command:

  ```
  sudo service wps-service start
  ```

  where *wps-service* is *wpshub*, *wpsondmd*, or *wpsagent*.

Stopping services

To stop a services:

1. Take the following action, depending on your operating system:
   - Windows: open **Services**, then locate the service you wish to stop (**WPS Hub**, **WPS Deployment Services**, or **WPS Agent**) and click the stop icon ( ).
   - For Linux where **Systemd** is supported, log in to the server where WPS Hub is installed and run the following command:

     ```
     sudo systemctl stop service
     ```

     where *service* is *wpshub*, *wpsondmd*, or *wpsagent*.
   - For Linux where **Systemd** is not supported, log in to the server where WPS Hub is installed and run the following command:

     ```
     sudo service wps-service-service stop
     ```

     where *wps-service* is *wpshub*, *wpsondmd*, or *wpsagent*.

Restarting services

To restart aservices:

1. Take the following action, depending on your operating system:
   - Windows: open **Services**, then locate the service you wish to restart (**WPS Hub**, **WPS Deployment Services**, or **WPS Agent**)and click the restart icon ( ).
   - For Linux where **Systemd** is supported, log in to the server where WPS Hub is installed and run the following command:run the following command:

     ```
     sudo systemctl restart service
     ```

where *service* is *wpshub*, *wpsondmd*, or *wpsagent*.

- For Linux where **Systemd** is not supported, log in to the server where WPS Hub is installed and run the following command: run the following command:

  ```
  sudo service wps-service restart
  ```

  where *wps-service* is *wpshub*, *wpsondmd*, or *wpsagent*.

Connecting to the Hub shell using SSH

Administration of the installation and configuration of Hub is performed using Hub's own command line interface (CLI) shell, accessed using Secure Shell (SSH).

Hub installations use port 8101 for SSH by default. This can be changed in the configuration file `org.apache.karaf.shell.cfg`, which is located as follows:

- Linux: `/var/worldprogramming/wpshub-4/etc/`
- Windows: `C:\ProgramData\World Programming\WPS Hub\4\etc`

Connecting to Hub using SSH from Linux

SSH can be used on Linux to connect to a Hub installation using Hub's own command line interface.

To start an SSH session from the Linux command line, type:

```
ssh admin@HubHost -p 8101
```

where *HubHost* is the name of the Hub host.

If you are using SSH from the same installation as Hub is installed on, then either use `localhost` as the *HubHost* in the above command, or run `/opt/worldprogramming/wpshub-4/bin/client`.

Connecting to Hub using SSH from Windows

SSH can be used on Windows to connect to a Hub installation using Hub's own command line interface.

A standard SSH client for Windows can be used to connect to a Hub installation's CLI. To connect, you will require the domain name of the Hub installation location and the SSH port that Hub is configured to use.

If required, the Hub CLI can be started locally. Ensure that the WPS Hub service is started and running, click **Start**, click **World Programming WPS Hub 4** and then click **WPS Hub standard shell**.

Applying the WPS Hub Licence Key

Before WPS Hub can be used, a WPS Hub Licence Key must be applied to the installation.

Before you begin, obtain a WPS Hub licence key.

To apply the WPS licence key:

1. Connect to WPS Hub using SSH (see *Connecting to the Hub shell using SSH* (page 16)).
2. To apply the licence key, type `wpshub:applylicence "licence key"`, where *licence key* is your licence key text.

   `Licence key successfully applied` is displayed.
3. To confirm that WPS Hub is licensed, type `wpshub:licenceinfo`.

   WPS Hub Licence Info is displayed, with an **Expiry** date in the future.

Now configure the WPS Hub database (see *Configuring the WPS Hub database* (page 21)).

Installing WPS Deployment Services

WPS Deployment Services allows large numbers of remote users to run SAS language and R language programs over HTTP. WPS Deployment Services is included in the latest WPS download, although if you want to use it then it must be installed separately following installation of WPS Hub.

Installing and starting WPS Deployment Services for Linux

This installation task requires a Linux server running WPS at verson 4.1.

To install and start Deployment Services for Linux:

1. Log in to a Linux server with WPS version 4.1 installed.
2. Install the WPS On Demand service by typing:

   ```
   sudo /opt/worldprogramming/wps-4/bin/wpsondmd --install -p port
   ```

   where *port* is a unique port that the service will use to communicate. This port number will be written to the *Deployment Services Configuration File* (page 19).

   A ‘service installed’ message indicates success.
3. Start the WPS On Demand service:

- **Linux where Systemd is active**: Enter: `sudo systemctl start wpsondmd` and press **Return**.
- **Linux installations where Systemd is not active**: Enter: `sudo service wpsondmd-service start` and press **Return**.

4. Now install the WPS Agent service by typing:

```
sudo /opt/worldprogramming/wps-4/bin/wpsagent --install -p port
```

where *port* is a unique port that the service will use to communicate.

A 'service installed' message indicates success.

5. Start the WPS Agent service:

- **Linux where Systemd is active**: Enter: `sudo systemctl start wpsagent` and press **Return**.
- **Linux installations where Systemd is not active**: Enter: `sudo service wpsagent-service start` and press **Return**.

If you want to configure more advanced options, then see *Deployment Services Configuration File* (page 19)

Installing and starting WPS Deployment Services for Windows

This installation task requires a Windows installation running WPS at verson 4.1 or higher.

To install and start Deployment Services for Windows:

1. On the Windows installation that you have just installed WPS onto (at least version 4.1xxx), open a command prompt with admnistrative permissions.
2. Run the following command:

```
"C:\Program Files\World Programming\WPS\4\bin\wpsondmd.exe" --install -p port
```

where *port* is a unique port that the service will use to communicate. This port number will be written to the *Deployment Services Configuration File* (page 19), which is used by Deployment Services to determine the Deployment Services service's running parameters.

A `Service WPS On Demand Runtime installed successfully` message is displayed.

3. Open **Services**, locate **WPS On Demand Runtime** and confirm that the status is **Running**. If necessary, start the service by clicking the **start** icon ( ).
4. Run the following command:

```
"C:\Program Files\World Programming\WPS\4\bin\wpsagent.exe" --install -p port
```

where *port* is a unique port that the service will use to communicate.

A `Service WPS Host Agent installed successfully` message is displayed.

5. Open **Services**, locate **WPS Host Agent** and confirm that the status is **Running**. If necessary, start the service by clicking the **start** icon ( ).

If you want to configure more advanced options, then see *Deployment Services Configuration File* (page 19)

Deployment Services Configuration File

When Deployment Services is installed, a configuration file is created. The configuration file defines a variety of parameters used by Deployment Services, and can be edited at any time to change these parameters. If the file is edited, the Deployment Services service needs to be restarted for the changes to take effect.

Configuration File Location

The configuration file locations are:

- Linux: `/var/worldprogramming/wps-4/wpsondmd/config.json`
- Windows: `C:\Program Files\World Programming\WPS\4\wpsondmd\config.json`

.

Configuration File Contents

By default, the configuration file will show the one port configured during installation of Deployment Services. More ports can be added if required, using the following syntax:

{ "threadCount": *<number of threads>*, "listenPort":

[

{ "port": *<port number>* },

{ "port": *<port number>*, "secure": *<secure: true | false>* }

],

"configdb": "*<config database filepath>*",

"sslCertificate": "*<sslCertificate filepath>*",

"sslPrivateKey": "*sslPrivateKey filepath*",

}

Where:

- *number of threads*: The number of threads decoding HTTP requests.

- *port number*: The port/s to configure for Deployment Services. Remember that if you change these on the host, the host definition in the WPS Hub web portal will also need to be changed.
- *secure*: true or false: defines whether the specified port is secure or not.
- *configdb*: location of the WPS Hub database.
- *sslCertificate*: location of the SSL Certificate, if applicable.
- *sslPrivateKey*: location of the SSL Private Key, if applicable.

Configuration File Example

Below is an example configuration file:

{ "threadCount": 10, "listenPort":

[

{ "port": 5555 },

{ "port": 5556, "secure": true }

],

"configdb": "/var/worldprogramming/wps-4/wpsondmd/config.db",

"sslCertificate": "/var/worldprogramming/wps-4/wpsondmd/cert.pem",

"sslPrivateKey": "/var/worldprogramming/wps-4/wpsondmd/key.pem",

}

Installing WPS agent

WPS agent is an optional component that can be used to monitor the state of the host. WPS agent is included in the WPS download, although if you want to use it then it must be installed separately following an installation of WPS.

Installing and starting WPS agent for Linux

To install and start WPS agent:

1. On the Linux server that you have just installed WPS onto (at least version 4.1xxx), run the following command:

   ```
   sudo /opt/worldprogramming/wps-4/bin/wpsagent --install -p port
   ```

   where *port* is the port that WPS agent will use to communicate. This port must be different from that used by Hub and Deployment Services.

A ‘service installed’ message indicates success.

2. Start the WPS agent service:

```
sudo systemctl start wpsagent
```

Configuring the WPS Hub database

WPS Hub requires a database to operate. This database can either the default internal database supplied with WPS Hub, or you can use one of four supported external databases.

You use the WPS Hub Shell to create and configure the WPS Hub database for a new installation of WPS Hub. For the database you can either use the internal WPS Hub database engine, or a supported external database. You can change the database provider later if required.

Supported external databases are:

- MariaDB, version 10 or later
- MySQL, version 8 or later
- PostgreSQL, version 10 or later

To configure WPS Hub using the default internal database, see *Configuring WPS Hub with the internal database* (page 21). To configure WPS Hub initially using another database provider see *Configuring WPS Hub with another database* (page 23).

If you are upgrading or reinstalling WPS Hub and want to use an existing WPS Hub database, see *Upgrading or reinstalling WPS Hub* (page 38).

Configuring WPS Hub with the internal database

WPS Hub is supplied with its own internal database, which can be configured as the database WPS Hub uses to operate.

Ensure that the WPS Hub licence is installed before configuring WPS Hub.

To configure WPS Hub using the default internal database:

1. Connect to WPS Hub using SSH (see *Connecting to the Hub shell using SSH* (page 16)).
2. Enter `wpshub:db-config-internal`, followed by Return.

   If there is an existing database configured, the filepath to this will be shown, along with details of scheduled backups and a current backup file, if either are present.
3. At the **Database file path** prompt, press **Return** to accept the default suggested path, or enter another path and press **Return**.

4. At the **Schedule backup?** prompt, type **yes** followed by return to configure a scheduled backup, or **no** followed by **return** to proceed without. If you are configuring a scheduled backup, skip to the next section () before continuing with step *5* (page 22) below.
5. Continue bit
6. Follow the prompts to create the WPS Hub database, schedule a backup if required and to define the database tables required by WPS Hub, until you reach the `Bootstrap now?` prompt.

   By default, the database is created as `wpshub.h2.db` in the WPS Hub installation folder. You can accept this location or specify another location. The location can be changed later if necessary.
7. When you reach the `'Bootstrap now?'` prompt, enter `yes` and follow the prompts to create a WPS Hub administrator user with a username and password of your choice. This user is created in the predefined WPS Hub group, HubAdministrators, and exists only within WPS Hub.
8. If you used the installation default password for SSH access:
   a. Enter a new shell admin password at the prompt, followed by return.
   b. Enter the password again to confirm, followed by Return.

Now start the Hub service (see *Starting services* (page 14)).

Scheduling backup of the WPS Hub database during configuration

Before you begin, complete steps *1* (page 21) to *4* (page 22) in *Configuring WPS Hub with the internal database* (page 21)

1. At the **Backup archive file path** prompt, either accept the quoted path by pressing **Return** or specify a new file path.
2. At the **Advanced scheduling?** prompt, type either no or yes, followed by **Return**, and then set the scheduling as follows:
   - If you type **No**: Specify a **Backup Interval** by typing **daily** or **weekly** followed by **Return**, and then specify a time by typing the **hour**, followed by **Return**, and then the **minute**, followed by **Return**.
   - If you typed **Yes**: Specify a cron expression to define the scheduling, using Quartx syntax, followed by **Return**.

Now return to step *5* (page 22) in *Configuring WPS Hub with the internal database* (page 21).

Configuring WPS Hub with another database

WPS Hub requires a database to function. WPS Hub is supplied with its own internal database, although a pre-existing external database can also be used.

Before you begin, you will require the network path to an existing database, together with access credentials. You must also ensure the WPS Hub licence is installed before configuring WPS Hub.

To configure WPS Hub to use an external database instead of the default internal database:

1. Connect to WPS Hub using SSH (see *Connecting to the Hub shell using SSH* (page 16)).
2. Type `wpshub:db-configure-database`, where *database* is one of: *mariadb* (used for both Mariadb and MySQL) or *postgresql* (*internal* is also an option; see the dedicated section *Configuring WPS Hub with the internal database* (page 21)).

   A command wizard is displayed.
3. Follow the steps in the wizard to configure WPS Hub with the chosen database.

   If required, the wizard steps enable you to migrate data from an existing database to the new one, or to reinitialise the database.

Now start the Hub service (see *Starting services* (page 14)).

Testing WPS Hub

Once WPS Hub has been downloaded, installed and its database configured, it can be tested to verify that the WPS Hub service is running correctly and that the WPS Hub administrator user can log in successfully.

If you are using HTTPS, this will need configuring before testing the WPS Hub web portal. To configure HTTPS, see *Configuring WPS Hub to use HTTPS* (page 25).

Testing the configuration

The WPS Hub configuration can be tested to verify that the WPS Hub service is running correctly and that the WPS Hub administrator user can log in successfully.

To test the configuration so far:

1. Connect to WPS Hub using SSH (see *Connecting to the Hub shell using SSH* (page 16))
2. Enter the shell command `wpshub:version`.

   The version of hub that you have installed is displayed. This means that the WPS Hub service is running, and that the initial setup process was successful.

Testing the WPS Hub web portal

The WPS Hub web portal can be tested locally on the Hub host and non-locally on another machine.

If you are configuring HTTPS, do this before you test access to the WPS Hub web portal.

To test access to the WPS Hub web portal:

1. If you have a web browser on the WPS Hub host (the machine where WPS Hub is installed), you can test the WPS Hub web portal locally:
   a. On the WPS Hub host, browse to `http://localhost:8181`. The WPS Hub web portal login screen is shown.
   b. In the WPS Web portal screen, enter the username and password you specified for WPS Hub administrator user you created earlier, then click **Login**.

      Your browser shows the home page for the WPS Hub web portal. This contains tiles for the main areas of functionality, credential management and administration.
2. Now test the non-local access to the WPS Hub web portal:
   a. On a client machine, open a web browser and browse to `https://`*`wpshub-host`*`:8181`, where *wpshub-host* is the name of the machine where WPS Hub is installed. You might need to enable access to port 8181 in the WPS Hub server's firewall. If you need help doing this, consult your IT department.

      The WPS Hub web portal log in screen is shown.
   b. In the WPS Web portal login screen, enter the **username** and **password** for the WPS Hub administrator user you created earlier, then click **Login**.

      Your browser shows the home page for the WPS Hub web portal. This contains tiles for the main areas of functionality, credential management and administration.

      Note:
      It is strongly recommended that HTTPS is enabled for all production systems.

Optional configuration

Before using WPS Hub in your organisation, you can complete some of the following optional configuration steps:

- Enable HTTPS to secure communications with WPS Hub (optional, but strongly recommended for production deployments, see *Configuring WPS Hub to use HTTPS* (page 25)).
- Use LDAP to import users and groups into WPS Hub from Active Directory (see *LDAP import of users and groups* (page 30)).

- Enable single sign on so that authorised users are automatically authenticated to WPS Hub (see *Configuring single sign on* (page 34)).

Configuring WPS Hub to use HTTPS

By default, the WPS Hub server is configured to use HTTP, which is not secure. In a production system, it is strongly recommended that you enable HTTPS, which encrypts the communications between WPS Hub server and the client browsers.

To configure HTTPS, you need a signed digital certificate, which can be:

- Purchased from an external certificate authority, for public use.
- Generated internally, for secure use within the issuing organisation.

We recommend a digital certificate with industry standard encryption (for example, an RSA key of at least 2048 bits, or an elliptic-curve key of at least 256 bits).

If you do not know which kind of certificate to use, or whether a suitable certificate already exists, consult your IT department.

For test systems, you can also use a self-signed certificate. In that case, the WPS Hub portal displays a browser security warning because the certificate you are using is not verified using a formal chain of trust.

WPS Hub certificates

WPS Hub uses a standard Java keystore to store WPS Hub security certificates. Depending on your requirements, you can use one of the WPS Hub shell command wizards to create a security certificate, store it in the WPS Hub keystore and configure HTTPS to use that certificate, or you can install your certificate directly in your own keystore, then tell WPS Hub to use that keystore.

WPS Hub provides the following shell commands to install security certificates and configure HTTPS:

- `wpshub:self-cert` creates a new self-certified certificate and stores it in the WPS Hub keystore. Optionally, you can also generate a certificate signing request (CSR) which you can pass to your IT department who can sign it internally, or get it signed externally, as required.
- `wpshub:cert-reply` imports a certificate reply containing a certificate signed in response to a CSR into the WPS Hub keystore. This formally-signed certificate replaces the original self-signed certificate.
- `wpshub:https-pemsignedcert` adds a new internally or externally signed certificate in PEM format to the WPS Hub keystore.
- `wpshub:https-pcks12signedcert` adds a new internally or externally signed certificate in PCKS12 format to the WPS Hub keystore.

- If you have a certificate in a different format, or want more control over the process, you can use the standard Java keytool utility directly to create a keystore in Java Key Store format and add the certificate to it. In this case, `wpshub:https-keystore` enables you to specify the keystore to be used by WPS Hub. If required, `wpshub:https-useentry` enables you to specify the keystore entry to use.

Pre-requisites

Before configuring WPS Hub to use HTTPS, ensure that:

- The WPS Hub service is running.
- You know a username and password for a WPS Hub user in the Hub Administrators group.
- You know whether you should create a new certificate, or use an existing one.
- If you are creating a new certificate that is to be formally signed, you know the values to supply for the various certificate fields (see *Creating and installing a self-signed certificate* (page 26)).
- If you are using an existing certificate in PCKS12 format, you know the password.

Creating and installing a self-signed certificate

To create and install a self-signed certificate:

Note:
Do not use this method for a production system. It is only suitable for a test system.

1. Connect to WPS Hub using SSH (see *Connecting to the Hub shell using SSH* (page 16)).
2. At the prompts, enter a username and password for a WPS Hub user in the **Hub Administrators** group. The WPS Hub shell starts and displays a command prompt.
3. At the command prompt, enter `wpshub:https-selfcert`. Follow the prompts to create a self-signed certificate and install it:
   a. For **Machine host name**, enter the `WPS Hub server domain name` (for an internal deployment, for example, you might enter `wpshub-server.mydomain.local`).
   b. For a self-signed certificate, you can accept the default values of “Unknown” for the **Organisational unit name**, **Organisation name**, **City or Locality**, **State or Province** and **Two letter country code**.
   c. At the **Certificate validity** prompt, enter the period in days that you want the certificate to be valid for.
   d. At the **HTTPS port number** prompt, either accept the default value of 8443, or specify the port number you want to use, if different.

A message is displayed confirming that the HTTP protocol is no longer enabled, and that HTTPS has been enabled. A self-signed certificate has been created and placed in the WPS Hub keystore.

4. At the **Generate a Certificate Signing Request for this certificate?** prompt enter `No`.
5. Test the HTTPS configuration as described in *Testing the HTTPS configuration* (page 29).

Creating and installing an internally or externally signed certificate

To create a new internally or externally signed certificate and install it:

1. Connect to WPS Hub using SSH (see *Connecting to the Hub shell using SSH* (page 16)).
2. At the shell prompt, enter `wpshub:https-selfcert`. Follow the prompts to create a certificate and generate a CSR.
   a. For the WPS Hub server domain name, enter the `Machine host name` (for example, for an internal deployment, you might enter `wpshub-server.mydomain.local`, or for a deployment with a registered address, you might enter `www.mycompany.com`). If you do not know the WPS Hub server domain name, contact your IT department.
   b. Enter the values for **Organisational unit name**, **Organisation name**, **City or locality**, **State or province** and **Two letter country code**. If you are unsure of the values to enter here, consult your IT department.
   c. At the **Certificate validity** prompt, enter the period in days that you want the certificate to be valid for.
   d. At the **HTTPS port number prompt**, either accept the default value of 8443, or specify the port number you want to use, if different.

   A message is shown confirming that the HTTP address is no longer enabled, and that HTTPS has been enabled. A self-signed certificate has been created and placed in the WPS Hub keystore.
3. At the **Generate a Certificate Signing Request for this certificate?** prompt enter `Yes` to create a certificate signing request (CSR).
4. At the **Certificate signing request output file** prompt, specify the folder and file name for the certificate signing request output file. This folder must already exist, and must be writable by both the WPS Hub process and the user logged into the WPS Hub server that started the WPS Hub shell.
   - Windows: you could use a folder at the top level of the C:\ drive that is neither a system folder nor `C:\Users`.
   - Linux: you could use `/tmp`.

   A CSR is created in the specified location, and the `wpshub:https-selfcert` command exits.
5. Locate the CSR and pass it to your IT department who can sign it internally, or arrange for it to be signed externally.

Note:

Do not run `wpshub:https-selfcert` again while you are waiting for the certificate reply file, as this creates a new self-signed certificate with a new private key in the keystore and overwrites the original certificate and key. In that case, when you receive the certificate reply file, the public key associated with it will no longer match the private key in the WPS Hub keystore and you will not be able to install the signed certificate.

6. When you receive the certificate reply file containing the signed certificate, copy it to the WPS Hub server, at a location that is writable by both the WPS Hub process and the user logged into the WPS Hub server.
7. Connect to WPS Hub using SSH (see *Connecting to the Hub shell using SSH* (page 16)).
8. Enter `wpshub:https-certreply`.
9. Follow the prompts to install the signed certificate in the keystore:
   a. At the **Certificate reply file** prompt, enter the filename of the certificate reply file.
   b. At the **Enter existing alias** prompt accept the default value of `wpshub`. This is the alias that references the certificate and associated private key in the WPS Hub keystore. You should not change this value unless you have modified the WPS Hub keystore externally.

   The original self-signed certificate in the keystore is replaced with the signed certificate and HTTPS restarts.
10. Test the HTTPS configuration as described in *Testing the HTTPS configuration* (page 29).

Installing an existing signed certificate

You can install an existing signed certificate, which can be internally or externally signed. The certificate can be two files in PEM format (a certificate file, and a private key file) or in PKCS12 format (a single password-protected file containing the certificate and a private key).

Before installing the certificate, ensure that the common name in the **Subject** field in the certificate matches the WPS Hub server domain name. For example, for an internal deployment, this might be `wpshub-server.mydomain.local`, or for a deployment with a registered address, this might be `www.mycompany.com`).

To install an existing formally signed certificate:

1. Connect to WPS Hub using SSH (see *Connecting to the Hub shell using SSH* (page 16)).
2. At the shell prompt, do one of the following:
   - To install a certificate in PEM format: enter `wpshub:https-pemsignedcert`.
   - To install a certificate in PKCS 12 format: enter `wpshub:https-pkcs12signedcert`.
3. Follow the prompts to install the existing signed certificate:

a. At the **Certificate file** prompt, enter the folder path and filename of the file containing the signed certificate.
b. For PCKS12 certificates only: At the **Private key** prompt enter the certificate password. You are shown some information about the certificate you have just imported.
c. At the **HTTPS port number** prompt, either accept the default value of 8443, or specify the port number you want to use, if different.

A message is shown confirming that the HTTP protocol is no longer enabled, and that HTTPS has been enabled.

4. Test the HTTPS configuration as described in *Testing the HTTPS configuration* (page 29).

Installing a certificate directly into the keystore

If you have a signed certificate but are unable to install it by any of the methods above, you can install it directly into a Java keystore, then configure WPS Hub to use that keystore.

To install a certificate directly into the keystore:

1. Use the standard Java Keytool utility to create a Java keystore containing the signed certificate.
2. Use the **wpshub:https-keystore** shell command to configure the WPS Hub to use this external keystore for its HTTPS configuration.
3. If necessary, open the shell and enter **wpshub:https-useentry** to select which certificate entry to use in the keystore. Use this if you add a new entry to the keystore (for example, on certificate renewal) and need to alter the entry name used.

Testing the HTTPS configuration

Once you have configured HTTPS, it should be tested by accessing the WPS Hub web portal.

Before testing the HTTPS configuration, you might need to restart the WPS Hub service for the certificate changes to take effect. This especially applies if you have modified the keystore directly, rather than via WPS Hub shell commands.

To test access to the WPS Hub web portal after enabling HTTPS:

1. Navigate to `https://wpshub-server:port` where *wpshub-server* is the name of the machine where WPS Hub is installed and *port* is the HTTPS port number specified when configuring HTTPS (the default value is 8443).

   You might need to enable access to the selected HTTPS port number in the WPS Hub server firewall. If you need help doing this, consult your IT department.
2. If you have installed a certificate signed by a trusted root authority, the WPS Hub portal logging in page should be displayed.

3. If you are using a self-signed certificate, for security reasons your browser might not display the site initially or display a warning. In that case, you need to tell your browser that the site is safe as follows:
   a. If you see an information button or other warning beside the address, click it to see a warning that the site is not secure.
   b. Once you see the message warning that the site is not secure, look for the option to visit the page anyway. Depending on your browser, you might need to click **Details** or **Advanced**, then **Go to the webpage (not recommended)** or similar. The WPS Hub web portal page login page is displayed, with a warning in the address bar.
4. In the WPS Web Portal logging in page enter the username and password you specified for the WPS Hub administrator user, then click **Login**.

   Your browser shows the home page for the WPS Hub web portal.

WPS Hub keystore locations

By default, the standard WPS Hub keystore is located as follows:

- Windows: `C:\ProgramData\World Programming\WPS Hub\4\etc\keystore`.
- Linux: `/var/worldprogramming/wpshub-4/etc/keystore`.

If you need to access the keystore directly, you can use the WPS Hub shell command `wpshub:https-passwords` to display the keystore and entry passwords.

LDAP import of users and groups

WPS Hub can import users and groups from an Active Directory server using its LDAP interface. The import can be configured to re-occur at a specified interval, thus keeping the imported users and groups syncronised to the specified Active Directory server. Hub can either import all users and groups in an Active Directory server, or a specified subset. LDAP imported users and groups can exist alongside users and groups created within WPS Hub.

WPS Hub also supports LDAPS for increased security where Active Directory LDAPS is available. If you do use LDAPS, you also need to install a CA trust certificate. A CA trust certificate is only required if you are using LDAPS (recommended), where your LDAPS configuration will have a privately-signed security certificate. This certificate is above the LDAPS server security certificate in the trust chain, and is needed so WPS Hub can verify the authenticity of the LDAPS server security certificate in communications with the LDAPS server. This is only required for privately-signed certificates: if the LDAP server security certificate is publicly signed, the required trust certificate is already installed in the WPS Hub TrustStore.

Installing a CA trust certificate (LDAPS only)

To use LDAPS authentication, you will need to install a CA trust certifcate. To install a CA trust certificate, it needs to be copied to the WPS Hub server and then imported into its TrustStore.

To install the CA trust certificate in the TrustStore:

1. Copy the CA trust certificate to the WPS Hub server, at a location that can be accessed by both the WPS Hub process and the user logged into the WPS Hub server.
   - Windows: you could use a folder at the top level of the `C:\` drive that is neither a system folder, nor `C:\Users`.
   - Linux: you could use `/tmp`.
2. For Windows installations, start the Java keytool as described below:
   a. In the Windows Start menu, select **Windows System**, then right-click **Command Prompt**. Select **More**, then **Run as Administrator**.
   b. When prompted, enter a Windows administrator username and password.
   c. To start the Java keytool, in the command window, enter (on a single line):

```
C:\Program Files\World Programming\WPS Hub\4\jre\bin\keytool.exe" import
-trustcacerts -file cert-file
-alias wps_ca
-keystore "C\Program Files\World Programming\WPS Hub\4\jre\lib\security
\cacerts
```

3. For Linux installations, start the Java keytool as described below:
   a. Enter:

```
sudo /opt/worldprogramming/wpshub-4/jre/bin/keytool -import -trustcacerts -
file <cert-file> -alias wps_ca -keystore /opt/worldprogramming/wpshub4/jre/
lib/security/cacerts
```

   where *cert-file* is the filename of the CA trust certificate.
   b. When prompted, enter the sudo password for the logged-in user.
4. At the `Enter Keystore Password` prompt, enter the password for the Java TrustStore. By default, this is the initial password supplied with the WPS Hub Java installation, and is set to `changeit`. If this does not work, consult your IT department.
5. If you are satisfied that this is the correct certificate, at the `Trust this certificate?` prompt, enter `yes`.

The certificate is added to the TrustStore.

Configuring WPS Hub to use LDAP

Before configuring an LDAP connection, ensure that:

- The WPS Hub service is running.
- You know a username and password for a WPS Hub user in the Hub Administrators group.
- You know the LDAP or LDAPS configuration settings for your system.

If you are using LDAPS (recommended) and the LDAPS server security certificate is privately signed, you also need to obtain and install the Certificate Authority (CA) trust certificate in the default Java TrustStore on the WPS Hub server. This is the certificate above the LDAPS server security certificate in the trust chain, and is needed so that WPS Hub can verify the authenticity of the LDAPS server security certificate in communications with the LDAPS server. This is only required for privately-signed certificates; if the LDAP server security certificate is publicly signed, the required trust certificate is already installed in the WPS Hub TrustStore. Consult your IT department if you do not know how to obtain the required trust certificate.

To configure WPS Hub to use LDAP to import users and groups from from Active Directory:

1. Connect to WPS Hub using SSH (see *Connecting to the Hub shell using SSH* (page 16)).
2. At the shell prompt, to configure LDAPS (recommended), or LDAP, enter `wpshub:ldap-config`, then follow the prompts:
   a. Enter the URL as *`ldaps://ldaps-host:port`* (for example, `ldaps://ldaps-server.example.com:389`), replacing *ldaps* with *ldap* if appropriate.
   b. Enter the LDAP Authentication principal in the format *`accountname@suffix`* (for example, `ldapsacc@example.com`).
   c. At the LDAP Authentication credentials prompt, enter the password for the LDAP authentication principal account.
   d. Enter the **LDAP searchBase** (for example, DC=example, DC=com).
   e. At the **Search subtree** prompt enter `yes`.
   f. At the **Import users from all LDAP groups** prompt, enter `yes` if you want to import all users, otherwise, enter `no` and follow the prompts to select the groups to import. If you choose to import selected groups, you can press the Tab key to auto-complete the group names. Include yourself in the imported users, so that you can verify that imported users can access WPS Hub.
   g. At the **Edit advanced properties** prompt, if you have non-standard Active Directory fields, or if you are using an LDAP directory other than active directory, enter `yes` and follow the prompts, otherwise enter `no`.
   h. At the **Set synchronisation interval** prompt, enter `yes` if you want to configure automatic synchronisation between WPS Hub and the LDAP database, otherwise enter `no`.

      If you select automatic synchronisation you are also prompted for the synchronisation interval unit (enter one of minute, hour or day) and the synchronisation interval count (enter the interval at which to synchronise, in your chosen units).

   At this point, a confirmation message shows, saying how many users and groups were found, and asking if you want to apply the specified configuration. If you see an error message instead, you have probably entered an incorrect LDAP configuration property. In this case, you are prompted for the properties again so you can fix the error.

3. When the configuration is correct, enter `yes` to apply it, then `yes` again to synchronise WPS Hub with the LDAP or LDAPS users and groups.

If the configuration is successful, you are shown an information message saying how many users and groups have been created, and, if applicable, information about any conflicts with existing local users and how they were resolved. The imported LDAP users have now been added to the predefined WPS Hub group HubUsers. These users can now log onto WPS Hub using their LDAP credentials.

Testing an LDAP connection

A WPS Hub LDAP connection can be tested by using the WPS Hub web portal to confirm that users and groups have been imported.

To test a WPS LDAP connection:

1. In your web browser, go to the WPS Hub web portal. If you have configured HTTPS as recommended, the portal address is `https://`*`wpshub-server`*`:`*`port`*, otherwise you can use `http://localhost:8181`.

   If the machine running the WPS Hub server does not have a web browser, you can use another machine with a browser, and navigate to `https://`*`wpshub-server`*`:`*`port`* (if you have configured HTTPS) or `http://`*`wpshub-server`*`:`*`port`* (otherwise). In the second case, you might need to enable access to port 8181 in the WPS Hub server firewall. If you need help doing this, consult your IT department.
2. In the WPS Hub web portal logging in screen, log in as the WPS Hub administrator user you created initially.
3. In the web portal, click the arrow in the **Administration** section, and verify that the expected users and groups have been imported.
4. If you have included yourself in the imported LDAP users, log out from the current WPS Hub administrator user by clicking the username at the top right of the portal page, and selecting **Log out**. Then log in using your imported LDAP credentials and verify that you can see the tile for **Credential Management** functionality at least. You cannot see the tile for **Administration** functionality unless your LDAP user is in the Hub Administrators group.

Single sign on

WPS Hub can be configured to accept Kerberos single sign on authentication. Once single sign on has been configured, WPS Hub uses domain credentials to authenticate portal users without requiring a separate login. WPS Hub can also use domain credentials to authenticate requests from WPS Analytics, for example to resolve an authentication domain for the currently signed-on domain user.

Once WPS Hub has been configured to enable single sign on, when a user navigates to the WPS Hub web portal, the browser attempts to get a Kerberos Service Ticket (ST) for the HTTP service on the WPS Hub server within the configured default Kerberos domain. Similarly, when WPS Analytics attempts to authenticate with WPS Hub, for example, to resolve the credentials for an authentication domain, WPS Analytics attempts to get a Kerberos Service Ticket (ST) for the HTTP service on the host name specified in the `HUB_SERVER` system option.

You can use the `klist` command to show the service tickets you currently have. This enables you to see whether the browser or WPS Hub has acquired a service ticket or not.

Single sign on does not work from a Windows client that is running on the same machine as the WPS Hub server. This is because the browser uses the NTLM protocol, which is not supported by WPS Hub. In this case, you need to log in to WPS Hub manually.

Configuring single sign on

To configure single sign on, you need to:

- Configure the Kerberos server for single sign on to WPS Hub (see *Configuring the Kerberos server for single sign on to WPS Hub* (page 34)).
- Configure WPS Hub to support single sign on (see *Configuring WPS Hub for single sign on* (page 34)).
- Configure WPS Hub clients to support single sign on (see *Configuring WPS Hub clients for single sign on* (page 35)).

Configuring the Kerberos server for single sign on to WPS Hub

To configure Kerberos single sign on to WPS Hub:

1. Ensure that an HTTP service principal name (SPN) exists for the WPS Hub server.
2. For this service, generate a keytab file that WPS Hub can use to authenticate to the domain controller. Because this file contains a credential, ensure that it is appropriately protected.

Configuring WPS Hub for single sign on

Before you begin, if your Kerberos server uses strong cryptography (recommended), you need to ensure that the Java Runtime Environment on WPS Hub is configured to use unlimited strength cryptography. To do this, download the JRE Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files from Oracle and extract the JAR files to the folder `<wpshub-application path>/jre/lib/security`, where `<wpshub-application path>` is the location where WPS

Hub is installed. If Hub is un-installed in the future, these new files will be removed by the uninstallation process.

To configure WPS Hub for single sign on:

1. Log in to the server where WPS Hub is installed and restart the WPS Hub service.
2. Connect to WPS Hub using SSH (see *Connecting to the Hub shell using SSH* (page 16))
3. At the shell prompt, enter `wpshub:sso-config` and then follow the prompts as follows:
   a. For `Service name`, enter `HTTP` (this is the SPN specified for WPS Hub in the Kerberos configuration).
   b. For `host name`, enter the WPS Hub server domain name. For an internal deployment, you can accept the default value, for example, `wpshub-server.mydomain.local`. For an external deployment, enter the registered address, for example, `www.mycompany.com`.
   c. For `Kerberos domain`, enter the name of your Kerberos domain. If you do not know the name, consult your IT department.
   d. For `Keytab file`, enter the name and location of the Kerberos keytab file for the WPS Hub server HTTP service. If you do not know the name, consult your IT department.

Configuring WPS Hub clients for single sign on

To configure WPS Hub clients to use single sign on from web browsers and from WPS Analytics, Kerberos authentication needs to be enabled for browsers in the domain.

To configure WPS Hub clients for single sign on:

1. Enable Kerberos authentication for all browsers in the domain. These settings are usually managed centrally by your IT department, although browsers can be configured individually if necessary.

Testing single sign on

Before testing single sign on, ensure you are not logged in to WPS Hub. To test single sign on:

1. Ensure you are logged in on your Windows or Linux client as a domain user.
2. In your browser, navigate to the WPS Hub web portal address.

If single sign on has been successful, you should see the home page for the WPS Hub web portal, without needing to log in.

Troubleshooting single sign on

Kerberos single sign on can be complicated to configure. This section contains some troubleshooting tips.

- Windows can't perform a Kerberos single sign on within the same machine. So when testing, the browser in which you are running the portal and the server running the WPS Hub must be on different machines.
- On Linux you can test whether a client can acquire a service ticket using the `kvno` command as follows:

  ```
  kvno HTTP/<hostname>
  ```

  This attempts to acquire a service. If this fails, neither a browser nor WPS Analytics can acquire a service ticket. For example, if you do not have a valid Kerberos TGT you might see an error such as:

  ```
  kvno: Ticket expired while getting credentials for HTTP/hostname
  ```

  Performing a `kinit` is then necessary to acquire the TGT that is necessary to acquire the ST:

  ```
  $ kinit
  Password for user@domain:
  $ kvno HTTP/hostname HTTP/hostname : kvno = 5
  ```
- On both Windows and Linux, you can use the klist command to list Kerberos tickets so you can see whether the client has successfully acquired one.
- For authentication to the WPS Hub to be successful, there must be a user defined in WPS Hub with a name that matches the authenticated Kerberos principal name. A useful test is therefore to ensure that it is possible to manually authenticate to the WPS Hub web portal as the user.
  - For a Windows client this is the domain user with which you authenticated to the client machine, without the domain name.
  - For Linux this is the user principal specified at the time the kinit was performed, without the domain name.

Downloading and installing the WPS Hub MS Excel plug-in for Windows

The WPS Hub MS Excel plug-in allows you to connect an instance of Microsoft Excel to a WPS Hub installation. Published programs from that WPS Hub Deployment Services directory can then be run from MS Excel, with input passed to the program from MS Excel and output from the program displayed in MS Excel.

Downloading the WPS Hub MS Excel plugin

The WPS Hub MS Excel plugin installation package can be downloaded from the WPL website.

WPL must provide access to the WPS Hub tab in the **Downloads** area of the WPL website.

To download the WPS Hub MS Excel plugin installation package:

1. Go to https://www.worldprogramming.com/support/downloads and log in to your account.
2. Click **Downloads** at the top of the screen.

   WPS Analytics product downloads are displayed.
3. Click **WPS Hub** to view your WPS Hub downloads. If you do not see the **WPS Hub** tab, contact WPL Support to request access.

   WPS Hub product downloads are displayed.
4. Locate the WPS Hub MS Excel plugin and click **Download (.msi)**. Ensure that you choose the correct version for your MS Excel installation, either 32 bit or 64 bit.
5. Save the download to an accessible area.

   The download is saved. To install the WPS Hub MS Excel plugin, proceed to *Installing the WPS MS Excel plugin for Windows* (page 37).

Installing the WPS MS Excel plugin for Windows

You will require:

- Access to an account with administrator privileges to install the WPS Hub MS Excel plugin.
- MS Excel installed, but not running.
- A previously downloaded MSI installation file (see *Downloading the WPS Hub MS Excel plugin* (page 37)).

To install the WPS Hub MS Excel plugin on Windows:

1. Log in as an administrator to the computer that you want to install the WPS Hub MS Excel plugin on.
2. Run the downloaded MSI file to start the installer.
3. Read the license agreement, and if you accept, select **I accept the terms in the License Agreement**.
4. Click **Install**.

   The installation process runs.
5. Click **Finish**.

The WPS Hub MS Excel plugin has now been installed and can be used straight away.

Upgrading or reinstalling WPS Hub

Upgrading or reinstalling a WPS Hub installation can be done by following the installation instructions, with the possible addition of three extra tasks: backing up the Hub database (recommended), clearing Hub's bundle cache (also recommended), and erasing the existing installation (if errors are encountered, but can be performed just in case).

Backing Up the Hub database

Upgrading or re-installing WPS Hub may introduce compatibility issues between the pre and post upgrade WPS Hub databases. As such, it is recommended that you perform a database backup (see *Manually backing up the WPS Hub database* (page 39)) before an upgrade or a re-installation; followed by a restoration afterwards (see *Restoring the WPS Hub database* (page 40)), which will reconcile the old and new databases if required. Ensure that you move straight from the installation to the restoration without starting the Hub Service in between.

Clearing Hub's bundle cache (Windows only)

When upgrading or reinstalling new versions of Hub for Windows, it is recommended that you clear Hub's *bundle cache* after installation of the new version, but before starting Hub services. To clear the cache, delete the directory `%PROGRAMDATA%\World Programming\WPS Hub\4\data\cache`.

Erasing the existing Hub installation

If you receive an error when attempting to upgrade an existing installation, you might need to erase the existing installation first (see *Erasing an existing WPS Hub installation* (page 40)).

WPS Hub Backing up and Restoring

It is recommended that the WPS Hub database and configuration files are backed up as part of your regular system backing up routines.

Configuration information for WPS Hub is stored in `.cfg` files (for default configuration information) and `.changes` files (for subsequent changes that you make to the default configuration). These files are stored in the following folders:

- Windows: `C:\Program Data\World Programming\WPS Hub\4\etc`
- Linux: `/var/worldprogramming/wpshub-4/etc`

Ensure that these folders are backed up regularly as part of your regular system backing up routines.

Backing up and restoring the WPS Hub database

It is recommended that you back up the WPS Hub database as part of your regular system backups, or prior to upgrading a WPS Hub Installation.

Manually backing up the WPS Hub database

The WPS Hub database can be backed up from the Hub command line interface.

To back up the WPS Hub database:

1. Connect to WPS Hub using SSH (see *Connecting to the Hub shell using SSH* (page 16)).
2. Enter the command: `wpshub:backuprepo --out backup filename.db.bk`. The backup will be stored in `/var/worldprogramming/wpshub-4/` in Linux, or `C:\ProgramData\World Programming\WPS Hub\4` in Windows.

   All data in the WPS Hub is now backed up.

Scheduling backup of the WPS Hub database

The WPS Hub default internal database can be scheduled to automatically back itself up.

To configure scheduled backup for the WPS Hub internal database, once the database has already been configured:

1. Connect to WPS Hub using SSH (see *Connecting to the Hub shell using SSH* (page 16)).
2. Enter `wpshub:db-config-internal`, followed by **Return**.
3. Accept the default **Database file path** by pressing **Return** (this will be the path you configured when setting up the database).
4. At the **Schedule backup?** prompt, type yes, followed by **Return**.
5. At the **Backup archive file path** prompt, either accept the quoted path by pressing **Return** or specify a new file path.
6. At the **Advanced scheduling?** prompt, type either no or yes, followed by **Return**, and then set the scheduling as follows:
   - If you type **No**: Specify a **Backup Interval** by typing **daily** or **weekly** followed by **Return**, and then specify a time by typing the **hour**, followed by **Return**, and then the **minute**, followed by **Return**.
   - If you typed **Yes**: Specify a cron expression to define the scheduling, using Quartx syntax, followed by **Return**.
7. At the **OK to use these tables?** prompt, type **yes**, followed by **Return**.

8. At the **Bootstrap now?** prompt, type **No**, followed by **Return**.

Restoring the WPS Hub database

The WPS Hub database is restored using the Hub command line interface. A restoration will repopulate the new database with the old data, even if the database structure in the target WPS Hub installation has changed compared to the WPS Hub installation at the time of backing up.

To restore the WPS Hub database:

1. Connect to WPS Hub using SSH (see *Connecting to the Hub shell using SSH* (page 16)).
2. To restore your WPS Hub repository, enter: `wpshub: restorerepo --in <backup file name>.db.bk`.
3. Start the WPS Hub service.

Backing up and restoring the WPS Hub configuration

It is recommended that you back up the WPS Hub configuration as part of your regular system backups.

Configuration information for WPS Hub is stored in `.cfg` files. These files are stored in the following folders:

- Windows: `C:\Program Data\World Programming\WPS Hub\4\etc`
- Linux: `/var/worldprogramming/wpshub-4/etc`

Ensure that these folders are backed up regularly as part of your regular system backing up routines.

Erasing an existing WPS Hub installation

If you receive an error when attempting to upgrade an existing installation, you may need to erase the existing installation first. Erasing the installation as described here won't delete WPS Hub configuration settings.

To erase an existing WPS Hub installation:

1. Depending on your operating system:

- For Linux, at the command line interface enter the command:`sudo yum erase wpshub-4`. Press `y` followed by Return to confirm when asked.
- For Windows, use the **Add or Remove Programs** feature.

The existing WPS Hub installation has been removed.

Legal Notices

Trademarks

WPS and World Programming are registered trademarks or trademarks of World Programming Limited in the European Union and other countries. (r) or ® indicates a Community trademark.

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

All other trademarks are the property of their respective owner.

General Notices

World Programming Limited is not associated in any way with the SAS Institute.

WPS is not the SAS System.

The phrases "SAS", "SAS language", and "language of SAS" used in this document are used to refer to the computer programming language often referred to in any of these ways.

The phrases "program", "SAS program", and "SAS language program" used in this document are used to refer to programs written in the SAS language. These may also be referred to as "scripts", "SAS scripts", or "SAS language scripts".

The phrases "IML", "IML language", "IML syntax", "Interactive Matrix Language", and "language of IML" used in this document are used to refer to the computer programming language often referred to in any of these ways.

WPS includes software developed by third parties. More information can be found in the THANKS or acknowledgments.txt file included in the WPS installation.