



1Integrate

Installation Guide

Product Version: v 5.1

Document Version: v 4.0.1

Document Date: 24/01/2025

Copyright 2025 1Spatial plc and its affiliates.

All rights reserved. Other trademarks are registered trademarks and the properties of their respective owners.

No part of this document or any information appertaining to its content may be used, stored, reproduced or transmitted in any form or by any means, including photocopying, recording, taping, information storage systems, without the prior permission of 1Spatial plc.

1Spatial
Unit F7, Stirling House
Cambridge Innovation Park
Denny End Road
Waterbeach
Cambridge
CB25 9PB
United Kingdom

Phone: +44 (0)1223 420414

Fax: +44 (0)1223 420044

Web: www.1spatial.com

Every effort has been made to ensure that the information contained in this document is accurate at the time of printing. However, the software described in this document is subject to continuous development and improvement. 1Spatial plc reserves the right to change the specification of the software. 1Spatial plc accepts no liability for any loss or damage arising from use of any information contained in this document.

Contents

1 Introduction	1
Audience	1
Licences	1
1Spatial Product Support	1
2 Prerequisites	2
Sizing a Server for 1Integrate	2
Storage Requirements	2
Microsoft Visual C++ Redistributable (Windows only)	3
Configuring a Database Server for 1Integrate's repository	3
Supported Database Servers	4
Creating the repository	4
Required Packages (Linux only)	5
ICU Libraries (Linux only)	5
Data Store Prerequisites	6
3 Installing 1Integrate	9
Deployment Preparation	9
Copying the directory	9
Configuring System Properties	9
Required	10
Login Banner Configuration	12
Configuration	12
NIC/Network Adaptor Configuration	13
Reconnect Repository on database restart	13

Custom Extensions	14
Including Custom Extensions	14
Configure Memory	14
Setting the environment variables	14
Changing the start-up scripts	15
Configuring Users and Permissions	16
Permissions	16
Group Permissions	17
Default Users	18
Custom Users	19
Encrypted Data Store Parameter Settings	19
Configuring AES with a Custom Key	19
4 Deployment	21
Standard Deployment	21
JAVA_HOME	21
Starting 1Integrate	22
Configure additional Engines	22
Engine Labels	24
Service Deployment (Windows Only)	25
JAVA_HOME	25
Using Specific User Accounts	26
Deploying the Service	26
install-Interface-Service.cmd	26
install-Engine-Service.cmd	26
Configure additional engines (Service)	27
5 Testing the Installation	28
6 Upgrading an Installation	29

JAVA_HOME	30
-----------------	----

1 Introduction

This guide explains how to install 1Integrate.

The procedures apply to both Windows and Linux environments, unless specifically indicated.

For more information about new features and changes in this release, and hardware and software requirements, refer to the *1Integrate Release Notes*.

Audience

This guide is intended for personnel responsible for the installation, configuration, and administration of software.

The procedures detailed in the guide should be performed by a system administrator who is familiar with the application environment of the organisation.

Licences

1Integrate licences will be issued via email.

1 Spatial Product Support

If assistance is required while installing 1Integrate, please call 1Spatial support on +44 (0)1223 423069, or visit the support section of the 1Spatial website via the Services menu at www.1spatial.com.

2 Prerequisites

Before installing 1Integrate, please ensure you have met all system requirements and installed all necessary prerequisite components:

- Check the [release notes](#) to find the system requirements for your 1Integrate version.
- ["Microsoft Visual C++ Redistributable \(Windows only\)" on the next page](#)
- ["Configuring a Database Server for 1Integrate's repository" on the next page](#)
- ["ICU Libraries \(Linux only\)" on page 5](#)
- ["Required Packages \(Linux only\)" on page 5](#)

Sizing a Server for 1Integrate

There are a number of minimum requirements for server size in order to install 1Integrate


Depending on your intended configuration you will need, as a minimum:

PER INTERFACE

- 2 CPU Cores
- 2GB RAM

PER ENGINE

- 1 CPU Core
- 1GB RAM

 **Note:** In order to achieve optimal performance, a dedicated core per engine is recommended.

While being processed, data is stored on disk. So the recommended available storage varies on a case by case basis.

While improving CPU and memory will increase performance, the biggest gains can be made from using fast disk storage, e.g. SSDs.

Storage Requirements

The database storage requirements for the 1Integrate repository (1Integrate's internal usage, not the source of spatial data to be processed) depend on the

usage of 1Integrate.

- The size and number of 1Integrate entities (rules, actions, data stores and sessions).
- How many sessions will be running in parallel, and how many non-conformances or reports they will generate, before they are stopped/ reset.
- The size of any data that is uploaded as files to a data store or via 1Data Gateway when 1Integrate is configured to store uploaded files in the repository.

A small set of rules and actions which read data from a database or service and do not run many sessions in parallel will need much less repository database storage (e.g. 500 MB) than one in which many huge data files are uploaded in parallel (which might need 500GB).

Microsoft Visual C++ Redistributable (Windows only)

Microsoft Visual C++ 2015-2019 64-bit Redistributable packages are required for Windows installations of 1Integrate.


These can be downloaded from the Microsoft [website](#) (`vc_redist_x64.exe`).

For more information about the version numbers of pre-requisites, please refer to the relevant release notes for your installation.

Configuring a Database Server for 1Integrate's repository

1Integrate requires access to an database server to store configuration data such as rule or session definitions, and conformance results. This is referred to as the "1Integrate repository".


By default, 1Integrate will use a H2 database as the Repository. This will store the 1Integrate metadata and entities on the same file system as the 1Integrate installation.

 **Warning:** This is intended for local, development work and will enable you to get started quickly. Do not use the default H2 database in production environments. See the following database options for recommended repository servers.

Supported Database Servers


The following additional Database Servers are supported:

- *Oracle Enterprise*
- *PostgreSQL*
- *Microsoft SQL Server*

 **Note:** Please refer to the relevant product documentation when installing your chosen database server.

Creating the repository

You will need to create a database and user (Oracle only requires a User) for your 1Integrate repository on your selected server.

 **Note:** When deploying 1Integrate to multiple servers, one database user is required per environment.

ORACLE

Using Oracle, 1Integrate requires a database user. The setup should be performed as recommended by your Database Administrator who must grant the user access to the database and at least the following permissions:


- `create session`
- `create sequence`
- `create table`

MICROSOFT SQL SERVER DATABASE

Using SQL Server Database, you will need a Database and a User that can access that database.

The setup should be performed as recommended by your Database Administrator who must grant the user access to the database and at least the following permissions:

- `create sequence`
- `create table`

 **Note:** Microsoft SQL server supports two methods of authenticating users, but these will differ depending on your operating system:

- Windows: SQL Server authentication and Integrated Windows Authentication
- Linux: SQL Server authentication

CREATE A POSTGRESQL USER

Create a user that can access that database with the default properties. This will ensure the user has all required privileges. Alternatively the set up should be performed as recommended by your Database Administrator who must grant the user access to the database and at least the following permissions:

- `create sequence`
- `create table`

Required Packages (Linux only)

The following package is a prerequisite for Linux installations of 1Integrate:

- **libX11**

You must install **libX11** via the system's package manager.


ICU Libraries (Linux only)

ICU libraries are required for data and timestamp support in Linux.

A root user (or a user with root access privileges) is required to copy the files from the installation package and run the `ldconfig` command.

INSTALL THE ICU LIBRARIES

1. Run the `su` command to switch to the root user.
2. Copy the `.so` files from the installation folder (within the ICU folder) to the `/usr/local/lib64/` folder.
3. Create a new configuration file: `/etc/ld.so.conf.d/integrate.conf`

 **Note:** Both the `.so` files and the `.conf` file must have read permissions for all users.

4. Inside the configuration file, reference the location of the ICU libraries, for example: `/usr/local/lib64/*`
5. Enter the following `ldconfig` command:

```
/sbin/ldconfig -v /usr/local/lib64/
```

Alternatively, edit `~/.bashrc` or similar for the user used to run 1Integrate to include `/usr/local/lib64` on the `LD_LIBRARY_PATH`, as in the following example:

```
# User specific
if [ -z "$LD_LIBRARY_PATH" ]; then
export LD_LIBRARY_PATH="/usr/local/lib64"
else
export LD_LIBRARY_PATH="/usr/local/lib64:$LD_LIBRARY_PATH"
fi
```

Data Store Prerequisites

1Integrate supports the following types of Data Stores for Read and Commit mapping and Copy To mapping. Pay particular attention to the prerequisites attached to certain Data Store types.

There are two methods for writing data in 1Integrate, however these are not supported by all Data Store types:

- Commit Task
- Copy To Task

Data Store Type	Read	Write	
		Commit	Copy To
Autodesk DWG	✓	✗	✓
Bentley Microstation Design V8 DGN	✓	✗	✗
Contextual Data Store	✓ ¹	✓	✓
Delimiter Separated Values (DSV)	✓	✗	✓
Esri ArcGIS Services	✓	✓	✓
Esri File Geodatabase	✓	✓	✓
Esri Shapefile	✓	✗	✓
FME Flow	✓	✗	✗
GeoJSON	✓	✗	✓
Geography Markup Language (GML)	✓	✗	✗
Google BigQuery	✓ ²	✗	✓ ³
MapInfo TAB	✓	✗	✓
MariaDB	✓	✓	✓

¹This format is only available as an Extension.


²This format is only available as an Extension.

³This format is only available as an Extension.

Data Store Type	Read	Write	
		Commit	Copy To
Microsoft Access	✓	✗	✗
Microsoft SQL Server	✓	✓	✓
Non-Conformance Report	✓	✗	✗
OGC GeoPackage	✓	✓	✓
OGC Web Feature Service (WFS)	✓ ¹	✗	✗
Oracle	✓	✓	✓
PostgreSQL/PostGIS	✓	✓	✓
Schema Only	✓	✗	✗

¹This format only supports 2D geometries.

3 Installing 1Integrate

 **Note:** Before proceeding, ensure you have completed all pre-requisite steps (see "Prerequisites" on page 2).


1Integrate runs as several separate applications: an interface application and one or more engine applications.

The following instructions describe how to configure an interface and an engine. This is suitable for running both parts of 1Integrate on one host.

Deployment Preparation

The following information is required prior to deploying 1Integrate:

- JDBC URL of the repository created
- Username and password for the database user
- Path of the **tnsnames.ora** file (if used)
- Location of a directory that can be used to store 1Integrate temporary files

 **Note:** On both Windows and Linux, paths must be entered with forward slashes (/) as the path separator instead of backslashes (\). Ensure that no spaces are entered after each parameter as this will result in an unsuccessful installation.


Copying the directory

Copy the directory (typically called 1Integrate-[version]) from the installation package to a local folder, then extract the files to a location such as C:\Program Files\1Spatial\1Integrate-[Version]\. These files will be used to configure settings for 1Integrate in the following procedures.




In the installation, the **server-interface** directory contains the configuration for the 1Integrate interface and the **server-engine** directory contains the configuration for the engine.






Configuring System Properties

The following parameters can be configured in the **settings.properties** file, located within the **SETTINGS** folder within the installation location (e.g. C:\Program Files\[1Integrate_Directory]\SETTINGS).

 **Note:** On Windows, folder paths within the **settings.properties** file must be specified using a forward slash. Spaces in folder names are allowed (e.g. C:/Program Files/1Spatial/1Integrate).

Required

Property	Type	Description
repository.driver	string	<p>The database type. Defaults to a H2 database, but Oracle, SQLserver or PostGresql are also supported. Define using the values: h2, oracle, sqlserver or postgresql.</p> <div>  Note: Do not use the default H2 database in production environments. See the other database options for recommended repository servers. </div>
repository.url	string	<p>The URL for the database in which to store 1Integrate metadata and entities (Rules, Actions, Sessions and so on), and files that are uploaded to the server used by the data store (such as MapInfo Tab files).</p> <p>By default, this will be populated by the H2 database details.</p> <p>For other supported databases the format is:</p> <ul style="list-style-type: none"> For Oracle: jdbc:oracle:thin:@[HOSTNAME]:[PORT]/[Service] or jdbc:oracle:thin:@[HOSTNAME]:[PORT]:[SID] For MS SQL: jdbc:sqlserver://[serverName][\\instanceName][:portNumber];databaseName=1Integrate <p>For MS SQL with Windows Authentication: repository.url=jdbc:sqlserver://[serverName][\\instanceName][:portNumber];integratedSecurity=true;databaseName=1Integrate</p> <div>  Note: You must define either a serverName or instanceName. See Microsoft documentation for further guidance. </div> <ul style="list-style-type: none"> For PostgreSQL: <div>  Note: You will only need to specify a port if you are not using the default. </div> <ul style="list-style-type: none"> To use the <i>public schema</i> to store the repository: jdbc:postgresql://[HOSTNAME]:[PORT(if non-default)]/[DATABASE] To use a <i>specified schema</i> to store the repository: jdbc:postgresql://[HOSTNAME]:[PORT(if non-default)]/[DATABASE]?currentSchema=[REPOSITORYSCHEMA]

Property	Type	Description
<code>repository.username</code>	string	Username to connect to the repository schema, in which to store 1Integrate metadata and entities (e.g. Rules, Actions, Sessions) as well as files that are uploaded to the server used by the data store (e.g. MapInfo Tab files).
<code>repository.password</code>	string	Password for the user described by <code>repository.username</code> .
<code>ts_license</code>	string	The location and name of the provided product licence file.
<code>temp.directory</code>	string	This is a temporary folder location for files that are eventually stored in the database.
<code>cache.directory</code>	string	<p>The location of the cache directory.</p> <p>When a Session is run, a folder is created called "1Integratecache", within which the cache is stored.</p> <div> <p> Note: This stores the data cache from data read by 1Integrate. This may require large amounts of disk space depending on the size of data being read into Sessions and the number of concurrent Sessions.</p> </div> <div> <p> Note: Multiple engines need to share the same cache location to allow a paused Session to be picked up and continued/rewound without restarting it.</p> </div> <p>For Linux, if left blank this will default to /tmp. For Windows, if left blank this will default to C:\Users\[user]\AppData\Local\Temp.</p> <div> <p> Note: It is recommended that you change the default <code>cache.directory</code> to a different location to avoid your cache being unexpectedly cleared by the system.</p> </div>
<code>interface.http.port</code>	integer	<p>The port to use for the interface.</p> <div> <p> Note: If possible, use the default port number specified in the file. If there is a clash with another application that already uses this port number, increment the value by 1 until an unused port number is found.</p> </div>
<code>engine.http.port</code>	integer	<p>The port to use for the engine.</p> <div> <p> Note: If possible, use the default port number specified in the file. If there is a clash with another application that already uses this port number, increment the value by 1 until an unused port number is found.</p> </div>


Login Banner Configuration

You can configure a login banner to appear at the top of the login page to include additional text for your users. This banner will not replace the 1Integrate login graphics or logo, but will appear above it, at the top of the page.



Login page with an example banner

The banner is defined by a file that includes a snippet of text or other HTML.

 **Note:** The Login box will remain at the same location and in the foreground, so if the banner is too large it will appear behind the login box.

HTML tags, while not required, can be included for further customisations. For a full list of supported tags, see the drop-down below.

SUPPORTED HTML TAGS


```
<h1> <h2> <h3> <h4> <h5> <p> <span> <br> <text> <body> <title> <html>
<head> <meta> <b> <strong> <i> <em> <mark> <small> <del> <ins> <sub>
<sup>
```

The file can have any extension and must be placed on the filesystem accessible to all interface servers in the deployment.

Configuration


1. Create your HTML file.
2. In the **settings.properties** file, add the following:

Parameter	Description
LoginBanner	Enter a file path to a HTML document you have defined e.g:loginBanner=C:/Apps/1Spatial/1Integrate/ Banner%20Folder/Heading_Banner.html

 **Note:** The HTML file must be accessible by each Interface server.

NIC/Network Adaptor Configuration

The Grid discovery used to find engines by default uses the first found non-loopback address, for example a machine with Ethernet adaptors "eth0" and "eth1" and Local Loopback "lo" will likely use "eth0".

 **Note:** If you do not need to override the adaptor default behaviour, then the following properties do not need to be included.


CONFIGURE NIC/NETWORK ADAPTOR

The following properties must be included in the **settings.properties** file:

```
grid.local.address=[NIC Address]
grid.discovery.tcp.port=[default: 51300]
grid.communication.tcp.port=[default: 51401]
```


Where:

- **grid.local.address** - specifies the IP address of the network adaptor used for grid communication.
- **grid.communication.tcp.port** and **grid.discovery.tcp.port** - allows environments to specify known ports (for example, when using a firewall).

 **Note:** The communication port must be a minimum of 100 greater than the discovery port, in order to avoid conflict.

Reconnect Repository on database restart

By default, when your deployment of 1Integrate disconnects from a repository (e.g. database restart or network issue) it will attempt to reconnect at the configured interval.

 **Note:** To make changes to this reconnection, see the Configuring System Properties table and locate the **repository.validation** and **repository.validation.millis** parameters detailed there and in the **settings.properties** file.

Custom Extensions


1Integrate can be extended in the following ways:

Custom Built-ins

For information on creating custom built-ins, please refer to the

Custom data stores

For information on custom data stores, please contact 1Spatial support.


 **Note:** When upgrading your installation of 1Integrate, ensure you follow the upgrade steps so that your custom extensions persist.

Including Custom Extensions

Custom extensions can be stored in the `EXTENSIONS` folder in the 1Integrate bundle.

If you want to define your own directory outside of the 1Integrate bundle, then you will need to change the `settings.properties` file. To do this, add the following, replacing the default value:

`1integrate.extension.dir=<directory location>`

 **Note:** Manually setting the directory location is recommended if you would like to persist custom extensions between installations and upgrades.

Configure Memory

To update the maximum available heap size in 1Integrate, you can change the start scripts or set an environment variable.

Setting the environment variables

You can set `INTEGRATE_ENGINE_OPTS` and `INTEGRATE_INTERFACE_OPTS` as environment variables.

If you do this, settings will persist between installations as you will not need to edit the start-up scripts.

Changing the start-up scripts

WINDOWS

1. Edit **start-Interface.cmd** or **start-Engine.cmd**.
2. Locate the following lines:


```
SET "INTEGRATE_INTERFACE_OPTS=%INTEGRATE_INTERFACE_
OPTS%"
```

or

```
SET "INTEGRATE_ENGINE_OPTS=%INTEGRATE_ENGINE_OPTS%"
```

3. Define the required size using a Java system property, e.g. to set the interface to 2GB:

```
SET "INTEGRATE_INTERFACE_OPTS=%INTEGRATE_INTERFACE_
OPTS% -Xms2g -Xmx2g"
```

 **Note:** For performance reasons it is recommended that your initial heap size matches your maximum heap size.

LINUX

1. Edit **start-Interface.sh** or **start-Engine.sh**.
2. Locate the following lines:


```
export "INTEGRATE_INTERFACE_OPTS=%INTEGRATE_
INTERFACE_OPTS%"
```

or

```
export "INTEGRATE_ENGINE_OPTS=%INTEGRATE_ENGINE_
OPTS%"
```


3. Define the required size using a Java system property, e.g. to set the interface to 2GB:

```
export "INTEGRATE_INTERFACE_OPTS=%INTEGRATE_
INTERFACE_OPTS% -Xms2g -Xmx2g"
```

 **Note:** For performance reasons it is recommended that your initial heap size matches your maximum heap size.

Configuring Users and Permissions

Users, passwords, and permissions can be edited within 1Integrate. Access to the different sections of 1Integrate is controlled by a series of permissions. Changing these permissions is achieved by changing **Roles**.

 **Warning:** By default, 1Integrate is deployed with example users and passwords included. This enables a quick set-up process, but for security reasons it is **HIGHLY RECOMMENDED** that you:

- Change all passwords from the default to unique values.
- Change the user names to ones relevant to your organisation.
- Do not store user names and passwords in plain text and use an external identity management system instead.

For stronger security, manage users in an external identity management system by using one of the following supported authentication protocols:

- [OIDC](#)
- [SAML](#)
- [LDAP](#).

This ensures that passwords and user names are not stored in the application server but managed by your IT department instead.

Permissions

Each user is assigned one or more Permissions. These permissions determine a user's privileges and the areas of the functionality to which they have access.

When configuring Permissions in the application server they will be interchangeably known as "Roles", but the two terms are interchangeable.

Permission	Description
1int-datastores-read	Grants the ability to read Data Store objects and folders at the endpoint.
1int-datastores-write	Grants the ability to write Data Store objects and folders at the endpoint..
1int-rules-read	Grants the ability to read <i>Rule</i> objects and folders at the endpoint..
1int-rules-write	Grants the ability to write Data <i>Rule</i> objects and folders at the endpoint..
1int-actions-read	Grants the ability to read <i>Action</i> objects and folders at the endpoint.
1int-actions-write	Grants the ability to write <i>Action</i> objects and folders at the endpoint.
1int-actionmaps-read	Grants the ability to read <i>Action Map</i> object and folders at the endpoint.
1int-actionmaps-write	Grants the ability to write <i>Action Map</i> objects and folders at the endpoint.
1int-sessions-read	Grants the ability to read <i>Session</i> objects and folders at the endpoint.
1int-sessions-write	Grants the ability to write and edit the <i>Session</i> objects and edit folders.
1int-sessions-control	Grants the ability to control a session with the "Play", "Pause", "Rewind" and "Stop" functions.
1int-sessions-results	Grants the ability to access all Session results, including both Task and Session results i.e. Validation errors.
1int-grid-read	Grants the ability to view the engine grid.
1int-grid-write	Grants the ability to edit the engine grid.
1int-api-keys	Grants the ability to manage the API Key functionality in the Administration section.
1int-access-groups	Grants the ability to manage and configure Access Groups.
1int-repository	Grants the ability to access the Repository Administration functions and to see the Environment and System Properties.

Group Permissions


There are two sets of group permissions available that can be used to quickly assign a common set of permissions to a user.

Group Permission	Description
1int-user	<p>The User is designed to be applied to standard users, this role includes:</p> <ul style="list-style-type: none"> • 1int-datastores-read • 1int-datastores-write • 1int-rules-read • 1int-rules-write • 1int-actions-read • 1int-actions-write • 1int-actionmaps-read • 1int-actionmaps-write • 1int-sessions-read • 1int-sessions-write • 1int-sessions-control • 1int-sessions-results • 1int-grid-read
1int-admin	<p>The Admin to includes all permissions and is designed for those that will be performing administrative functions.</p> <p>Includes all the permissions of 1int-user with the addition of:</p> <ul style="list-style-type: none"> • 1int-grid-write • 1int-api-keys • 1int-access-groups • 1int-repository

Default Users

The following users are created by default upon installation:


Username	Password	Assigned permissions
INTFull	integrate1	This default User has the 1int-admin Group Permission applied.
INTAdmin	integrate101	This default User has the 1int-admin Group Permission applied.
INTUser	integrate102	This default User has the 1int-user Group Permission applied.

 **Note:** You will need to restart 1Integrate for any changes to user and permissions to take effect.

Custom Users

To configure Users and Permissions, navigate to the `\[1Integrate_Directory]\SETTINGS` folder. This folder contains the following files:

- **users.properties** contains a list of usernames and passwords, in the form `username=password`.

 **Note:** All users listed in the previous table are included as default.

- **roles.properties** contains a mapping from user names to 1Integrate permissions in the form `username=permission1,permission2,permission3`

Encrypted Data Store Parameter Settings


1Integrate stores all Data Store configuration in the repository, including sensitive parameters, such as database passwords. Sensitive information is encrypted, with two options available:

- **Advanced Encryption Standard (AES)**

This is the standard method of encryption in 1Integrate and will be used as default.


- **AES with a custom key**

This method requires additional configuration. For more see [Configuring AES with a custom Key](#) below.

 **Note:** AES with a custom key is the most secure method of encryption in 1Integrate. Be aware that sensitive information (e.g. passwords) created on a custom key installation will be invalidated if the key is removed or changed. Sensitive information will also only be compatible with other 1Integrate installations configured with the same key.

Configuring AES with a Custom Key

AES with custom key can be configured both at the point of installation or after.

 **Note:** It is recommended you carry out the implementation of a custom key at the point of installation to avoid invalidating already stored passwords.

3 Installing 1Integrate

1. Open the `settings.properties` file for your 1Integrate installation.
2. Add and set:

`encryption.key=<Input your custom key>`

3. If 1Integrate is running, restart your installation.

4 Deployment


Once you have installed 1Integrate, it is now ready to be deployed. The method of deployment will differ depending on your Operating system or deployment type.

Standard Deployment

Before running your start-up scripts, you must first ensure your deployment is configured to refer to the correct version of Java. For the Java version supported for your release of 1Integrate, please refer to the release notes.

JAVA_HOME

Once you are sure you have the correct version of Java installed, you will need to ensure that the JAVA_HOME environment variable points to this for the user running 1Integrate.

 **Note:** If the JAVA_HOME variable is already defined on your system with the correct version of Java, then you do not need to do anything.

If the JAVA_HOME environment variable is not set, you have the following options:

- Set the JAVA_HOME environment variable for the user running 1Integrate to point to the correct version of Java.
- Follow the steps below for Windows and Linux to set JAVA_HOME from the 1Integrate start-up script.

To set the JAVA_HOME environment variable in the start-up scripts, carry out the following steps:

WINDOWS

1. Edit **start-Interface.cmd** and **start-Engine.cmd**.
2. Locate the following lines:

```
@REM Optionally uncomment and set this path to be  
the correct for your environment  
REM SET "JAVA_HOME=C:\PROGRA~1\Java\[jdk_version]
```

3. Uncomment the second line by removing "REM", so that it begins with "SET". Ensure that it is pointing to the location of Java home for the correct version of Java installed as part of the pre-requisites. The resulting line should appear as follows:

```
SET "JAVA_HOME=C:\PROGRA~1\Java\[jdk_version]"
```

4. Once configured, run the scripts to start the 1Integrate components.

LINUX

1. Edit **start-Interface.sh** and **start-Engine.sh**.
2. Uncomment the line just below "#Optionally uncomment and set this path to be the correct for your environment":

```
export JAVA_HOME="/opt/jdk"
```

3. Once configured, run the scripts to start the 1Integrate components.

Starting 1Integrate

You can now run the start-up scripts:


- **start-Engine**
- **start-Interface**


Once run, 1Integrate will be ready to use. For information on ensuring your installation and deployment has been successful, please refer to ["Testing the Installation" on page 28](#).

Configure additional Engines

1Integrate can be deployed with multiple engines each of which can run either on the same server as the interface or on separate machines (as long as they are on the same network as the interface).

Engines can be run in parallel, deployed using different port numbers. This requires a license from 1Spatial that enables the required number of parallel engines.


 **Note:** On a machine running Windows, this may present a path name length error. If this occurs, copy the **server-engine** folder to a higher level directory, rename it, and paste it back into the **1Integrate_[version]** folder.

 **Note:** For more advanced environments, contact either your distributor or 1Spatial prior to continuing to discuss an appropriate deployment plan.


You will need to copy and edit a different file, depending on if you are using Windows or Linux.

WINDOWS

1. Create a copy of the **server-engine** folder and rename it **server-engine2** etc.

 **Note:** You must suffix server-engine with an integer.


2. Inside the cloned directory, open the **configuration\engine.conf.bat**
3. Update SET "ENGINE_NUMBER=1" to the corresponding number e.g
SET "ENGINE_NUMBER=2"
4. Create a copy of the **start-Engine.cmd** file and rename it.

 **Note:** When deploying multiple engines, ensure you run each **start-Engine.cmd** file you have created.

5. Edit this file to refer to "ENGINE_NUMBER=2" instead of "ENGINE_NUMBER=1" (or "ENGINE_NUMBER=3" for the third engine etc.)


LINUX

1. Create a copy of the **server-engine** folder and rename it **server-engine2**.

 **Note:** You must suffix server-engine with an integer.

2. Create a copy of the **start-Engine.sh** file and rename it **start-Engine2.sh**.

Edit this file to refer to **server-engine2** instead of **server-engine** (or **server-engine3** for the third engine etc.) and change the port offset at the end of the startup line from 100 to 200 (or 300 for the third engine etc.).

 **Example:** The line
`./bin/standalone.sh -P=SETTINGS/settings.properties -
Djboss.server.base.dir=server-engine=Djboss.socket.binding.port-
offset=100`
would be changed to
`./bin/standalone.sh -P=SETTINGS/settings.properties -
Djboss.server.base.dir=server-engine2
=Djboss.socket.binding.port-offset=200.`

Engine Labels

You can label engines you are deploying to group them, and control which Sessions run on different groups.

WINDOWS

1. Open the `server-engine` directory for the engine you will be applying labels to.
2. Within the `configuration` directory, open the **engine.conf.bat** file.
3. Locate and uncomment the following lines:

```
SET "engine.labels="
```

4. Add engine labels as a comma separated list:

```
SET "engine.labels=label1,label2,label3"
```

5. Repeat the steps for each engine being deployed.

LINUX

1. Open the **start-Engine.sh** script for the engine you are applying labels to.
2. Locate and uncomment the following lines:

```
export engine.labels=""
```

3. Add engine labels as a comma separated list:

```
export engine.labels="label1,label2,label3"
```

4. Repeat the steps for each engine being deployed.


Service Deployment (Windows Only)

You can deploy 1Integrate as a service through additional configuration and by running two scripts supplied in the installation package.

JAVA_HOME

Before running your start-up scripts, you must first ensure your deployment is configured to refer to the correct version of Java. For the Java version supported for your release of 1Integrate, please refer to the release notes.

Once you are sure you have the correct version of Java installed, you will need to ensure that the JAVA_HOME variable is defined.

 **Note:** If the JAVA_HOME variable is already defined on your system with the correct version of Java, then you do not need to do anything.

If the JAVA_HOME environment variable is not set, you have the following options:

- Set the JAVA_HOME environment variable for the user running 1Integrate to point to the correct version of Java.
- Follow the steps below to set the JAVA_HOME environment variable from the 1Integrate service start-up scripts.

To set the JAVA_HOME environment variable in the start-up scripts, carry out the following steps:

1. Edit **start-Interface-Service.cmd** and **install-Engine-Service.cmd**.
2. Locate the following lines:

```
@REM Optionally uncomment and set this path to be  
the correct for your environment  
REM SET "JAVA_HOME=C:\PROGRA~1\Java\[jdk_version]"
```

3. Uncomment the second line and remove "REM", so that it begins with "SET". Ensure that it is pointing to the location of Java home for the correct version of Java installed as part of the pre-requisites. The resulting line should appear as follows:

```
SET "JAVA_HOME=C:\PROGRA~1\Java\[jdk_version]"
```


4. Once configured, run the scripts to start the 1Integrate components.

Using Specific User Accounts

By default, the services run as the local system user. To use a specific user account you will need to run the **start-Interface-Service.cmd** and **install-Engine-Service.cmd** with the following parameters:


```
start-Interface-Service.cmd <domain>\<username>  
<password>
```

```
start-Engine-Service.cmd <domain>\<username>  
<password>
```

 **Note:** Using a named user account is required to use AzureAD authentication for SQL Server connections made by 1Integrate

Deploying the Service

Once you have checked your JAVA_HOME variable, you can deploy 1Integrate as a service by running the following scripts:

 **Note:** Both scripts must be run by an admin user account.

install-Interface-Service.cmd


The interface script will create a service with the name "**1INTService**" and the display name "**1Integrate Interface [version]**".

install-Engine-Service.cmd


The engine script will automatically create *N* services with the name "**1INTEngine[N]Service**" and the display name "**1Integrate Engine [version] # [N]**" based on how many engines have already been configured.

Configure additional engines (Service)

1. Create a copy of the **server-engine** folder and rename it **server-engine2** etc.

 **Note:** You must suffix server-engine with an integer.

2. Inside the cloned directory, open the **configuration\engine.conf.bat**
3. Update SET "ENGINE_NUMBER=1" to the corresponding number e.g
SET "ENGINE_NUMBER=2"
4. You are now ready to start the **install-Engine-Service.cmd** script, which will deploy all configured engines.

 **Note:** Along with the installation there are two uninstallation scripts. You will not need to define JAVA_HOME, as per the installation script. Uninstallation must be carried out prior to an upgrade, using the scripts packaged with the current installation, not the upgrade.

You can label engines you are deploying to group them, and control which Sessions run on different groups.

WINDOWS

1. Open the **server-engine** directory for the engine you will be applying labels to.
2. Within the **configuration** directory, open the **engine.conf.bat** file.
3. Locate and uncomment the following lines:


```
SET "engine.labels="
```

4. Add engine labels as a comma separated list:

```
SET "engine.labels=label1,label2,label3"
```


5. Repeat the steps for each engine being deployed.

5 Testing the Installation

 **Note:** Empty your browser cache before testing your installation.

1Integrate can be accessed through the following site:


[http://\[machine\]:\[service_port\]/1Integrate](http://[machine]:[service_port]/1Integrate)


 **Note:** Use the port number specified during installation, by default this is 8080.

Log in as a user with administrator permissions.

To verify the engine installations, click the **Admin** tab and check that the Grid Topology matches the number of interfaces and engines installed.


6 Upgrading an Installation

 **Note:** Before performing an upgrade to an existing installation, ensure **all sessions are stopped** and perform a **backup** of your repository.

 **Note:** These instructions apply when upgrading from one version to the immediately subsequent release only (e.g. from 1.1 to 1.2). If you are performing an upgrade from any older version, please consult your release notes or contact 1Spatial Support.

UPGRADE AN INSTALLATION

Upgrading an installation consists of unpacking your new release folder, copying across your existing settings files, and editing the Java location in your interface and engine component files.


 **Note:** The following instructions use an example where currently version 1.A is installed, and we want to upgrade to 1.B. We have used C:\1Spatial as a directory within which to store our installation package files, organised into C:\1Spatial\Product-1.A and C:\1Spatial\Product-1.B.

1. Unzip your new Release Package, and the 1Integrate folder within it.
2. Copy the following properties files from the **SETTINGS** folder in your *existing* installation directory (e.g. C:\Program Files\[1Integrate_Directory]\SETTINGS, into the **SETTINGS** folder in your *new* installation directory:
 - **settings.properties**
 - **roles.properties**
 - **users.properties**
3. Copy any custom extensions such as built-ins or data stores (.jar files) from your *existing* **EXTENSIONS** directory into the corresponding **EXTENSIONS** folder within your new installation directory. If you have previously defined a directory containing custom extensions, ensure the `1integrate.extension.dir` parameter in your new **settings.properties** file points to the correct location.
4. Before running your start-up scripts, you must first ensure your deployment is configured to refer to the correct version of Java. For the Java version

supported for your release of 1Integrate, please refer to the release notes.

JAVA_HOME

Once you are sure you have the correct version of Java installed, you will need to ensure that the JAVA_HOME environment variable points to this for the user running 1Integrate.

 **Note:** If the JAVA_HOME variable is already defined on your system with the correct version of Java, then you do not need to do anything.

If the JAVA_HOME environment variable is not set, you have the following options:

- Set the JAVA_HOME environment variable for the user running 1Integrate to point to the correct version of Java.
- Follow the steps below for Windows and Linux to set JAVA_HOME from the 1Integrate start-up script.

To set the JAVA_HOME environment variable in the start-up scripts, carry out the following steps:

WINDOWS

1. Edit **start-Interface.cmd** and **start-Engine.cmd**.
2. Locate the following lines:

```
@REM optionally uncomment and set this path to be  
the correct for your environment  
REM SET "JAVA_HOME=C:\PROGRA~1\Java\[jdk_version]"
```

3. Uncomment the second line by removing "REM", so that it begins with "SET". Ensure that it is pointing to the location of Java home for the correct version of Java installed as part of the pre-requisites. The resulting line should appear as follows:

```
SET "JAVA_HOME=C:\PROGRA~1\Java\[jdk_version]"
```

4. Once configured, run the scripts to start the 1Integrate components.

LINUX

1. Edit **start-Interface.sh** and **start-Engine.sh**.
2. Uncomment the line just below “#Optionally uncomment and set this path to be the correct for your environment”:

```
export JAVA_HOME="/opt/jdk"
```

3. Once configured, run the scripts to start the 1Integrate components.