1Integrate

Installation Guide

WildFly Application Server

Product version: v 2.7

Document version: v 1.8

Document date: 20/11/2020

US Patent Number 9542416 B2 (2017-01-10)

1Spatial
Tennyson House
Cambridge Business Park
Cambridge
CB4 0WZ
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

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.

Licenses

1Integrate licences will be issued via email.

1Spatial 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 "System Requirements" below (including the version of Java required)
- "Microsoft Visual C++ Redistributable (Windows only)" on the next page
- "Configuring the Database Server" on the next page
- "ICU Libraries (Linux only)" on page 9
- "Open Motif Libraries (Linux only)" on page 10
- "Loading Data Formats using FME" on page 28

System Requirements

Please see the *1Integrate Release Notes* for specific system requirements for this release.

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:

- 1 core for interface
- 1 core per session queue
- 1 core 'spare'

This is on top of the requirements for your chosen operating system and any databases running on the same machines.

CPU

For CPUs you will require a minimum of 2gb per session queue, but to ensure best operation we recommend 4gb.

1Integrate is designed so that many features are not memory bound, instead swapping to disk when possible. Only the positional shifting built-ins and the network connectivity built-ins are memory-bound functions and thus disk speed and size are also important.

Disk space is used to store data by each session queue and the amount of disk space can vary greatly dependent on operation.

Example:

4 session queues on one machine, each doing national load, will need significantly more disk space than a number of sessions loading small amounts of data or sessions spread across multiple machines.

Storage

The amount of space required for storage can be estimated by the space used to store data in Oracle, or alternative file formats, this will give you the right order of magnitude. You will also require some space for the repository, but this will only be a significant proportion if you are creating millions of non-conformances. We recommend solid state disks (SSDs) for all storage to ensure data can be accessed quickly and effectively.

Example: For 2 session queues we would recommend:

- A minimum of 4 cores if there is no database on the machine, but ideally 8 or more
- No less than 8 cores if sharing with a database and other services
- At least 6gb memory, but ideally more if there is a database or other server present
- Enough disk space for all the data to be concurrently loaded
- Disk space for non-conformance reports

Microsoft Visual C++ Redistributable (Windows only)

Microsoft Visual C++ 2013 64-bit Redistributable packages are required for Windows installations of 1Integrate.

These can be downloaded from the Microsoft website (**vcredist_x64.exe**).

Configuring the Database Server

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

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

Note: If creating a SQL Server database, follow the process through Microsoft SQL Server Management Studio (MSSMS). The name should be "1Integrate" and all other defaults should be used.

Creating a Database User

A database user is required for the 1Integrate repository.

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

Create an Oracle Database User:

To create a database user, run the following SQL*Plus commands and specify the location of the datafile (including the full file name and extension) as recommended by your Database Administrator:

```
Prompt:> sqlplus [your_system_user_name]/[your_
password]@[tnsname]

SQL> create tablespace [repository_tablespace_name]
datafile '[datafile_location]' size 10m reuse
autoextend on next 10m maxsize unlimited;

SQL> create user [repository_username] identified
by [repository_password] default tablespace
[repository_tablespace_name];

SQL> GRANT create procedure, create sequence,
create session, create table, create view, create
trigger, unlimited tablespace, create type TO
[repository_username];

SQL> ALTER user [repository_username] quota
unlimited on [repository_tablespace_name];
```

Create a SQL Server Database User:

1. Create a new user following the **New Login** process in MSSMS.
2. Set the login name to `rsuser`.
3. Select **SQL Server Authentication**, set the password to `rsuser` and de-select **Enforce Password Policy**.
4. Set the default database to 1Integrate.
5. In **User Mapping**, select 1Integrate.
6. Specify the user as `rsuser` and select all database role memberships except `db_denydatareader` and `db_denydatawriter`.

7. In the MSSMS, select the root node of SQL Server and change the security authentication to **SQL Server and Windows Authentication mode**.
8. To grant permissions to ruser, right click on the root node of the SQL server. Then on the **Permissions** tab, grant all required permissions to ruser.

> **Note:** To allow the application server to log in to the database, the TCP/IP protocol must be enabled. You can enable this in the SQL Server Configuration Manager in the SQL Server Network Communication section.

Create a PostgreSQL User:

Create a user with the default properties.

When creating your repository, ensure your new user is the owner of this database. This will ensure the user has all required privileges.

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 `ld_config` 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 `ld_config` 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
```

Open Motif Libraries (Linux only)

The 64-bit Open Motif libraries are a prerequisite for Linux installations of 1Integrate.

The required Open Motif libraries are as follows:

- libX11.so.6
- libm.so.6
- libpthread.so.0
- libnsl.so.1
- libdl.so.2
- libc.so.6
- libcrypt.so.1
- libXau.so.6
- libXdmcp.so.6

Data Store Prerequisites

1Integrate supports the following types of data stores for input (reading) and output (writing), either as standard or using FME Desktop (which requires a licence). Pay particular attention to the prerequisites attached to certain data store types:

> **Note:** All formats listed below as using FME Desktop require FME Desktop Professional Edition, unless otherwise indicated.

Data Store Type	As standard		Using FME Desktop	
	Read	Write	Read	Write
Bentley Microstation Design (V8)	✗	✗	✓	✓

Data Store Type	As standard		Using FME Desktop	
	Read	Write	Read	Write
Comma Separated Value (CSV)	✗	✗	✓	✓
Autodesk AutoCAD DWG/DXF	✗	✗	✓	✓
Esri ArcGIS Server	✓	✓	✓	✓
Esri Enterprise Geodatabase	✗	✗	✓[1]	✓[2]
Esri File Geodatabase	✓	✗	✓	✓
Esri Shapefile	✓	✓	✓	✓
MapInfo Tab	✓	✓	✓	✓
Microsoft SQL Server Spatial	✗	✗	✓[3]	✓[4]
Oracle	✓	✓	✗	✗
PostGIS	✓	✓	✓	✓
Google BigQuery	✓	✓	✗	✗

[1] This format is only available on Windows. Requires FME Desktop (64-bit) Esri Edition, and ArcGIS Desktop, ArcGIS Desktop Background Processing (64-bit) and Database client (64-bit).

[2] This format is only available on Windows. Requires FME Desktop (64-bit) Esri Edition, and ArcGIS Desktop, ArcGIS Desktop Background Processing (64-bit) and Database client (64-bit).

[3] This format is only available on Windows. The Microsoft SQL Server 2012 Native Client must be installed.

[4] Windows only. Requires FME Desktop Database Edition. The Microsoft SQL Server 2012 Native Client must be installed.

3 Installing 1Integrate on WildFly

Note: Before proceeding, ensure you have completed all pre-requisite steps (see "Prerequisites" on page 6).

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

The following instructions describe how to configure an interface and a session queue. 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 WildFly directory

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

In the WildFly installation, the **standalone** directory contains the configuration for the 1Integrate interface and the **standalone-engine** directory contains the configuration for the session queue.

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\1Spatial\1Integrate\1Integrate-[version]-wildfly\SETTINGS).

Warning: References to`sessionqueue` are being updated to `engine`. When copying the **settings.properties** file to a new installation during an upgrade, ensure these references are updated. For more information, please see the **Release Notes** within your installation package.

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).

Parameter	Description
Repository Database Connection Details	
repository.driver	The database type (oracle, sqlserver or postgresql).

Parameter	Description

repository.url	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).
	This is in the format:
	- for Oracle: `jdbc:oracle:thin:@[PORT]:[PORT]/[Service]`
	Or
	`jdbc:oracle:thin:@[PORT]:[PORT]:[SID]`
	- for MS SQL: `jdbc:sqlserver://[serverName[\\instanceName][:portNumber]];databaseName=1Integrate`
	for MS SQL with Windows Authentication: `repository.url=jdbc:sqlserver://[serverName[\\instanceName][:portNumber]];integratedSecurity=true;databaseName=1Integrate`
	Note: You must define either a `serverName` or `instanceName` See the Microsoft documentation for further guidance.
	- for PostgreSQL:
	- To use the *public schema* to store the repository:
	`jdbc:postgresql://[URL_of_database]/[database_name]`
	- To use a *specified schema* to store the repository:
	`jdbc:postgresql://[HOSTNAME]:[PORT]/[DATABASE]?currentSchema=[REPOSITORYSCHEMA]`

Parameter	Description
repository.username	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). **Note:** Ensure the selected user has the required privileges to access the database.
repository.password	Password for the user described by repository.username.
repository.valid-sql	Add the validation parameter to check for database restart. **For SQL Server or PostgreSQL:** repository.valid-sql=select 1 **For Oracle:** repository.valid-sql=select 1 from dual **Note:** Further changes are required in the **standalone.xml** to auto reconnect to restarted databases.
Licence file Location	
ls_license	The location and name of the provided product licence file. **Note:** The licence file is provided by 1Spatial Support and can be placed in any folder.
Location for Temporary Files	
temp.directory	This is a temporary folder location for files that are eventually stored in the database. For Linux, it is recommended that this is set to `/tmp` or `/var/tmp`. For Windows, it is recommended that this is set to `C:/WINDOWS/TEMP`.

Parameter	Description
Location for the Cache Directory	
cache.directory	The location of the cache directory. When a session is run, a folder is created called "1Integratecache", within which the cache is stored. **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. 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`.
WildFly Server Port Settings	
interface.http.port	The port to use for the interface. **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.
engine.http.port	The port to use for the session queue. **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.

Parameter	Description
Additional Settings	
studiotask.countTotalObjects	If set to true, a progress bar is displayed in the user interface during open data tasks to allow the user to see what proportion of the total objects have been loaded so far. This reduces performance for an open data task. If set to false, only the number of objects currently loaded is shown and the user is not shown the progress. This improves performance for an open data task, particularly when loading data from non-database sources. **Note:** Counting the total number of target objects in order to calculate the progress may take excessive time to complete. It is recommended that this parameter is set to **false** for production deployment or when running sessions via web services.
grid.local.address	(optional) See "NIC/Network Adaptor Configuration" on the facing page.
grid.communication.tcp.port	

Configure Login Banner

Optionally, you can configure a login banner to appear at the top of the Login page:

1. First, you must create a valid HTML file with plain text and optional heading tags. Text will be displayed as black.
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=[Location]/[File].html` **Note:** The HTML page must be placed on the same server as the one running the Interface (s).

NIC/Network Adaptor Configuration

The Grid discovery used to find session queues 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

It is possible to configure your WildFly deployment of 1Integrateto reconnect to repositories on a database restart.

> **Note:** You will need to make additions to the **settings.properties** file (for more, see the Configuring System Properties table and the `repository.valid-sql` parameter.

Once you have configured the **settings.properties** file you will also need to update the two versions of the **standalone.xml** in your **standalone** and **standalone-engine** folders of the1Integrate directory.

Place the following inside the data source validation tags for `jdbc/Repository` and `jdbc/RepositoryLocal`:

```
<check-valid-connection-sql>${repository.valid-
sql}</check-valid-connection-sql>
```

```
<validate-on-match>false</validate-on-match>
<background-validation>true</background-validation>
<background-validation-millis>10000</background-
validation-millis>
```

Configuring Users and Roles

Users and Roles can be edited within 1Integrate.

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

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:

- As a minimum, on installation, change all passwords from the default to unique values.
- change the user names to ones relevant to your organisation.

For stronger security and management, consider using other authentication mechanisms such as using your organisation's Lightweight Directory Access Protocol (LDAP) Service e.g. Microsoft Active Directory. This ensures that passwords and usernames are not stored in the application server but managed, as normal, by an IT department.

Roles

The following roles are available in 1Integrate:

Role	Description
rs_admins	The administrator can set up system parameters and has all the privileges of the other roles. These include creating and modifying: ▶ rules ▶ data stores ▶ sessions ▶ actions ▶ action maps The administrator can also define sessions and upload files.

Role	Description
rs_data_ engineers	A data engineer can: - create and modify actions - create and modify action maps - define and run sessions
rs_data_ loaders	A data loader can upload files into 1Integrate on the Data Stores page only.
rs_data_ quality_ stewards	A data quality steward can define and run sessions only.
rs_rule_ definers	A rule definer can create and modify: - rules - data stores - sessions The rule definer can also run sessions.
rs_users	A user can only view data presented on the interface.
rswsuser	A web service user can use the web services.

Users

The following users are and roles are created by default upon installation:

Username	Password	Assigned roles
INTFull	integrate1	rs_users rswsuser rs_rule_definers rs_data_loaders rs_data_engineers rs_data_quality_stewards
INTAdmin	integrate101	rs_admins rs_users rswsuser
INTUser	integrate102	rs_users

Username	Password	Assigned roles
INTWService	integrate103	rswsuser
INTRule	integrate104	rs_rule_definers
INTLoader	integrate105	rs_data_loaders
INTEng	integrate106	rs_data_engineers
INTQuality	integrate107	rs_data_quality_stewards

WildFly Users

To configure users and roles, navigate to the **\wildfly-[version]\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 following table are included as default.

- **roles.properties** contains a mapping from user names to 1Integrate roles in the form `username=role1,role2,role3`

LDAP

For stronger security and management, Consider using other authentication and authorisation mechanisms such as your organisation's Lightweight Directory Access Protocol (LDAP) Service e.g. Microsoft Active Directory. This ensures that passwords and usernames are not stored in the application server but managed, as normal, by an IT department.

Authenticate using LDAP

The default WildFly configuration of storing passwords as plain text is not a recommended for production use. To configure your organisations LDAP service in WildFly, perform the following configuration:

Configure an LDAP service:

You will need to amend the `standalone.xml` file, to locate this go to: 1Integrate_home\standalone\configuration.

Use **LdapExtLoginModule** to defer authentication (usernames and passwords) and optionally authorisation (roles) to your existing LDAP server.

You will need to substitute the default "1SMS" security domain section, depending on the level of authentication or authorisation you would like from your active directory.

> **Note:** Example LDAP configurations have been provided below, but you will need to populate the relevant with configurations specific to your organisation.

The following extract is the default "1SMS" which is to be replaced::

```
<security-domain name="1SMS" cache-type="default">
    <authentication>
        <login-module flag="required"
code="UsersRoles">
            <module-option
value="${jboss.home.dir}/SETTINGS/users.properties"
name="usersProperties"/>
            <module-option
value="${jboss.home.dir}/SETTINGS/roles.properties"
name="rolesProperties"/>
        </login-module>
    </authentication>
</security-domain>]
```

Mapping LDAP groups to 1Integrate roles:

It is possible to configure authentication so that usernames and passwords are managed by LDAP, but the authorisation (roles assigned to specific LDAP groups) is managed within the configuration files. This provides high security but removes the need for the LDAP service to be told about 1Integrate roles.

> **Note:** You can only map **LDAP Groups** to roles if using this method.

To achieve this, configure the `standalone.xml` file in the following way (the section highlighted in bold shows the addition when compared to a deployment in which you will be managing authorisation *and* authentication in your LDAP):

```
<security-domain name="1SMS" cache-type="default">
    <authentication>
        <login-module
code="org.jboss.security.auth.spi.LdapExtLoginModul
e" flag="optional">
```

```
              <module-option
name="java.naming.factory.initial"
value="com.sun.jndi.ldap.LdapCtxFactory"/>
              <module-option
name="java.naming.provider.url"
value="ldap://localhost:389"/>
              <module-option
name="java.naming.security.authentication"
value="simple"/>
              <module-option name="bindDN" value="
[user]"/>
              <module-option name="bindCredential"
value="[password]"/>
              <module-option name="baseCtxDN"
value="
[ou=users,ou=yourgroup,dc=yourcompany,dc=local]"/>
              <module-option name="baseFilter"
value="(cn={0})"/>
              <module-option name="rolesCtxDN"
value="cn=users,dc=yourcompanyl,dc=local"/>
              <module-option name="roleFilter"
value="(member={1})"/>
              <module-option
name="roleAttributeIsDN" value="true"/>
              <module-option name="roleAttributeID"
value="cn"/>
              <module-option
name="roleNameAttributeID" value="cn"/>
              <module-option name="roleRecursion"
value="0"/>
              <module-option name="searchScope"
value="SUBTREE_SCOPE"/>
              <module-option name="searchTimeLimit"
value="5000"/>
              <module-option
name="java.naming.referral" value="follow"/>
          </login-module>
          <login-module
code="org.jboss.security.auth.spi.RoleMappingLoginM
odule" flag="required">
              <module-option name="rolesProperties"
value="${jboss.home.dir}/SETTINGS/roles.propertie
s"/>
              <module-option name="replaceRole"
value="true"/>
```

```
          </login-module>
     </authentication>
</security-domain>
```

Note: In order authorise in LDAP and authenticate in WildFly, the `roles.properties` file must be populated with the **groups** that match those from the LDAP directory with the associated role(s). e.g. LDAPGroup = rs_admins.

Note: LDAP Group names with spaces are not recognised.

Example of the `roles.properties` file which is mapping LDAP groups to the relevant roles:

```
ADMINISTRATORS= rs_admins,rs_users,rswsuser,rs_
rule_definers,rs_data_loaders,rs_data_engineers
ENGINEERS=rs_data_engineers
WORKERS=rs_users,rs_rule_definers
```

Manage Users, Groups and Roles in LDAP:

If you would like to manage authentication and authorisation in your LDAP server, use the following in place of the default:

```
<security-domain name="1SMS" cache-type="default">
     <authentication>
          <login-module
code="org.jboss.security.auth.spi.LdapExtLoginModul
e" flag="optional">
               <module-option
name="java.naming.factory.initial"
value="com.sun.jndi.ldap.LdapCtxFactory"/>
               <module-option
name="java.naming.provider.url"
value="ldap://localhost:389"/>
               <module-option
name="java.naming.security.authentication"
value="simple"/>
               <module-option name="bindDN" value="
[user]"/>
               <module-option name="bindCredential"
value="[password]"/>
```

```
              <module-option name="baseCtxDN"
value="
[ou=users,ou=yourgroup,dc=yourcompany,dc=local]"/>
              <module-option name="baseFilter"
value="(cn={0})"/>
              <module-option name="rolesCtxDN"
value="cn=users,dc=yourcompanyl,dc=local"/>
              <module-option name="roleFilter"
value="(member={1})"/>
              <module-option
name="roleAttributeIsDN" value="true"/>
              <module-option name="roleAttributeID"
value="cn"/>
              <module-option
name="roleNameAttributeID" value="cn"/>
              <module-option name="roleRecursion"
value="0"/>
              <module-option name="searchScope"
value="SUBTREE_SCOPE"/>
              <module-option name="searchTimeLimit"
value="5000"/>
              <module-option
name="java.naming.referral" value="follow"/>
          </login-module>
     </authentication>
</security-domain>
```

Starting the Deployment

Before opening the component files, they must first be configured to refer to the correct version of Java.

The procedure is different for Windows and Linux.

Windows:

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

```
@REM Uncomment and set this path to be 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, open the files to start the 1Integrate components.

Linux:

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

   ```
   export JAVA_HOME="/usr/java/[jdk_version]"
   ```

3. Once configured, open the files to start the 1Integrate components.

Deploying 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 **standalone-engine** folder to a higher level directory, rename it, and paste it back into the **wildfly-[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 **standalone-engine** folder and rename it **standalone-engine2**.

2. Create a copy of the **start-Engine.cmd** file and rename it **start-Engine2.cmd**.
3. Edit this file to refer to **standalone-engine2** instead of **standalone-engine** (or standalone-engine3 for the third engine etc.) and change the port offset at the end of the WildFly `CALL` line from 100 to 200 (or 300 for the third engine etc.).

   Note: For example, the line

   ```
    CALL bin\standalone.bat -
   P=SETTINGS\settings.properties -
   Djboss.server.base.dir=standalone-engine-
   Djboss.socket.binding.port-offset=100
   ```

   would be changed to

   ```
   CALL bin\standalone.bat -
   P=SETTINGS\settings.properties -
   Djboss.server.base.dir=standalone-engine2 -
   Djboss.socket.binding.port-offset=200.
   ```

Linux:

1. Create a copy of the **standalone-engine** folder and rename it **standalone-engine2**.
2. Create a copy of the **start-Engine.sh** file and rename it **start-Engine2.sh**.

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

   Note: For example, the line

   ```
   ./bin/standalone.sh -
   P=SETTINGS/settings.properties -
   Djboss.server.base.dir=standalone-engine
   =Djboss.socket.binding.port-offet=100
   ```

   would be changed to

   ```
   ./bin/standalone.sh -
   P=SETTINGS/settings.properties -
   Djboss.server.base.dir=standalone-engine2
   =Djboss.socket.binding.port-offet=200.
   ```

Installing as a Service (Windows Only)

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

The two scripts to run are:

- **install-Interface-Service.cmd**
- **install-Engine-Service.cmd**

The **install-Interface-Service.cmd** script has a `JAVA_HOME` parameter that must be set to the correct Java installation directory.

The engine script has an `ENGINE_COUNT` parameter which indicates the number of Session Queue services to create during the install.

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

When configured as a Windows service, 1Integrate will start automatically whenever Windows starts. The 1Integrate service can be controlled via the Windows Services admin application.

> **Note:** Along with the installation there are two uninstallation scripts. It is recommended that you set the `ENGINE_COUNT` to match that of the install script when performing an uninstallation. You will not need to define `JAVA_HOME`, as per the installation script.

Loading Data Formats using FME

1Integrate can handle spatial data from a number of sources such as an Oracle database, Esri Shapefiles or MapInfo Tab files.

> **Note:** For more details on the formats supported, please see the 1Integrate online help.

To access other formats such as DWG files or Esri Enterprise geodatabases, 1Integrate uses functionality from Safe Software's FME. To use this capability, you must have FME Desktop installed and available to 1Integrate, with an FME Desktop license available for use.

> **Note:** In order to be editable, tables must have a primary key defined. You can read data from tables that do not have primary keys, but you will not be able to write data back to those tables.

Note: For PostGIS, only tables with at least one geometry column are supported.

To allow 1Integrate to access data that is only available via an FME license, use the following procedure, depending on your operating system:

Configure FME on Windows

Note: The following steps are only necessary if FME has not already been added to the system or user's PATH environment variable in Windows. However, when using WildFly with AdoptOpenJDK ensure the `plugins` path is defined, as detailed below.

On Windows, set the FME_HOME directory to the path in **bin\standalone.conf.bat** by adding the following lines to the top of the file:

Note: The .bat file is located in the 1Integrate directory e.g. `\1Integrate[version]\wildfly[version]\bin`

Using AdoptOpenJDK

Note: As of 1Integrate 2.7, 1Integrate only supports AdoptOpenJDK on WildFly installations. Please see the Release Notes for additional information.

```
@rem add FME to PATH
set PATH=[FME path];[FME path]/plugins;%PATH%
```

Where `[FME path]` is the location where FME is installed (e.g. C:\PROGRA~1\FME).

Configure FME on Linux

Set the FME_CORE directory to the LD_LIBRARY_PATH in **bin/standalone.conf** by adding the following lines to the top of the file:

```
# add FME to LD_LIBRARY_PATH
export LD_LIBRARY_PATH="[FME_CORE_PATH]:$LD_
LIBRARY_PATH"
```

Where `[FME_CORE_PATH]` is the location of the fme/fmecore/ directory.

Offline Help

By default, clicking the Help button within 1Integrate opens the online WebHelp. However, this button can be configured to open local offline documentation instead.

Note: The configuration of offline help is only advised for environments without access to the internet, as local help files will not receive updates as often as the online help.

Before performing the configuration, ensure your local help files are places somewhere accessible on your local network. If the help files have not been provided with your release, please contact your vendor or 1Spatial Support.

Configure Offline Help :

1. Open the **settings.properties** file (located within C:\Program Files\1Spatial\1Integrate\1Integrate-[version]_wildfly\SETTINGS).
2. Edit the value for the **help_base_url** parameter to the location of your local help files.

   ```
   help_base_url=[location of help files]
   ```

   Where `[location of help files]` is replaced with the path to your local help files.

4 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

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

Log in as a user with administrator permissions.

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

If you need to access the 1Integrate Web Services API, then a web page with documentation and a link to the WSDL document can be found at:
http://[machine]:[service_port]/soap

5 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 on Wildfly consists of unpacking your new release folder, copying across your existing settings files, and editing the Java location in your interface and session queue 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 Wildfly folder within it (e.g. **C:\Program Files\1Spatial\Product-1.B\Product-1.B_wildfly\wildfly-[version]**).
2. Copy the following properties files from the **SETTINGS** folder in your *existing* installation directory (e.g. **C:\Program Files\1Spatial\Product-1.A\Product-1.A_wildfly\wildfly-[version]\SETTINGS**, into the **SETTINGS** folder in your *new* installation directory:
   - **settings.properties**
   - **roles.properties**
   - **users.properties**
3. Copy any built-ins and custom data stores (**.jar** files) from your *existing* installation directory (e.g. **C:\Program Files\1Spatial\Product-1.A\Product-1.A_wildfly\wildfly-[version]\standalone\deployments\Product-1.A.ear\APP-INF\lib** and

C:\Program Files\1Spatial\Product-1.A\Product-1.A_wildfly\wildfly-[version]\standalone-sessionqueue\deployments\Product-1.A.ear\APP-INF\lib) into the corresponding **lib** folders within your new installation directory.

4. Before opening the component files, they must first be configured to refer to the correct version of Java.

   The procedure is different for Windows and Linux.

   Windows:

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

      ```
      @REM Uncomment and set this path to be
      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, open the files to start the 1Integrate components.

   Linux:

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

      ```
      export JAVA_HOME="/usr/java/[jdk_version]"
      ```

   3. Once configured, open the files to start the 1Integrate components.