



1Edit

Installation Guide

Product Version: v 3.5

Document Version: v 3.2

Document Date: 10/07/2024

Copyright 2024 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

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 Installation Pre-requisites	1
Side-loading Applications	1
Enable side-loading of trusted applications	1
Visual C++ Re-distributable Packages	1
2 Installation and Upgrade	2
Upgrading 1Edit	3
3 Configuration	4
Project Template File	4
Installing Sample Projects	5
Integration with 1Spatial Management Suite Worklist Service	5
Configuration file format	6
Distribution	8
Deploying the Config file	8
4 Troubleshooting	9
When Installing 1Edit	9
When Running 1Edit	9


1 Installation Pre-requisites

Before installing 1Edit, check that the following pre-requisites have been fulfilled.

Please see the *1Edit Release Notes* for specific hardware and system requirements.

Side-loading Applications

Because 1Edit is installed as a side-loaded application (not from the Microsoft app store) then your operating system must be licensed and enabled for side-loading.

 **Note:** For more information on side-loading applications within Windows, please see the official [Microsoft documentation](#).

Enable side-loading of trusted applications

Before installing, every machine must have set the **Allow all trusted applications to install** Group Policy setting, which your IT department can help you with.

If your IT department doesn't use a group policy, then each machine can be configured to allow side loading.

Allow Side loading on Windows 10:

1. Open the Windows **Settings** on your device
2. Click **Update & Security > For developers**
3. Ensure **Developer mode** is on.


Visual C++ Re-distributable Packages

Some machines require the free Microsoft® Visual C++ 2013 Re-distributable Packages to be installed.

If the application installs, but closes when trying to open a project then try installing these packages.

You can download them [directly from Microsoft](#).


2 Installation and Upgrade

 **Note:** Due to the way that Windows® 10 manages applications, you will need to perform the following steps for each user who will be logging in and using 1Edit on the machine.

A single batch file (**.bat**) is used to install 1Edit. This file performs installation by "side-loading" the software.

INSTALL 1EDIT

1. Log in to the machine as the user who will run 1Edit.
2. Copy the installation folder to a temporary location on the local machine, or access it from a mapped network drive.

 **Note:** Security settings typically prevent you from installing directly from the C: drive, so place the installation folder within a sub-folder.

3. If the folder is zipped, unzip it now. You cannot install from a zipped folder.
4. Navigate to the unzipped folder and double-click on **install1edit.bat**.
5. 1Edit will install automatically, messages in the window will indicate success or failure.

If your administrator has included in the app package a predefined 1Edit Template and config file then these will automatically get deployed too (see ["Integration with 1Spatial Management Suite Worklist Service" on page 5](#)).


You can delete the temporary installation folder now.

Upgrading 1Edit

Depending on the version of 1Edit you are upgrading, you will need to perform different processes.

UPGRADING VERSION 3.0 AND ABOVE


If your current installation is version 3.0 or later, then the installer will automatically upgrade 1Edit and retain any existing projects, templates and configuration files used by the application.

 **Note:** Do not uninstall the previous version of 1Edit, as you will lose any existing projects and will need to re-deploy the additional configuration files.

UPGRADING VERSIONS BEFORE 3.0

If your current installation of 1Edit is before 3.0, you will need to carry out the following steps:

1. Install your new version of 1Edit, following the **Install 1Edit** steps above.
2. Copy configuration **templates**, **projects** and **gmlSource** from your old installation to your new installation.

 **Example:** Copying resources for the updgrade

1. Locate the following folder:
`C:\Users\<username>\AppData\Local\Packages\1Edit_65g7xm77fxxm8\localState\...`
2.
 - a. For 1Edit versions up to and including 3.2.0: Copy the configuration templates, projects and gmlSource to:
`C:\Users\<username>\AppData\Local\Packages\1Spatial.1Edit_65g7xm77fxxm8\localState\...`
 - b. For 1Edit versions from 3.2.1 onwards: Copy the configuration templates, projects and gmlSource to:
`C:\Users\<username>\AppData\Local\Packages\1Spatial.1Edit_65g7xm77fxxm8\localState\...`

3. Reapply your required 1Edit settings.
4. Uninstall your old installation.

3 Configuration

Project Template File


A project template defines a standard schema, set of styles, set of validation rules and topology definitions. A template is a useful way of defining a standard configuration into which you can load different data without having to repeatedly load schemas, define styles, configure topology etc.


You can optionally use a template when loading GML into 1Edit, but you *must* have a template deployed in order to download jobs as projects using the 1SMS worklist.

Using 1Edit, create a project containing no data but with the correct extents, SRID and schema and optionally one or more styles, zero or more topologies and zero or more validation rules.


When the project is saved as a template from the projects page, it is saved as a **.ibck** (image backup) file. The name of the file does not matter, but it must have a **.ibck** extension.


You can install this new template from the project page by opening the app bar and selecting **Import Template**. This will allow you to browse to the newly selected template, and replace any current template with the template you have just saved.

 **Note:** Importing a template in this way simply copies it to the following folder:
C:\Users\<current_username>\AppData\Local\Packages\1Spatial.1Edit_65g7xm77fxxm8\LocalState\Templates

 **Note:** 1Edit supports the use of multiple templates and gives you the option of choosing between multiple templates. If multiple templates are installed, then you will be given the choice of templates e.g. when importing a Job.


To distribute template files automatically to all users, they can be placed into the **Template** folder within the installation folder. When users install 1Edit, the template files will be automatically put in the right place.


 **Note:** You can view the Templates being used on a Project on the Project Properties pages. See [Managing Projects](#) for more details.

 **Note:** If you update the Config or Template files after distribution, then the easiest way to distribute the new versions to all users is to replace the files within the 1Edit installation folder and distribute this folder to all users to re-install. 1Edit will not be updated if the version has not changed, but the new files will be deployed into the right location for each user.

Installing Sample Projects

If you have been provided with a sample 1Edit project, then this process is required to put pre-configured project data in the right place.

 **Note:** If you have sample GML data then you do not need this step, just create a new project from within 1Edit, as documented in the 1Edit help.

 **Note:** If you have upgraded from a previous release of 1Edit with the project already in place then there is no need to perform this step. This is only required for sample data after the first install, or if 1Edit has been uninstalled or if the data has changed.

INSTALL SAMPLE PROJECTS


1. Launch 1Edit to create the correct folder structure.
2. Open an Explorer window and copy the provided data folder to
C:\Users\<current_
username>\AppData\Local\Packages\1Spatial.1Edit_
65g7xm77fxxm8\LocalState\Projects
3. Return to the 1Edit project page and press the refresh button if the project is not visible.


Integration with 1Spatial Management Suite Worklist Service

In order to use the 1Edit Worklist page and integrate with the 1Spatial Management Suite (1SMS), 1Edit requires a configuration file defining the location of the 1SMS Worklist service.

If your administrator has bundled this file in the installation package then this does not need to be placed manually. You can check this by looking for a **services.xml** file in:


```
C:\Users\<current_username>\AppData\Local\Packages\1Spatial.1Edit_65g7xm77fxxm8\LocalState\Config
```

 **Note:** You can also check by looking at the projects page to see if there is a **Worklist** button present; this button is only available when the **services.xml** file is in place.


 **Note:** 1SMS also requires a template file, [Templates](#) define the standard schema, styles, rules and topology setup. You must have created and deployed a template file in order to use the 1SMS worklist to download and open jobs.

Configuration file format

An **.xml** configuration file must be provided when connecting to 1SMS. and called **services.xml**, it must follow a specific structure. Examples of this structure can be found in the **Example File Formatting** section below.

EXAMPLE FILE FORMATTING

When connecting to a secured 1SMS 2.3 environment you must include the version attribute and the `<oauthserver>` parameter (as in the example below):

 **Note:** When connecting to an unsecured 2.3 environment the `<oauthserver>` parameter can be omitted.

```
<services>
<SMSEnvironments>
  <SMSEnvironment version="2.3">
    <name>[NAME_EXAMPLE]</name>
    <worklistroot>http://[server]:[port]/wf</worklistroot>
    <oauthserver>http://[server]:[port]/1sms/mstoken</oauthserver>
  </SMSEnvironment>
</SMSEnvironments>
```

```
</services>
```

When connecting to 3.0+ environment both the version and `<oauthserver>` parameter can be omitted:

```
<services>
<SMSEnvironments>
  <SMSEnvironment>
    <name>[LATEST NAME EXAMPLE]</name>
    <worklistroot>http://[server]:[port]/1sms</worklistroot>
  </SMSEnvironment>
</SMSEnvironments>
</services>
```

You can connect to multiple 1SMS environments by utilising the following structure:

```
<services>
<SMSEnvironments>
  <SMSEnvironment version="2.3">
    <name>[NAME_EXAMPLE]</name>
    <worklistroot>http://[server]:[port]/wf</worklistroot>
    <oauthserver>http://[server]:
[port]/1sms/mstoken</oauthserver>
  </SMSEnvironment>
  <SMSEnvironment>
    <name>[NAME_EXAMPLE2]</name>
    <worklistroot>http://[server]:[port]1sms</worklistroot>
```


```
</SMSEnvironment>

</SMSEnvironments>

</services>
```

Distribution


To distribute this file automatically to all users and have it installed for all installations, it can be placed into the **Config** folder within the installation folder. When users install 1Edit, the config file will be automatically put in the right place.

 **Note:** If you update the Config or Template files after distribution, then the easiest way to distribute the new versions to all users is to replace the files within the 1Edit installation folder and distribute this folder to all users to re-install. 1Edit will not be updated if the version has not changed, but the new files will be deployed into the right location for each user.

Deploying the Config file

If the config file was not bundled with your installation, then it must be deployed.

1. Launch 1Edit to create the correct folder structure.
2. Drop the **services.xml** file into the following folder:
C:\Users\<current_username>\AppData\Local\Packages\1Spatial.1Edit_65g7xm77fxxm8\LocalState\Config

 **Note:** Ensure that 'Hidden Items' is checked in the Windows Explorer View tab, in order to see the AppData folder

3. Restart 1Edit.

4 Troubleshooting

When Installing 1Edit

The installation fails with a message in the installation window.

Check that side-loading is enabled (see ["Side-loading Applications" on page 1](#)).

Note that the default local administrator user is not domain joined and does not have side-loading enabled so ensure you are logged in as a "normal" domain user.

The installation fails with the message 'Backup folder C:\Users\<user>\Downloads\1Spatial.1Edit.Backup already exists. Remove before running installer again.'

When installing version 3.2.1 or above of 1Edit and upgrading from a previous 3.x version then a copy of projects and templates etc are placed in the 1Spatial.1Edit.Backup folder. These are restored to the correct place after installing the new version, and this folder is left in place in case there are issues.

You will only get this error if you have performed multiple installs of 3.2.1 or above on the same machine for the same user. Check that the files in 1Spatial.1Edit.Backup are no longer required and delete the folder before trying the install again.

When Running 1Edit

The application does not start up ("This app can't open") and the start screen tile subsequently has a cross in the corner.

This is usually because there is no side-loading license key.

Alternatively you need a ["Side-loading Applications" on page 1](#).

The application starts up, but crashes as soon as a project is opened.

Install the C++ re-distributable on the machine (see ["Visual C++ Re-distributable Packages" on page 1](#)).