1Edit

Installation Guide

Product version: v 2.6

Document version: v 1.0.2

Document date: 25/04/2018

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

Contents

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.

Side-loading licence

Windows 8.1 Enterprise (any version) and Windows 8.1 pro (from Update 1 onwards) include a side-loading licence key if they are domain joined (i.e. you login as a domain user, it doesn't mean you have to be connected to the network at the time).

If you are not using these versions, access to side-loading keys is also available with the following Volume licensing programs:

- ▶ Enterprise Agreement
- ▶ Enterprise Subscription Agreement
- ▶ Enrollment for Education Solutions (under a Campus and School Agreement)
- ▶ School Enrollment
- ▶ Select and Select Plus

Otherwise you will need to purchase a side-loading key from a Microsoft reseller.

If you don't have a side-loading license key as described above, then for evaluating 1Edit in a *non-production* environment you can a use a (free) Microsoft account when installing the software, to allow it to run under a temporary development license.

This will provide a license to evaluate and test side-loaded apps for 30 days.

> 📋 **Note:** Microsoft does not permit you to use this temporary development license in a production environment.

<u>Acquire a temporary developer license:</u>

1. Open Windows PowerShell as administrator.

   To do this press **Start**, type `PowerShell`, right-click (or swipe down) on the **PowerShell** shortcut and select **Run as administrator** at the bottom of the screen.

2. Type the following command into the PowerShell window, then press **Enter**:

   `Show-WindowsDeveloperLicenseRegistration`

3. Read and agree to the licence conditions.

   You will not be able to proceed without accepting the licence conditions.

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.

<u>Allow trusted apps to install:</u>

1. Login as an administrator user.

2. Press **Start**.

3. Type `gpedit.msc`, and press **Enter**.

4. In the Group Policy editor Navigate to

   <u>Computer Configuration\Administrative Templates\Windows Components\App Package Deployment</u>

5. Double-click the **Allow all trusted apps** to install option and set it to **Enabled**.

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 Installing 1Edit

> **Note:** Due to the way that Windows® 8.1 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.

If you have already installed a previous version of 1Edit, then this installer will automatically upgrade 1Edit and retain any existing projects, templates and configuration files used by the application. Do not uninstall the previous version of 1Edit, otherwise you will lose existing projects and will need to re-deploy the additional configuration files.

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 directing 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 10).

You can delete the temporary installation folder now, but If you are using survey tools, then see "Installing the 1Edit Device Extension" on page 8.

3 Installing the 1Edit Device Extension

If using external surveying hardware, then the 1Edit Device Extension must be installed on the machine.

This can be installed by any user, but the installation requires administrator privileges.

Once the device extension is installed as any administrator user, all users who use 1Edit on this machine will be able to connect to supported GPS and TotalStation devices.

Install the 1Edit Device Extension:

1. Log in to the machine as a user with administrator privileges.

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 directing 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 **installDeviceExtension.bat**.

5. The 1Edit Device Extension will install automatically, messages in the window will indicate success or failure.

Testing the installation

Perform the following steps to test the 1Edit Device Extension installation:

1. Check the following files are present in C:\Program Files (x86)\1Spatial\1Edit\devices:

   ▶ AmalgamCOM.dll

   ▶ amalgam-devices-communication.dll

   ▶ amalgam-devices-leicagps.dll

   ▶ amalgam-devices-leicatotalstation.dll

- ▶ amalgam-devices-sharedcomponents.dll

- ▶ amalgam-devices-teststub.dll

- ▶ DeviceProxyStub.dll

- ▶ GeoComS2K.dll

- ▶ BrokeredDeviceComponent.winmd

2. From a command prompt, change the directory to: C:\Program Files (x86)\1Spatial\1Edit\devices

   Run the following command to check it succeeds: `regsvr32.exe DeviceProxyStub.dll`

   Run the following command to check it succeeds: `icacls.exe . /T /grant *S-1-15-2-1:RX`

   If the commands succeed them the device extension is correctly installed.

Uninstalling the 1Edit Device Extension

1. Log in to the machine as a user with administrator privileges.

2. From a command prompt, change the directory to: C:\Program Files (x86)\1Spatial\1Edit\devices

3. Run the following command and check it succeeds: `regsvr32.exe DeviceProxyStub.dll /u`

4. Delete all files in the following directory: C:\Program Files (x86)\1Spatial\1Edit\devices

4 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 config file to define the location of the 1SMS worklist service.

If your administrator has bundled a config file in with the installation then this does not need to be placed manually. You can check this by looking for a **services.xml** file in: C:\Users\[user]\AppData\Local\Packages\1Edit_qw7y987j427n8\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 present when a config file is in place.

1SMS also requires a template file, which defines 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.

Config file

The config file must be called **services.xml** and have a specific structure.

For a 1SMS environment without secure authentication (e.g. when calling from within the same network):

```
<services>
    <environment>Any name here</environment>
    <worklistroot>http://1SMSserver.company.com:port/wf</worklistroot>
</services>
```

For a 1SMS environment with secure OAuth authentication and https:

```
<services>
    <environment>Any name here</environment>
    <worklistroot>https://1SMSserver.company.com:port/wf</worklistroot>
```

```
    <oauthserver>https://1SMSserver.company.com:port/w
f/mstoken</oauthserver>
</services>
```

You will need to set the URLs as appropriate for your 1Workflow worklist service, which may also have been configured with additional https security (in which case the URL starts with https not http). A 1SMS/1Workflow administrator can provide these URLs.

The <environment> tag is an optional element which can be used to specify the name to appear at the top of the worklist page to help tell the user which server they are pointing at, for example 'Live Production Data' or 'Test Environment'.

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\ [user]\AppData\Local\Packages\1Edit_ qw7y987j427n8\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.

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\[user]\AppData\Local\Packages\1Edit_qw7y987j427n8\LocalState\Templates

To distribute a template file automatically to all users, it can be placed into the **Template** folder within the installation folder. When users install 1Edit, the template 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.

5 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\ [user]\AppData\Local\Packages\1Edit_ qw7y987j427n8\LocalState\Projects

3. Return to the 1Edit project page and press the refresh button if the project is not visible.

6 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 4).

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.

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.

If you are on Windows 8.1 pro then typically this means that you should perform a Windows update to at least Update 1.

Alternatively you need a side loading license (see "Side-loading licence" on page 4).

❝ 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 5).

Using the Device Extension

> **The following error occured while trying to apply COM port change:**
> **Failed to open the GPS device connection on COM port: 8**
> **Until the cause of the error is resolved, GPS support will remain disabled**

Try to kill the COM surrogate dllhost process in the task manager and try connecting again.

Closing 1Edit in between projects should minimise this problem.