

1Spatial Management Suite

Administration Guide

Document version: v 1.9

Document date: 21/12/2020



Copyright 2020 1Spatial plc and its affiliates.

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

1Integrate/1Validate: US Patent Number 9542416 B2 (2017-01-10)

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 Introduction	5
Audience	5
Licenses	5
1Spatial product support	5
2 1SMS Administration Overview	6
Architecture	6
Recovery Plan Procedures	7
Backup Recommendations	7
Restart Recommendations	7
1SMS Roles	8
3 1Exchange Configuration	10
FME Server	10
Snowflake	10
URL Locations	10
Security Settings	11
Configuring GO Publisher and GO Loader Projects	12
Configuring GO Publisher	14
Configuring GO Loader	15
Version Enabling Database Data (GO Loader and GO Publisher only)	16
Create Conflict Resolution Name Mappings (GoLoader and GoPublisher only)	17
Exchange Policies	18
URL Locations	18
FME Server	18
GO Loader and GO Publisher	19
1Exchange Product Configuration	19
Managing Exchange Policies	20
Extraction FME Workspace Published Parameters	25
Submission FME Workspace Published Parameters	25
1Exchange Tablespace Maintenance	30
BPEL Audit Trail	30
GO Publisher and 1Exchange LOB Columns	30
4 1Plan Administration	32
Configuring Metadata Fields	32

Mandatory Fields	33
Automatically Generating Job Names	33
Job Types	34
Enumerated Values	35
History Records	36
Archive Jobs	37
Workflow Integration	38
Map Layers	38
Map Configuration	40
Gazetteer	41
User Session Timeouts (WebLogic Only)	43
Tuning the 1Plan Service (WebLogic Only)	44
5 1Transact Administration	45
6 1Validate Administration	46
Validation Rules	46
Uploading Validation Rules	46
Validation Rules	47
Uploading Validation Rules	47
Contextual Datastore	48
7 1Workflow Administration	50
Settings	50
Access	50
Settings Per Job Type	50
Global Settings	57
General	57
Notifications	59
Services	59
	60
Configuring Email Notifications	60
Notifications Settings	61
Notification Recipient Configuration	62
Automatic Conflict Resolution	63
Reconfiguring 1Workflow	64
8 1Generalise Administration	67
Installing System Flowlines	67
Backing up Data	67
Migrating Data	67
Deleting Data	67
Invalidating the repository cache	68

9 1Height Administration	69
Accessing 1Height	69
Configuring 1Height	69
Triangulated Irregular Networks (TIN)	72
Adding a TIN	73
Height quality	73
Heighting policies	74
Heighting Geometry Features	75
Using the 1Height User Interface	76
1Workflow Integration	77
1Height SOAP Web Service	77

1 Introduction

This guide explains how to perform configuration and administration tasks for the 1Spatial Management Suite.

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 provided 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

1Spatial Management Suite licences will be issued via email.

1Spatial product support

If assistance is required during installation, 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 1SMS Administration

Overview

There are several aspects of the 1Spatial Management Suite that can be customised to suit the working practices of your organisation and the specifics of the feature data that will be handled by the system:

- ▶ **1Workflow administration** - controls optional steps in the structured processing of jobs as well as tasks such as notifying users of changes in job status
- ▶ **1Plan administration** - configures the job types and structure of the metadata associated with each job type
- ▶ **1Exchange administration** - defines and customises schema mapping to your system. 1Exchange is responsible for the mapping back and forth between your feature data and GML data that used by the editors.
- ▶ **1Validate administration** - adds validation rules that implement the quality rules for your data
- ▶ **1Transact administration** - manages long transactions in an underlying data storage technology

Architecture

1Spatial Management Suite is deployed on a **Java Platform, Enterprise Edition** application server, using Oracle databases to hold feature and job control data. The overall process flow is controlled by a workflow hosted under Oracle's BPEL workflow manager.

The structure of the database schemas and the installation and configuration of the web services that make up the system are described in the *1Spatial Management Suite Installation Guide*. For most installations, the Web services will be installed on multiple physical servers potentially under a clustered environment.

Each installation of the system requires customisation of the 1Spatial Management Suite system to cater for your expected work practices and your feature data.

Recovery Plan Procedures

1Spatial Management Suite is resilient to service failures, however consideration must be made for complete system failure.

All data (configuration, feature, and job control) is held in the Oracle database, therefore the disaster recovery process requires that the Oracle database is periodically backed up (and in the event of failure, restored) and the components are restarted (see "Restart Recommendations" below).

Backup Recommendations

A single, scalable database platform for both feature and application data greatly simplifies backup and recovery procedures, as all transactional states for the planning and data capture process (including comprehensive workflow state in the BPEL dehydration store) are held in a single self-consistent location.

It is advised that in a recovery situation, a full database recovery is performed to ensure that all components of the 1Spatial Management Suite are in a consistent state.



Note: Partial database recovery is not supported.

The dynamic data for the 1Spatial Management Suite is entirely stored in the database, therefore there are no other data backup requirements. For information about backing up Oracle databases, refer to the Oracle Documentation Library.

Backup and restore should be performed at a database-wide level. This guarantees both consistency between the feature database and the component metadata states, and is also Oracle's recommended safe way for backing up and restoring workspaces (1Spatial Management Suite uses Oracle's Workspace Manager technology).

All persistent data is stored in the database; backup and recovery is restricted purely to managing the feature and component repository database.

Restart Recommendations

In the event of a system restart, it is recommended that all jobs are in a PLANNED, ALLOCATED, or COMPLETED state (see Lifecycle of a Job for more information).

If this is not possible, wait for jobs to reach a static state (for example, wait for activated jobs to get to "prepared" and wait for submitted jobs to get to "invalid", "conflicted", or "completed" state). Any jobs that have been

downloaded and are being edited can be submitted again once the system has restarted.

1SMS Roles

1SMS creates roles for access to different levels of functionality for each product.

Each user who needs to use 1SMS must be allocated the appropriate roles.




Note: This role allocation is typically performed by the IT department.

The list of roles and their corresponding permissions are displayed below.

Interface	Roles						
	1SM S_ admin	exchang e_ user	plann er	transa ct_ user	validat e_ user	work er	global_ supervi sor
1Exchange							
/1exchange	✓	✓	✗	✗	✗	✗	✗
/admin	✓	✗	✗	✗	✗	✗	✗
SOAP	✓	✓	✗	✗	✗	✗	✗
1Plan							
/1plan	✓	✗	✓	✗	✗	✗	✗
/1plan/jobsummary/	✓	✗	✓	✗	✗	✗	✓
/admin	✓	✗	✗	✗	✗	✗	✗
SOAP	✓	✗	✓	✗	✗	✗	✗
1Transact							
/1transact	✓	✗	✗	✓	✗	✗	✗
SOAP	✓	✗	✗	✓	✗	✗	✗
1Validate							
/admin	✓	✗	✗	✗	✗	✗	✗
SOAP	✓	✗	✗	✗	✓	✗	✗

Interface	Roles						
	1SMS_admin	exchange_user	planner	transact_user	validate_user	worker	global_supervisor
1Workflow							
/1sms	✓	✗	✗	✗	✗	✓	✗
/admin	✓	✗	✗	✗	✗	✗	✗
SOAP	✓	✗	✗	✗	✗	✓	✗

 **Note:** 1Workflow uses SOAP to access the components.

 **Note:** In order to progress a job from ALLOCATED to COMPLETED, a user requires the following roles: exchange_user, planner, transact_user, validate_user and worker.

3 1Exchange Configuration

1Exchange can make use of FME Server and Snowflake products to publish and import data. FME Server is the default deployment and is available for users with or without the Snowflake products configured.

FME Server


If using FME Server to export and import data from the Oracle database, the mapping between the Oracle schema and required data formats is defined by a Workspace. This Workspace is deployed within FME Server and is referenced from a 1Exchange Exchange policy.

Snowflake

1Exchange can also make use of two **Snowflake** components to publish and import GML data:

- ▶ **GO Publisher** which is responsible for exporting from an Oracle feature schema to a GML file
- ▶ **GO Loader** is responsible for loading the changed GML data back into the Oracle schema

If using GO Loader and GO Publisher to import and export GML data from the Oracle database, the mapping between the Oracle schema and GML is defined by a **Product**. This **Product** definition is installed with 1Exchange during the installation process (see the *1Spatial Management Suite Installation Guide*) and referenced from a 1Exchange exchange policy.

 **Note:** In this instance, a **Product** refers to a configuration file that defines the way that data is published and imported using **GO Loader** and **GO Publisher**. For more information please refer to [Snowflake documentation](#).

URL Locations

The 1Exchange administration page is available at: [http://\[server\]:7115/1exchange/admin/](http://[server]:7115/1exchange/admin/)

The SOAP web service is located at: [http://\[server\]:7115/soapview/ExchangeService?WSDL](http://[server]:7115/soapview/ExchangeService?WSDL)

The REST web service is located at: [http://\[server\]:\[port\]/restview/rest/](http://[server]:[port]/restview/rest/)

Note: The ports used here are the default and you may have changed them during installation.

Security Settings

To increase security, you can control which media types (MIME types) and file extensions are allowed to be uploaded. By default, the values are set to a wildcard (*) which allows any type.

Specify a whitelist:

1. In the 1Exchange administration page, select the **Security Settings** icon:



2. From here you can enter one item per line into the designated box to create a whitelist:

- ▶ File Extensions

E.g:

zip

jpg

jpeg

- ▶ Media Types (MIME types)


E.g:


application/zip

image/jpeg

text/plain

Note: If you are trying to restrict any text file types, e.g. JSON, XML, TXT, be aware that the Media Type **must** be defined as `text/plain`, or these files will be rejected. Subsequently, you may decide to restrict the type of text files uploaded by defining extension types in the File Extensions Whitelist e.g. `.JSON`, `.CSV`, `.XML`

 **Warning:** Files can only be uploaded if both the extension AND Media Type is allowed from the respective whitelists. Specifying a * (wildcard) in either list means that any value is accepted.


 **Note:** If you want to place restrictions on file size, this must be configured in your application server:

- ▶ For WebLogic this is a server setting in the WebLogic console.
- ▶ In WildFly, this is maintained by editing the domain.xml.

Configuring GO Publisher and GO Loader Projects

Before configuring **GO Publisher** and **GO Loader**, perform the following data preparation tasks for each table that will be managed:


Data Preparation :

 **Note:** Ensure that you have not version enabled the database before carrying this out, as it changes the schema. For more information, please refer to "Version Enabling Database Data (GO Loader and GO Publisher only)" on page 16.

1. Add an **sms_job_id** integer column (typically a NUMBER(n) in Oracle).

This is used to store the job ID within which that feature was last edited, to enable job tracking and change-only validation.

This can be called anything, but the recommended default is **sms_job_id**. It will be incremented for each job created, therefore it needs to be long enough to store the highest number of jobs possible in the future, for example NUMBER(9).

 **Note:** Ensure that this job ID column name is recorded correctly in the 1Workflow admin page, in order to create specific job types in 1Plan and configure these job types to use one of the Exchange policies.

2. Ensure each table has a string primary key that will hold a GUID value of 38 characters, typically a VARCHAR2 in Oracle. For easier configuration, call this column **IDENTIFIER**.

3. Ensure that the table is suitable to be version enabled by Oracle Workspace Manager.

Typically this means ensuring that the table name or the column names are all no more than 25 characters long. For full restrictions, see the Oracle Workspace Manager documentation.



Note:

If using 1Edit, ensure the data in the database tables are rounded to the same grid tolerance that will be used in 1Edit, and if building topology in 1Edit ensure the data has been all topologically structured together at that tolerance.

If the data has different rounding in the database than in the client, when a feature gets edited and submitted back it may fail validation because the rounded co-ordinates no longer match the surrounding data, so tiny gaps and overlaps may be detected.

A recommended way to achieve this is to use 1Integrate to build topology using at least 2x rounding tolerance (or if building topology in 1Edit, use the same tolerance, typically 10x rounding tolerance) then write it back before rounding it in Oracle. The building of topology removes close vertices and ensures that rounding does not introduce errors.

Before configuring GO Publisher and GO Loader, you will need to get the GML XSD for your data:

Generate Feature Data XSD:

1. Generate the feature data XSD to represent the structure of the data to be served as GML.

You can either replicate the structure of the data in the database (the easiest option), or you may want to change the structure in the GML to a form more suitable for editing.

You may already have a GML structure prepared as an XSD or if not, there are two recommended options:

- ▶ To create a schema that is a duplicate of the data structure in Oracle you can use Safe Software's FME to read the tables required from the database and write to GML 3.2.1 format (this will only produce a feature class for Oracle tables that contain rows).
- ▶ Create any structure of schema using a tool such as Enterprise Architect from Sparx systems to create a UML model, combined

- ▶ with an open source UML to GML XSD converter such a ShapeChange from Interactive Instruments.
2. For easier configuration, ensure that the element name for each geometry type matches the name of the column in Oracle.

For example if the column name is GEOMETRY, ensure that it is shown as follows:

```
<element name="GEOMETRY"
  type="gml:GeometryPropertyType" minOccurs="0"/>
```

Configuring GO Publisher

GO Publisher Desktop is used to configure how the GML is *exported from* the database.

1. Create a new project and make sure you set the schema repository to be the folder containing the feature data XSD.

This ensures that relative paths are used in the project and therefore it can be deployed within 1Exchange as it will not be hardcoding the paths from the machine used for this configuration step.

2. Select the feature schema in the repository to add it to the project.



Note: Do not browse to the file. If it is not in the repository, change the repository location.

3. Keep the default XML root element as `gml:FeatureCollection`
4. Connect to a copy of the database feature schema.

This does not need to be in the same database instance or schema name as will be used in production; it must be the same table structure so that the desktop tool can do the correct mapping of the required table names. At runtime, the JNDI connection is used to read from the required database and schema.


5. In the **Project** tab, ensure that each table and column to be exported is ticked and mapped to the correct XML path.


When the XSD and the table structure match, the Automatic generation of XML button should match almost everything automatically.

6. In order for a conflict job to successfully go to a prepared state each table name in the GO Publisher project requires a unique key.

In the SQL Filters tab, for each table select **Edit** on the right-hand side.

Populate the **Unique Key** column, click **OK** and then click **Save**.

 **Note:** If you open the GO Publisher project in a text editor, you should see the unique key is now declared (e.g. `uniqueKey="IDENTIFIER"`).

 **Note:** If the unique key is not set, you might see the following error in the 1Exchange logs:

```
Could not obtain results from _CONF table:
ORA-00904: "null": invalid identifier
```


Configuring GO Loader

GO Loader Desktop is used to configure how the GML is *loaded into* the database:

1. Ask 1Spatial Support for the GO Loader plug-in used to set the job ID on data load.

This file is called **go-loader-workflow-plugins[version].jar** and should be copied into the GO Loader plug-ins folder at: \Program Files\Go Loader [version]\plugins.

2. Get the GML XSD for your data and also extract the **cou-gml-update.xsd** XSD from your installation zip file (located in: ms-exchange-release-[version]-release.zip\ms-exchange-release-[version]\install\dependencies\goloader\gl_product_template\schemas).
3. Create a new GO Loader project, and using the GML Schema tab, add the two XSDs mentioned above and then parse the schema.

 **Note:** Ensure you have both of the correct XSDs. Any modifications to these XSDs will require the next steps to be repeated. If you omit the **cou-gml-update** XSD, no features will be written to the database.

4. Ensure that the filter in the GML filter tab includes the featurtypes and columns that need to be extracted and do not include any properties that start `gml:` with the exception of `gml:identifier` which must always be included.

If the XSD does not use named geometry properties and includes gml types such as `gml:GeometryPropertyType` instead, include these types as well.

5. Within the **Translations** tab, visit each feature type and property and check that the table name and column name is correct.

Having a GML structure that matches the Oracle structure makes configuration easier because they will already be matched.

Ensure that you make the following changes:

- ▶ The only GML property that has to be set to **Insert** should be the top level `gml:identifier` which should have the column value set to the value of the primary key column in the database (this is why it helps to call that column IDENTIFIER). Discard the `gml:identifier codespace` sub property.

- ▶ Right-click on the table name and select the **jobname must be Integer** plug-in.

Insert the `plugin:AsInt` value and ensure it maps to the correct job ID column in the database (for example, `sms_job_id`).

Discard the `plugin:AsString` value.

If `sms_job_id` is in the list of GML Elements for this table, the translation method needs to be set to **Discard** so that the user can map the `ASInt` to `sms_job_id`.

- ▶ Ensure each required geometry property is set to **Insert** and maps to the correct geometry column in the database.
- ▶ Ensure that all geometry sub properties such as `srsDimension` or `gml:identifier` are set to **Discard**.

6. In the **Tables** tab for each table, right-click on the primary key column (typically IDENTIFIER) and select **Set the column as the table's Feature ID column**.

Version Enabling Database Data (GO Loader and GO Publisher only)

Version enabling should occur after configuring GO Loader and GO Publisher, as this will change the schema.

To perform GO Loader and GO Publisher configuration after version enabling, copy the database schema before version enabling and then configure against the unversioned copy in future. For each table to be managed, workspace enable each table.


In Oracle, use the function:

```
DBMS_MW.EnableVersioning('TABLE_NAME')
```

See the Oracle Workspace Manager documentation for more details.

Create Conflict Resolution Name Mappings (GoLoader and GoPublisher only)

When using GO Loader and GO Publisher, in order to use 1Exchange alongside 1Transact to perform conflict resolution, an XML file must be created to list the Oracle table name associated with each GML feature class. This is required by 1Transact to resolve conflicts in Oracle based on a GML input.

 **Note:** If this file is not uploaded as part of the configuration, conflicts cannot be resolved.

Create an XML file, **translationConfig.xml** with a structure as follows:

```
<?xml version="1.0" encoding="UTF-8"
standalone="yes"?>
<TranslationSummary
xmlns="http://www.snowflakesoftware.com/agent/trans
lationConfig">
  <tableTranslation>
    <table>TABLE_NAME</table>
    <element>
      <name> TABLE_NAME </name>
      <namespace>[namespace copied from the
feature data XSD]</namespace>
    </element>
    <primaryKeyColumn>IDENTIFIER</primaryKeyCo
lumn>
    <identifierElement>
      <name>identifier</name>
      <namespace>http://www.opengis.net/gml
/3.2</namespace>
    </identifierElement>
  </tableTranslation>
</TranslationSummary>
```

This file will be uploaded as part of the "1Exchange Product Configuration " on page 19.

Exchange Policies

An **Exchange Policy** specifies settings regarding the data that is extracted from and imported to the feature database schema. This could potentially be different for different job types, so multiple policies can be configured. And Exchange Policy is created in the 1Exchange Admin UI.

URL Locations

The 1Exchange administration page is available at: [http://\[server\]:7115/1exchange/admin/](http://[server]:7115/1exchange/admin/)

The SOAP web service is located at: [http://\[server\]:7115/soapview/ExchangeService?WSDL](http://[server]:7115/soapview/ExchangeService?WSDL)

The REST web service is located at: [http://\[server\]:\[port\]/restview/rest/](http://[server]:[port]/restview/rest/)



Note: The ports used here are the default and you may have changed them during installation.

FME Server

If using FME Server to convert data from the Oracle Database into any format and back again, each Exchange Policy will need to specify the FME Server instance and workspaces.

Using the 1Exchange admin UI you can input the parameters of your FME Server and Workspace An example set up is shown below:

Policy Type:

Snowflake

FME

FME Server Url:

FME Server Username:

FME Server Password:

FME Server Extraction Workspace:

FME Server Submission Workspace:

Automatically Grow Extents:

Buffer Distance in metres (mandatory):

Geometry Columns to Use (optional):

✓

Figure 3-1: Example parameters for an FME Exchange Policy

GO Loader and GO Publisher


Each **Exchange Policy** references a GO Publisher and a GO Loader configuration that defines how the data is converted from the database into GML and back again. These configurations are called **Products** (see "1Exchange Product Configuration " below).

Note: In this instance, a **Product** refers to a configuration file that defines the way that data is published and imported using **GO Loader** and **GO Publisher**. For more information please refer to [Snowflake documentation](#).


An example of the implementation of multiple policies using **GO Loader** and **GO Publisher** would be where the **Exchange Policy** defines an optional buffer surrounding the job extents to bound the features extracted. This may be significantly different for different job types. Similarly, there could be multiple products exporting different feature classes and the Exchange policy sets up the mapping between job type and the features products used.


1Exchange Product Configuration

The **Products** configuration process allows you to define an **Exchange Product**. Multiple products can be created, each referencing a **GO Publisher Project** (.gpp) and a **GO Loader Project** (.glp) and (optionally) each reading and writing to a different database schema.

 **Note:** In this instance, a **Product** refers to a configuration file that defines the way that data is published and imported using **GO Loader** and **GO Publisher**. For more information please refer to [Snowflake documentation](#).

In turn, these **Products** are used when configuring **Exchange Policies** in the 1Exchange administration page (see "Exchange Policies" on page 18).

 **Note:** 1Exchange requires GO Loader Desktop v 1.8 and GO Publisher Desktop v 4.0.

 **Note:** A set of GO Loader and GO Publisher *Products* can be configured at install time. Any additional products must be configured using this process. Policy configuration will still need to be performed for these products (including any from the installation process), via the admin page (see "Managing Exchange Policies" below).

In order to perform the product configuration, you will require the following:

- ▶ Feature Schema XSD
- ▶ GO Loader project (.glp)
- ▶ GO Publisher project (.gpp)
- ▶ (optional) Conflict resolution name mappings (.xml)

Perform Product Configuration


Perform 1Exchange Product Configuration

1Exchange Configuration Parameters

Managing Exchange Policies

At least one Exchange policy must be configured using the administration interface to use 1Exchange and the 1Spatial Management Suite, however multiple policies can be configured.

The 1Exchange administration interface is accessed from: [http://\[machine\]:\[port\]/1exchange/admin](http://[machine]:[port]/1exchange/admin)

 **Note:** The port number is defined during installation, but by default this is set to 7115.




There are two different Exchange Policy types available:

- ▶ Snowflake
- ▶ FME Server

1Workflow administrators can configure which Exchange policy to use for each job type, via the Workflow admin page.

Creating an Exchange Policy

Create an Exchange Policy:

1. Navigate to the 1Exchange Administration interface.
2. Click  and enter a name in the **Policy Name** field.
Click  to confirm the policy name.
3. Enter the policy details (see "Snowflake Exchange Policy Parameters" below or [FME Exchange Policy Parameters](#)).
4. Click  to save the policy.

Snowflake Exchange Policy Parameters:

Parameter	Description	Typical Value
Extraction Buffer	The distance by which to extend the extraction window when extracting features. This can be useful to ensure that an extract contains more data than in the originally provided geometry extent.	0
SRS	The SRS to be applied to the geometry when using a MBR or string of coordinates. When passing in the geometry through the user interface, this OGC SRS name is used to construct a valid GML geometry so that the extraction process can extract the correct data. This value is not used when calling the 1Exchange web service because the SRS name will already have been embedded in the GML geometry. For example, <code>urn:ogc:def:crs:EPSG::4326</code> .	
Extraction Project	A GO Publisher product registered during or after installation. A list of available products is available for selection.	

Parameter	Description	Typical Value
Submission Project	<p>A GO Loader product registered during or after installation.</p> <p>A list of available products is available for selection. When only one product is available, this is automatically selected.</p>	
Extraction Adaptor	<p>Adaptor to be applied to extracted features to convert from GML to a different required format.</p> <p>The parameters depend on each particular adaptor. For example, the standard FME server adaptor requires the URL, user name, and password for FME server and the name of the extraction workspace.</p>	
Submission Adaptor	<p>Adaptor to be applied to features before submission to transform the returned data back into change only GML 3.2 as expected by GO Loader.</p> <p>If the returned updated data includes all the data for the job, this process will also need to compare with the originally delivered file to detect the differences and generate a change-only GML file.</p>	
Grow Extents Buffer	Applied when growing extents (see "Grow Extents" on page 28).	
Grow Extents Geometries	A list of geometry columns that are queried to find intersecting features to use to grow the job extent before buffering it (see "Grow Extents" on page 28).	


FME Exchange policy Parameters:


Parameter	Description	Typical Value
FME Url	The location of the FME server	http://[hostname]: [optionalport]
FME Username	The username associated with the FME Server	
FME Password	The password associated with the FME Server	


Parameter	Description	Typical Value
FME Extraction Workbench	The repository and file location of the extraction workbench	[MyRepository]/ [MyWorkspace.fmw]
FME Submission Workbench	The repository and file location of the submission workbench	[MyRepository]/ [MyWorkspace.fmw]
Automatically Grow Extents		
Buffer Distance in metres (mandatory)	Applied when growing extents (see "Grow Extents" on page 28).	
Geometry Columns to Use (optional)	A list of geometry columns that are queried to find intersecting features to use to grow the job extent before buffering it (see "Grow Extents" on page 28).	

Updating or Deleting an Exchange Policy

To update an Exchange policy, select an existing policy and modify the parameters as required (see "Snowflake Exchange Policy Parameters" on page 21).

To confirm your changes, click .

 **Note:** If you navigate away from the current policy (for example, by selecting another Exchange policy), you must confirm whether or not you want to discard or save your changes.

To delete an Exchange policy, select an existing Exchange policy, click  and confirm the deletion when prompted.

Copying an Exchange Policy

To copy an Exchange policy, select an existing policy and click .

The name for the new Exchange policy is displayed. A default name consisting of a prefix `Copy_of_` and the name of the original Exchange policy is given.

With the exception of its name, the new policy will have the same attribute values as the original policy.

Applying Adaptors to an Exchange Policy (GO Loader and GO Publisher)

An exchange policy can optionally specify Extraction and/or Submission adaptors. These will be applied to the data after each extraction and before each submission.

To use an *Extraction Adaptor*, select **Use Extraction Adaptor**. This is carried out within the 1Exchange Admin Policy configuration window.

Once an Adaptor type has been selected, a list of available adaptors is displayed; by default this is the FME adaptor.

Select an adaptor from the list and the list of input parameters is displayed. In the case of the FME Adaptor, these are:

Parameter	Description	Typical Value
FME Url	The location of the FME server	https://<machine>:<port>
FME username	The username for the FME server pointed to by FME Url	N/a
FME password	The password for the FME server pointed to by FME Url	N/a
Workspace name	The name of the extraction/submission workspace.	Including containing repository (e.g. [repository]/[extraction_workspace]).


To use a *Submission Adaptor*, select **Use Submission Adaptor**, also located within the 1Exchange Admin Policy configuration window. A list of available adaptors is displayed.

Select an adaptor from the list and the list of input parameters is displayed.

If you are using the FME Adaptor, the list of parameters is the same as above, but the workspace will differ to the extraction workspace as it will be performing the reverse translation.

Additional adaptors can be added to 1Exchange as REST services. By default, 1Exchange includes a FME Server adaptor but other software can be used for converting data by implementing the REST service to invoke that software.

Setting up an FME Workspace for use as an Adaptor:

 **Note:** Ensure that any translated files end up in the root directory of the zip created by the **Data Download service**.

There are number of requirements on the FME Workspace that are needed in order for an exchange adaptor to work.

1Exchange will call FME Server to run the Workspace and will provide a number of parameters:

Extraction FME Workspace Published Parameters

Parameter	Type	Required	Description
Extraction_ File	Filename (Multiple)	Required	The GML 3.2.1 features file.
Job_ Metadata_ File	Filename (Multiple)	Required	An XML file containing the job metadata
Validation_ Report_ File	Filename (Multiple)	Optional	A validation report generated by 1Validate or 1Integrate. Only supplied for child jobs, or where pre-validation is carried out.

The job metadata file and validation report can be ignored, or can be processed in the workspace as required.

The extraction workspace should carry out any required translation on the supplied GML 3.2.1 file and needs to output to a file which is returned to 1Exchange by FME Server. 1Exchange will package up the returned file into adapted_features.zip.

Submission FME Workspace Published Parameters

Parameter	Type	Required	Description
Updates_ File	Filename (Multiple)	Required	The edited file which the user uploaded through the Worklist.
Extraction_ File	Filename (Multiple)	Required	The original GML 3.2.1 features file.

The submission workspace should carry out any required translations on the updates file to make it schema compatible with the original GML 3.2.1. It should then compare Updates_File to Extraction_File to determine the

changes and then generate a **GML** change only file which is returned to 1Exchange.

Both the submission and extraction workspace must be published to FME Server with services: **Data Download** and **Job Submitter**.

Add an Adaptor Implementation:

1. Add an entry in the adaptors.properties file in the form [adaptor-label]=[adaptor-url].

For example, for the FME adaptor the entry is:

```
FME\  
Adaptor=http://localhost:7115/adaptor/fme/rest/  
fme_adaptor
```

2. Define the REST service implementation.

The API is:

```
public interface ExchangeAdaptorRestService  
{  
    @GET  
    @Produces({ MediaType.APPLICATION_XML,  
              MediaType.APPLICATION_JSON })  
    Capabilities getCapabilities();  
  
    @GET  
    @Path("/{processId}")  
    @Produces({ MediaType.APPLICATION_XML,  
              MediaType.APPLICATION_JSON })  
    AdaptorProcess describeProcess(@PathParam  
    ("processId") String processId);  
  
    @POST  
    @Path("/{processId}")  
    @Produces({ MediaType.APPLICATION_XML,  
              MediaType.APPLICATION_JSON })  
    @Consumes(MediaType.MULTIPART_FORM_DATA)  
    Response execute(@PathParam("processId")  
    String processId, @Context HttpServletRequest  
    request) throws Exception;  
}
```

Where:

- ▶ `getCapabilities` - returns a `Capabilities` object that represents `List<AdaptorProcesses>`
- ▶ `describeProcess(processId)` - returns an `AdaptorProcess`, that contains the following information:
 - ▶ `id` - unique ID to identify the process in this adaptor
 - ▶ `description` - description of the process
 - ▶ `inputs` - list of input parameters
 - ▶ `output` - output parameter of the process
- ▶ `execute(processId, request)` - indicates the process to execute using the parameters in the request

As an example, for the FME adaptor the process is:

id	fme_adaptor
<code>description</code>	Converts data using an FME workspace
<code>inputs</code>	FME server URL, user name and password, workspace name
<code>output</code>	Currently unused, but could be the name of adapted file inside features.zip on extraction

All adaptors also have fixed inputs that do not need to be specified. These are:

- ▶ **Extraction adaptor** - file containing the extracted GML 3.2.1 data to be converted.

For the FME adaptor provided, this is a published parameter of the workspace with name `Extraction_file`.

- ▶ **Submission adaptor** - A file containing the original data after it was converted by the Extraction adaptor and a file containing the feature updates.

If the feature updates are provided in a change-only format then the adaptor only needs to convert the updates file to GML 3.2.1. If the feature updates are provided as a full file including the changes, the adaptor can compare against the original file to detect the changes and create a change-only GML file.

For the FME adaptor provided, this is a published parameter of the workspace with name `Updates_file`.

Policy Metadata

To see all metadata for an Exchange policy, select the policy. A panel is displayed on the right-hand side showing the following attributes of the

Exchange policy:


FME Exchange Policy metadata:

- ▶ **Name** - Name for the Exchange policy
- ▶ **Path** - URL for the Exchange policy
- ▶ **Created by** - User who created the Exchange policy
- ▶ **Created date** - Timestamp when the Exchange policy was created
- ▶ **Policy buffer** - Distance by which to buffer the extraction window
- ▶ **Policy srs_name** - Default SRS for the Exchange policy
- ▶ **Policy grow_extents_list**
- ▶ **Policy grow_extents_buffer**
- ▶ **Policy fme_url**
- ▶ **Policy fme_username**
- ▶ **Policy fme_password**
- ▶ **Policy fme_extraction_workbench**
- ▶ **Policy fme_submission_workbench**

Snowflake Exchange Policy Metadata:

- ▶ **Name** - Name for the Exchange policy
- ▶ **Path** - URL for the Exchange policy
- ▶ **Created by** - User who created the Exchange policy
- ▶ **Created date** - Timestamp when the Exchange policy was created
- ▶ **Policy go_publisher_project** - Name of the GoPublisher product
- ▶ **Policy go_loader_project** - Name of the GoLoader product
- ▶ **Policy buffer** - Distance by which to buffer the extraction window
- ▶ **Policy srs_name** - Default SRS for the Exchange policy

Some attributes are configurable, however others (such as the creation user name and timestamps) are assigned by the system.

To hide the metadata panel, click . To display the metadata, re-select the Exchange policy.


Grow Extents

1Workflow can automatically grow the extent of a job. This is useful to ensure that enough data is extracted to allow all the features that are wholly or partially within the original area of interest to be safely edited and validated.

Often, validation rules require all the features that surround a feature to be available to perform the validation. To guarantee that a feature can be edited

and validated, administrators usually require that jobs are automatically expanded to include all the features that surround those that are partially within the original area of interest. To achieve this, 1Workflow asks 1Exchange to query the base data for features to use to grow the job extent, and then adds a buffer to the result.

1Workflow contains a setting to indicate whether or not the original job extents should be grown automatically.

 **Note:** Activating a job takes slightly longer if job extents are being grown.

For each Exchange policy, an administrator can define how the job extent should be grown. The Exchange policy in the administration page contains the following fields:

- ▶ **Grow Extents Buffer** (mandatory) - Real with default value of 0.1, expressed in meters.


If a value is entered with a value lower than 0.1, it will automatically be set back to 0.1.

A buffer is always applied when growing extents. If no geometry columns are specified below, only the buffer will be applied to the original geometry.

- ▶ **Grow Extents Geometries** (optional) - Comma separated list of geometry columns that are queried to find intersecting features to use to grow the job extent before buffering it.

The format of this setting is `[table_name].[geometry_name]`, for example `LAND.GEOMETRY.WATER.POLY_GEOM`.

Any geometries (points or lines) from these geometry columns that intersect the boundary of the original job extent are used to grow the job extent to ensure enough data is available for editing and validation. Any holes in the resulting enlarged job extent are removed to ensure complete data.

 **Note:** 1Plan and 1Edit will continue to display the original unexpanded area of interest (as this defines where the edits should be happening), however the enlarged extent is used when extracting data and identifying the data to use for validation.

Additional extraction rules are applied by 1Exchange as normal after identifying the initial set of target features using the enlarged extent (rules such as including additional features that are referenced by those within the extraction extent).

1Exchange Tablespace Maintenance

Over time, tablespace maintenance will be required so that expansion of your database files does not result in you running out of disk space.

The main source of potential problems are the BPEL audit trail and the GO Publisher and 1Exchange Large Object (LOB) columns.

BPEL Audit Trail

BPEL is used to manage the workflow of jobs. Over time the instance metadata can grow quite large and will need to be purged.

Please see the official Oracle documentation for instructions regarding [Managing Database Growth](#).



Note: Our recommendation is to implement a scheduled purge script as outlined in the Oracle documentation.

GO Publisher and 1Exchange LOB Columns

GO Publisher and 1Exchange use database Binary Large Objects (BLOBs) to persist job artefacts for in-flight jobs.

The three columns most likely to encounter tablespace issues are:


- ▶ SMSGoPublisher.Published_File.Data
- ▶ SMSGoPublisher.Publish_Job.Zip
- ▶ SMSExchange.Default_Binval.Binval_Data

Rows are automatically deleted from these tables once a job has transitioned sufficiently through the lifecycle.

The BLOBs are defined using basic syntax, so Oracle defaults their storage to RETENTION, meaning that the database will retain modified LOB data in the LOB segment in accordance with the database's UNDO_RETENTION setting. Oracle will retain the modified LOBs for all jobs processed during this UNDO_RETENTION window.

Oracle will retain the LOB segments until one of the following happens:

- ▶ The table is dropped
- ▶ The table is truncated
- ▶ The table is moved to another tablespace (via `alter table ... move ...`)

 **Note:** Our recommendation is to identify all the tables with large LOB segments and move them to a different tablespace and then back again to their original tablespace. This will reduce their size back down to a minimum.

4

1Plan Administration

The 1Plan administration interface is available at: [http://\[server\]:7122/1plan/admin/](http://[server]:7122/1plan/admin/)

It can be used to perform the following tasks:

- ▶ Configuring metadata fields
- ▶ Defining job types
- ▶ Modifying enumeration values
- ▶ Enabling history records
- ▶ Specifying mode options
- ▶ Using Map Layers
- ▶ Archiving jobs in bulk



Note: After making any changes, refresh the 1Plan page for these changes to take effect.

The SOAP web service is located at: [http://\[server\]:7112/pcs/JobSoapService?WSDL](http://[server]:7112/pcs/JobSoapService?WSDL)

The REST web service is located at: [http://\[server\]:7112/pcs/rest/](http://[server]:7112/pcs/rest/)



Note: The ports used here are the default and you may have changed them during installation.

Configuring Metadata Fields

Jobs that are managed by 1Plan contain job metadata that is used for job planners to provide information and data for use by editors or other planners.

1Plan is deployed with a standard set of metadata fields, however these fields can be hidden or renamed and new bespoke ones added according to the organisation's business needs.

You can manage metadata fields from the 1Plan admin interface, within the **Job Metadata** page.

You can add new metadata fields, determine the client (the name of the item that is displayed in the panel when the user selects the field) and data type, specify user access, and delete user-defined fields.

Note: System-defined fields cannot be deleted or have their types changed, however the **Client** name can be changed to modify the visible text in the interface.

Click the arrow at the top of each column to change the order of the data to ascending or descending (alphabetically).

To change any metadata field, click on it and enter the new information.

To add a new field, click **+** and enter the relevant information.

To delete a field, click on the empty cell at the beginning of the corresponding row then click **-**.

To accept changes click **✓** and to abandon any changes click **✗**.

Mandatory Fields

A mandatory field must be filled in for a job before it can be saved.

You can specify if user-editable metadata fields should be mandatory or optional using the **Mandatory** column. This can be set to "true" (so the field is mandatory) by ticking a check box, or "false" (so the field is optional) by leaving the check box empty.

Job Metadata

ID	Client	Data Type	Realm	Enumeration	Planning User Access	Mandatory
id	ID	STRING	SYSTEM		READ_ONLY	true
priority	Priority	ENUMERATED_VALUE	SYSTEM	JOB_PRIORITY	READ_WRITE	<input checked="" type="checkbox"/>
extent	Extent	GEOMETRY	SYSTEM		READ_WRITE	false
parentid	Parent Id	REFERENCE	SYSTEM		READ_ONLY	false

Figure 4-1: Specifying if the metadata field "priority" should be mandatory

Note: Some fields must always be mandatory (for example if they are either required by the system), or must always be optional (for example if they are not user-editable). In these cases, you will not be able to edit the **Mandatory** column.

Automatically Generating Job Names

Job names are mandatory; administrators can specify that newly created jobs have automatically generated names by combining other job metadata values.

This can be enabled or disabled from the 1Plan admin interface, within **Job Metadata > Job Name Generation**.

To enable automatic job name creation, click **Job Name Generation** and select **On**.



Figure 4-2: Automatic Job Name Generation

Once enabled, the fields used to generate the job name can be specified. The fields are used in the order specified and underscores are used to separate the field values.

To change fields, click on a field name then select another field from the drop-down list.

To add a new field, click **+** and select a metadata field from the drop-down list.

To delete a field, click on the empty cell at the beginning of the corresponding row then click **-**.

To accept changes click **✓** and to abandon any changes click **X**.

The names are automatically populated in the user interface in the Job Data panel. To override a name, click on the field and enter a new name.

To revert back to the automatically populated name, delete all text entered before committing your changes.

Job Types

Job types are fundamental to the 1Spatial Management Suite; there must be at least one job type defined.

Different job types can be used to configure different data extraction rules or validation rules. You can manage the job types for a deployment by modifying the job type enumeration values (see "Enumerated Values" on the next page).

The 1Workflow administration page allows administrators to define the 1Exchange extraction rules and the 1Validate validation rules for each job type.

Enumerated Values

Enumerated values are a set of named values that can be picked from a list. Common examples of these are a set of job types or statuses.

You can manage Enumerated values from the 1Plan admin interface, within **Job Metadata > Enumerated Values**.

Enumerated Values


Name	Display Name
JOB_TYPE	Type
JOB_STATUS	Status
JOB_PRIORITY	Priority


+

Figure 4-3: Enumerated Values table

The **Name** value corresponds to a table in the 1Plan repository. The **Display Name** is how the drop-down list is displayed within the 1Plan interface.

To change a value, click on a cell within the table to overwrite it.

To delete an enumerated value, click on the empty cell at the beginning of the corresponding row, then click .

 **Note:** System-defined fields cannot be deleted or modified.

To add an enumerated value click  to add a new row, then overwrite the values in the table cells.

To accept changes click , or to abandon your changes click .

You can now further define your enumerated values by selecting the Enumerated Domain from the admin menu (e.g. **Job Metadata > Enumerated Values > Priority**).

Enumerated Domain "JOB_PRIORITY"



ID	Client	Realm	Primary Colour
HIGH	High	SYSTEM	
NORMAL	Normal	SYSTEM	

Figure 4-4: Settings for the "Priority" Enumerated Value

You can define your enumerated values by entering an **ID** of the field in the database, define the **Client** (the name of the item that is displayed in the 1Plan interface), and select the **Primary Colour** to be displayed for this value.

History Records

Administrators can specify whether or not the full history of each job's change in lifecycle is retained. This information is useful for auditing and process streamlining purposes.

Enable or disable the History setting using the radio buttons within the **History** page in the 1Plan admin interface.

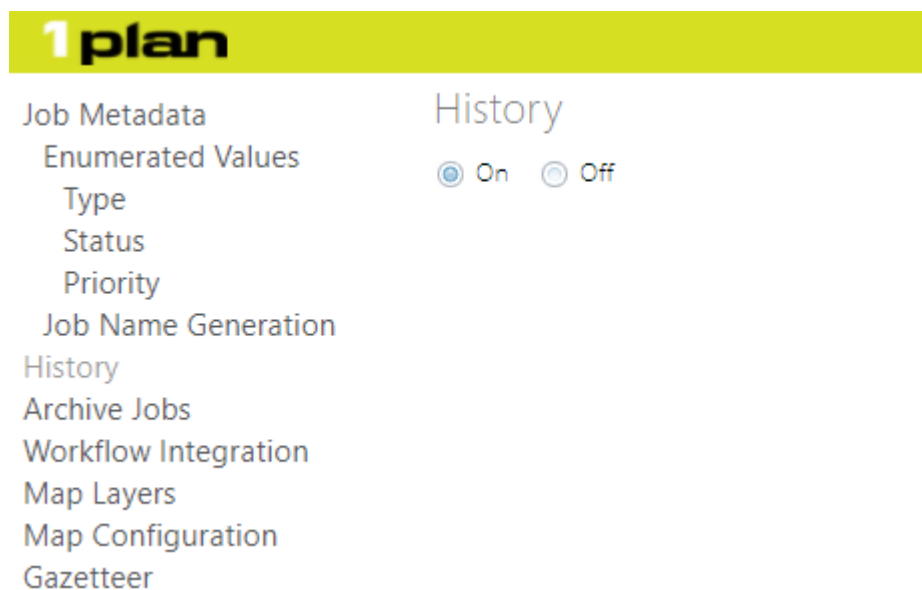


Figure 4-5: History setting

Turn this option **On** to write data to the JOB_HISTORY table whenever a job's status changes.

If the option is turned **Off**, no data is written to the JOB_HISTORY table but the ability to audit historic job lifecycle information will not be possible. The JOB_HISTORY table can be periodically purged if needed by deleting unwanted rows from it.

Archive Jobs

Jobs can be archived so that they are kept for record keeping but are otherwise not visible.

This can either be performed on individual completed jobs or in bulk in the admin interface.

Archive Jobs in Bulk:

1. Navigate to the admin interface at [http://\[server\]:\[port\]/1plan/admin/](http://[server]:[port]/1plan/admin/).
2. Select **Archive Jobs** from the menu.
3. Select a completed date and optionally a job type to filter completed jobs.

Query completed jobs, completed on or before: with the type

[Preview jobs available for archive](#)

Figure 4-6: Filter by completed date and (optionally) a job type

4. Click **Preview jobs available for archive**.

Any jobs meeting your criteria will be displayed in a table. You can view each job by clicking **View Job**.

Query completed jobs, completed on or before: with the type

[Preview jobs available for archive](#) [Archive jobs](#)

1 - 1 of 1 item 10 | 25 | 50 | 100

ID	Name	Type	Status	Creation date	Actual End Date	
145	145-1smstester	FIELD_EDIT	COMPLETED	10/03/2017	10/03/2017	View Job →

Figure 4-7: View the filtered jobs

5. To archive all jobs displayed in the table, click **Archive jobs**.
6. A confirmation message will display how many jobs in total will be archived.

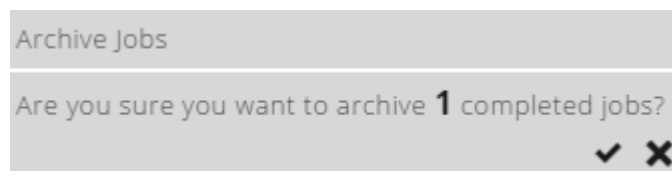


Figure 4-8: Confirm archive

Click to archive the jobs.


Workflow Integration

You can manage how 1Plan obtains job data from the 1Plan admin interface, within the **Workflow Integration** page.

Figure 4-9: *Workflow Integration settings*

First select **Workflow** or **Standalone** mode.

If you select **Workflow**, enter the URL where the 1Workflow SOAP service end point is located. If the URL is invalid, a message is displayed.

To accept changes click , or to abandon any changes click .


Map Layers

The **Map Layers** page (accessed from the admin interface) allows you to specify which web services to use as basemap layers for background mapping and as selectable layers for creating job extents.


These layers are available in the 1Plan user interface in the **Layers** panel.


To add a new map layer, click  and enter the relevant information:



Parameter	Description
Display name	The name of the layer. This name is displayed in the 1Plan user interface in the Layers panel.

Parameter	Description
Protocol	<p>Specify whether the Map Service uses the WMS, WFS or WMTS protocols.</p> <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;"> <p> Note: A Basemap layer can only use WMS or WMTS protocol. A selectable type layer can only use the WFS protocol.</p> </div>
Layer/Feature Type Name	<p>Specifies which WMS/WMTS layers to display or which WFS feature types to select from.</p> <p>This is a comma separated list with no spaces, for example: ROADS,RIVERS,BUILDINGS.</p> <p>The names of the layers or feature types must be defined by the service's GetCapabilities response.</p>
Type	<p>Select BASEMAP if you are using a WMS or WMTS service for backdrop map display as a basemap or overlay, or SELECTABLE if a WFS service will be used to create job extents by selecting geometries from the data.</p>
URL	<p>The URL of the map service, not including any parameters such as '?request=...':</p> <p>For example, if the following URL is used to retrieve the GetCapabilities document:</p> <pre>https://www.mycompany.com/geoserver/wms?request=GetCapabilities</pre> <p>Then the URL to provide is:</p> <pre>https://www.mycompany.com/geoserver/wms</pre>
Tile Set	<p>This column is used for WMTS maps. If you are using WMTS maps, this parameter must contain the name of the tile set to display for the WMTS server.</p>
Tile Rendering	<p>This column is for WMS only. Define if a WMS layer is served up as a single tile (Single) or multiple tiles (Multi).</p> <p>Serving WMS images as multiple smaller tiles is often faster, but some WMS servers will create identical map labels inside every tile if a feature spans multiple tiles. To avoid this, you can force 1Plan to request WMS images as a single tile.</p>

Parameter	Description
Tile Format	This column is for WMS only. Define if a WMS layer is served up as a PNG (faster for generated maps with flat colours) or as a JPEG (faster for aerial photos).
Geometry Column Name	This column only applies to WFS selectable layers. Many WFS servers do not need to specify the name of the geometry column to query and can therefore be left blank, however some WFS servers (in particular Esri ArcGIS server WFS) require the name of the geometry column from the WFS layers to be populated. For example, for Esri ArcGIS server, this value is typically SHAPE.

To delete a map layer, click on the empty cell at the beginning of the corresponding row, then click .

To accept changes click , or to abandon any changes click .

You can specify the order in which map layers appear in the Layers panel in the user interface. Select a map layer row in the table and click   to move the layer up or down the list.

Map Configuration

You can specify the maximum map extents for the data and jobs displayed in 1Plan from the 1Plan admin interface, within the **Map Configuration** page.

Map Configuration

Initial Map Extent

Min X	<input type="text" value="703,910"/>
Max X	<input type="text" value="705,542"/>
Min Y	<input type="text" value="807,594"/>
Max Y	<input type="text" value="808,769"/>

Maximum Map Extent

Min X	<input type="text" value="400,000"/>
Max X	<input type="text" value="800,000"/>
Min Y	<input type="text" value="500,000"/>
Max Y	<input type="text" value="1,000,000"/>

Misc Parameters

Spatial Reference System	<input type="text" value="EPSG:2157"/>
--------------------------	--

Figure 4-10: Map Configuration settings

The projection must specify the coordinate reference system in which job extents will be stored and in which the map services will be requested to display the maps. This should be an OGC GML compliant srsName value that matches your data, such as `EPSG:4326` or `urn:ogc:def:crs:EPSG::2157`.

You also specify the initial map extents that are used when a user first logs in to 1Plan. This could be the same as the maximum map extents (to start off viewing all the data) or could be a smaller extent to start the users displaying a more local area.

Note: 1Plan uses cookies to keep track of users' last map location so they will be returned to their previous extents when they log back in to 1Plan.

Gazetteer

A gazetteer allows users to search for named locations (such as street names or town names) and zoom to that location in 1Plan.

You can configure a gazetteer from the 1Plan admin interface, within the **Gazetteer** page.

Note: To configure a gazetteer, you must configure a WFS containing a single featuretype with at least a geometry property and a name property.

Gazetteer

URL	<input type="text" value="http://[machine]:[port]/geoserver/wfs"/>
Feature Type name	<input type="text" value="GAZETTEER"/>
Spatial Reference System	<input type="text" value="EPSG:2157"/>
Geometry column	<input type="text" value="GEOM"/>
Search text column	<input type="text" value="NAME"/>
	✓ ✗

Figure 4-11: *Gazetteer settings*

Specify the following values:

- ▶ **URL** - The URL of the map service, not including any parameters such as `?request=...`

For example, if the following URL is used to retrieve the GetCapabilities document:

`https://www.mycompany.com/geoserver/wms?request=GetCapabilities` then provide the URL

`https://www.mycompany.com/geoserver/wms.`

For password protected URLs, the service must be configured so that the security credentials can be passed as part of this URL, for example:

`https://Username:Password@www.mycompany.com/WMS/services/MapServer/WMServer`

- ▶ **Feature Type name** - The name of the feature type to query within this WFS.

To search for values within more than one table, either create a single gazetteer table from multiple source tables, or create a view that combines the rows from multiple tables and serve the view as a WFS featuretype.

- ▶ **Spatial Reference System** - the projection must specify the coordinate reference system in which the data is stored.

This should be an OGC GML compliant `srsName` value that matches your data, such as `EPSG:4326` or `urn:ogc:def:crs:EPSG::2157`.

- ▶ **Geometry column** - The name of the geometry property on which to centre the map.
- ▶ **Search text column** - A string property that is searched to find a list of potential names to show to the user.


User Session Timeouts (WebLogic Only)

The 1Plan service is used by both 1Plan and the 1Spatial Management Suite worklists.

WebLogic provides a default timeout for this service which means that after 3600 seconds (1 hour) of inactivity, users will be logged out of the 1Plan interface or the worklist interface.

Configuring User Session Timeouts:

1. Connect to the WebLogic Server Administration Console.
2. Select **Deployments > MSPlanDeployment**.
3. In the Modules and Components area, click **/1plan**.
4. In the **Change Centre**, select **Lock & Edit**.
You are prompted to save the Deployment Plan if one does not already exist.
5. In the **Configuration** tab, change the Session Timeout (in seconds) from 3600 to another value or to -1 to disable the timeout altogether.
6. Click **Save**.
7. You are prompted to save the Deployment Plan. You can use the default name and location or enter information to change it.

 **Note:** If you change the file location from the default, ensure that you save this file in a location where it will not be deleted; it is required when the servers restart.

8. Use the breadcrumbs to navigate back to the **MSPlanDeployment Overview** page.
9. In the Modules and Components area, click **/pcs**.
Repeat steps 4 to 7.
10. In the Change Centre, select **Activate Changes**.
A confirmation message is displayed.

Tuning the 1Plan Service (WebLogic Only)

1Plan on WebLogic is deployed with a Work Manager, designed to limit the amount of simultaneous work the back-end can perform, this will ensure that the server does not fall under too heavy a load.

The maximum workload constraints are configurable and can be tuned to suit your environment as required. These constraints are directly tied to the maximum allowed connection on the 1Plan datastore within WebLogic.

Increasing the maximum number of allowed database connections increases the amount of simultaneous work that can be done by 1Plan.

The Work Manager can also be monitored in the WebLogic console:


1. Open the Deployments list
2. Click or expand MSPlanDeployment
3. Click /pcs
4. Select the Monitoring tab
5. Select the Workload tab

In the Workload tab you will be able to see the total number of pending requests to 1Plan's API. If this number is regularly particularly high, it's indicative of tuning being required.

5 1Transact Administration

1Transact is accessed through a browser-based user interface, a web services API, and an SQL API.


- ▶ Management operations such as creating and deleting versions or checkpoints are available through both the web services API and the SQL API.
- ▶ Data operations are only available through the SQL API because they require data operations using an SQL session.

 **Note:** This help system discusses configuration and administration for a *PostgreSQL* database. For information about configuration and administration on an *Oracle* database, refer to the Oracle Workspace Manager documentation provided by Oracle.

The REST web service is located at: [http://\[server\]:\[port\]/ms-transact-review/rest/](http://[server]:[port]/ms-transact-review/rest/)

If you need to access the 1Transact Web Services API, then the WSDL document can be found at:

[http://\[machine\]:\[service_port\]/ms-transact-soapview/TransactService?WSDL](http://[machine]:[service_port]/ms-transact-soapview/TransactService?WSDL)

 **Note:** The ports used here are the default and you may have changed them during installation.

6 1Validate Administration

The 1Validate administration page is available at: [http://\[machine\]:8021/admin/](http://[machine]:8021/admin/)


The SOAP web service is located at:

- ▶ [http://\[server\]:8021/admin/soap/ValidateAdminService?WSDL](http://[server]:8021/admin/soap/ValidateAdminService?WSDL) (simple user)
- ▶ [http://\[server\]:8021/soap/ValidationService?WSDL](http://[server]:8021/soap/ValidationService?WSDL) (advanced administrator)

Validation Rules

Validation rules are written in the 1Validate administration interface.

Validation rules to be run when data is submitted need to be carefully planned and written to ensure that they have sufficient coverage of possible issues.

 **Note:** In order to be run within the 1Spatial Management Suite, rules must be placed in a subfolder of a root folder called PRODUCTION (for example `\\PRODUCTION\MyRules`).

Uploading Validation Rules

Validation rules must be installed within the same folder or subfolder. Set the name of this folder in the 1Workflow administration interface.


Rules can be uploaded into the system from an XML backup. This backup contains the rules and any metadata stores used by these rules.

If 1Validate rules need to be modified, they can be updated while the system is still running. By modifying rules in the live ruleset (or adding or removing rules from this folder), the next time a job is submitted it will use the new set of rules.

1Spatial recommends that rules are managed more formally by copying old sets of rules into folders with a relevant version name, so the rules that have been run on particular jobs is traceable and transparent.

Upload Validation Rules:

1. In the 1Validate administration interface, select the **Rules** icon.
2. Select **Tools > Upload**.
3. **Browse** to the backup file and select it.
The file name is displayed next to the **Browse** button.
4. Select **Submit**.
A table of rules is displayed.
5. Select **Close**.
6. Check that the new rules were added to the correct folder as specified in 1Workflow.
7. Move the new rules to the correct folder.


 **Note:** In order to be run within the 1Spatial Management Suite, rules must be placed in a subfolder of a root folder called PRODUCTION (for example `\\PRODUCTION\MyRules`).

8. If using metadata stores, locate this under the **Datastores** tab and test the connection.

Validation Rules

Validation rules are written in the 1Validate administration interface.

Validation rules to be run when data is submitted need to be carefully planned and written to ensure that they have sufficient coverage of possible issues.

 **Note:** In order to be run within the 1Spatial Management Suite, rules must be placed in a subfolder of a root folder called PRODUCTION (for example `\\PRODUCTION\MyRules`).

Uploading Validation Rules


Rules can be uploaded into the system from an XML backup. This backup contains the rules and any metadata stores used by these rules.

If 1Validate rules need to be modified, they can be updated while the system is still running. By modifying rules in the live ruleset (or adding or removing rules from this folder), the next time a job is submitted it will use the new set of rules.

1Spatial recommends that rules are managed more formally by copying old sets of rules into folders with a relevant version name, so the rules that have been run on particular jobs is traceable and transparent.

Upload Validation Rules:

1. In the 1Validate administration interface, navigate to the **Rules** page.
2. Select **Tools > Upload**.
3. **Browse** to the backup file and select it.
The file name is displayed next to the **Browse** button.
4. Click **Submit**.
A table of rules is displayed.
5. Click **Close**.
6. If necessary, move the new rules to the correct folder.


 **Note:** In order to be run within the 1Spatial Management Suite, rules must be placed in a subfolder of a root folder called PRODUCTION (for example \\PRODUCTION\MyRules).

Contextual Datastore

1Validate reads the feature data to validate directly from the source feature schema through an interface known as a datastore.

There are datastores for reading data from Oracle datasets, file based systems, and so on. There is also a specialised datastore (the *Contextual Datastore*) for efficiently including features that are referenced by or that reference the features within a spatial extent from a version-enabled Oracle schema.

Create an instance of the Contextual Datastore:

1. Click .
2. Create a folder called `Production`, if it does not exist already.
3. Select this folder and select **New > Datastore**.
4. In **Data Store Type** select **ContextualDataStore**.
5. In **Data Store Schema Name** enter the name of your datastore.
6. In **Datasource JNDI Location** enter the name of the datasource created in your application server for the feature schema.
7. In **Load Policy URL** enter the path and file name of the load policy, specific to your schema.



Note: This file will be provided by 1Spatial based on your schema.

8. Click **Test the Connection**.

7 1Workflow Administration

1Workflow administration provides:

- ▶ Customisable workflows and customisations that are applicable across all job types.
- ▶ A notification mechanism to email users at specific points in the lifetime of the workflow.
- ▶ Control who gets assigned conflict correction and validation correction jobs.

Settings

There are two types of settings for 1Workflow:

- ▶ Global settings that are applied to all jobs . For more information about Global settings please refer to " Global Settings" on page 57
- ▶ Job type specific settings. For more information please refer to "Settings Per Job Type" below

Access

The 1Workflow administration web interface is available at: [http://\[server\]:7122/1sms/admin/](http://[server]:7122/1sms/admin/)

The SOAP web service is located at: [http://\[server\]:7122/wf/WorkflowConfigSoapService?WSDL](http://[server]:7122/wf/WorkflowConfigSoapService?WSDL)


The REST web service is located at: [http://\[server\]:\[port\]/wf/rest/](http://[server]:[port]/wf/rest/)

Settings Per Job Type

Some settings can be applied to separate job types. For example, an organisation sets up Photogrammetry and Field edit jobs that follow the same process, but have slight customisations. In the **Settings Per Job Type** section within the 1Workflow Admin pages, the job types as defined in 1Plan are displayed.


To add new job types, first add them to 1Plan as described in the *1Plan Online Help*. Then, refresh the 1Workflow administration interface to display the new job type.

For each setting, click  at the bottom of the section to apply any changes made.

To cancel changes and revert to the last saved state, click .

The Settings Per Job Type settings are listed below:

Setting	Description
Mandatory	
Select whether job data is downloaded and uploaded as changes (Manual) or by directly updating the Oracle Workspace using a separate tool or process (Batch).	If the Batch job data is selected, the following "Batch edit module URL" will become available for update. For more information see Configuring Batch Workflow Mode .
Batch edit module URL	The URL of the Batch edit module, called as part of the job.
The 1Exchange policy to use for the job	The 1Exchange configuration to use for this job type (e.g. the data and data formats to exchange). For more information see "Exchange Policies" on page 18.
Embargo	
Enable Embargo to put jobs through a pre-download release approval step	The Embargoed status in 1Workflow is intended to offer a point of review for data which is to be worked on in a job before it is downloaded. This is designed to help ensure the accuracy and usability of data before any edits are made. For more information see Embargoed Status .
Grow Extents	
Automatically grow the extent of a job to ensure that any features partially included will be wholly within the area of interest (Requires Oracle Spatial License)	Automatically growing the extents of a job can ensure that all surrounding features are loaded to ensure the sufficient context for editing and validation. For more information see "Grow Extents" on page 28

Setting	Description
Read-only	
If enabled, jobs of this type cannot have changes uploaded (this will disable edit validation)	Prepare data extracts for sending out but disable the upload functionality as no edits are allowed.
Downstream Endpoint	
Invoke a custom endpoint after job completion	If you wish for a job to invoke a further action outside of the standard job lifecycle upon completion of the job, you can activate a downstream listener SOAP service to do so.
Custom downstream endpoint URL	The URL location of the downstream listener you wish to invoke.
Downstream Job	
Type of job to create as a downstream job of the in progress job on its completion	<p>If you wish for a new job of a different type to be created with the same metadata when a job in progress completes, then specify the job type to be created here.</p> <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;"> <p> Note: Ensure there are no circular references, otherwise the system would be infinitely creating jobs every time they are completed.</p> </div>
Downstream job assignment	The option for the worker and group to be inherited or named.
Worker Group	The Worker Group to be assigned if Named worker group and worker were selected for the Downstream job assignment.
Worker	The Worker to be assigned if Named worker group and worker were selected for the Downstream job assignment.
Automatically allocate the created downstream job	Select if you want the downstream job to be automatically allocated to a group and worker.
Automatically activate the created downstream job	Select if you want to automatically activate the downstream job.

Setting	Description
Validation of Edits	
Enable validation of job edits	After a job is edited it will pass through validation, to ensure accuracy of work carried out. These are defined by the validation rules within the validation rules folder.
Validation rules folder	The location of the validation rules. This needs to be inside a root folder called 'Production' and must contain a CRITICAL folder and optionally a WARNING" folder. E.g. for this structure, "Production\MyValidationRules\CRITICAL" specify 'MyValidationRules'. The folders are case sensitive.
Validation data store name	The name of the validation data store which specifies how to load the data. Must be configured to load all the required tables and attributes for validation, including the job id column. It needs to be inside a root folder called 'Production' E.g. for this structure, "Production\My1SMSDatastore" specify 'My1SMSDatastore'.
Refresh the job's workspace before attempting to run validation	By refreshing the job's workspace, any new changes that have been applied to the live data will be included during validation. This ensures that validation is done in the context of the latest version of the data. If there are version conflicts then the job will become conflicted and these need to be resolved before validation will continue again.
Pre-release Validation	
Enable pre-release validation	Pre-release validation is used to catch pre-existing validation issues, before the job is released to the worker.


Setting	Description
Validation rules folder	The location of the validation rules. This needs to be inside a root folder called 'Production' and must contain a CRITICAL folder and optionally a WARNING" folder. E.g. for this structure, "Production\MyValidationRules\CRITICAL" specify 'MyValidationRules'. The folders are case sensitive.
Validation data store name	The name of the validation data store which specifies how to load the data. Must be configured to load all the required tables and attributes for validation, including the job id column. It needs to be inside a root folder called 'Production' E.g. for this structure, "Production\My1SMSDatastore" specify 'My1SMSDatastore'.
Child Jobs (for further information see: "Child Job Settings" on the next page)	
Validation failure assignment	If the job fails validation, select who will carry out the fixes required.
Conflict resolution assignment	If the job is conflicted, select who will carry out the fixes required.
Quarantine review assignment	Select who will be in charge of the quarantine review.
Worklist Customisation	
Include a link to a custom client tool in Worklist for job editing	If a custom tool is used for viewing or editing of data, you can provide the URL of the tool here. This link will be visible from the worklist and will make an HTTP GET request to the URL.
Client tool URL	The URL of the tool to link to. The job id will be passed in with this request by replacing all instances of {jobid} with the job id. e.g. http://myserver:port/myTool/job/{jobid} or http://myserver:port/myTool?JobID={jobid}


Setting	Description
Client tool link text to display in Worklist	Select the text to display above the Job metadata.

Child Job Settings

Validation failure assignment:


Controls how the child job is created as a result of a validation failure


Setting	Description
Unassigned	The child job is created unassigned to any specific user or group.  Note: The job will be in the proposed state, rather than allocated.
Allocated to the worker from the invalid job	The child job is created with the same worker as the parent and is automatically allocated to that user.
Allocated to the worker group from the invalid job	the child job is created with the same worker group as the parent (the worker is cleared) and is automatically allocated to that group.
Allocated to a specific worker group	The child job is created with the specified group and is automatically allocated to the entered group.


 **Note:** Allocating to a specific group is used where there may be a separate QA team to handle conflicts or validation errors. This setting will work in tandem with the notification mechanism. For example, if you set it to allocate to a group and the allocated notification is set to email the worker group, the whole of the QA group would be emailed on the creation of a validation failure child.

Conflict resolution assignment :

Controls how a conflict resolution job is handled.


 **Note:** If your deployment is using an FME Workspace to convert data formats from Oracle, and you want to perform manual conflict resolution, please contact 1Spatial support.

Setting	Description
Unassigned	<p>The child job is created unassigned to any specific user or group.</p> <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;"> <p> Note: The job will be in the proposed state, rather than allocated.</p> </div>
Allocated to the worker from the conflicted job	The child job is created with the same worker as the parent and is automatically allocated to that user.
Allocated to the worker group from the conflicted job	the child job is created with the same worker group as the parent (the worker is cleared) and is automatically allocated to that group.
Allocated to a specific worker group	The child job is created with the specified group and is automatically allocated to the entered group.

 **Note:** Allocating to a specific group is used where there may be a separate QA team to handle conflicts or validation errors. This setting will work in tandem with the notification mechanism. For example, if you set it to allocate to a group and the allocated notification is set to email the worker group, the whole of the QA group would be emailed on the creation of a validation failure child.

Quarantine review assignment :

Controls who is assigned to the role of quarantine review.


Setting	Description
Allocated to the worker's group from the quarantined job	The child job is created with the same worker group as the parent (the worker is cleared) and is automatically allocated to that group.
Assigned to the planner from the quarantined job	<p>Assign the role of quarantine review to the person that planned the job.</p> <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;"> <p> Note: The job will be in the proposed state, rather than allocated.</p> </div>


Setting	Description
Allocated to the list of reviewers and/or reviewer groups from the quarantined job	A list of users or groups can be provided through the API as a comma separated list.
Allocated to a specific worker group	Assign the role of quarantine review to a specified named group.

Global Settings

Global settings that are applied to all jobs and are split into three sections:

- ▶ "General" below
- ▶ "Notifications" on page 59
- ▶ "Services" on page 59


For each setting, click  at the bottom of the section to apply any changes made.


To cancel changes and revert to the last saved state, click .

General

The General section contains the following settings which are applied globally:

Setting	Description
Conflict Resolution	
Include Diff Views With Conflict Extraction	Conflict child jobs will include information about the conflicting features based on the Oracle Workspace Manager CONF views. In addition to this, the Workspace Manager DIFF views can also be requested which includes information about the other users' non-conflicting changes which can help to put the conflicts into context

Setting	Description
Try to automatically resolve any conflicts	When selected, 1SMS will attempt to fix any conflicts detected, before creating a child for fix-up action. For more information, please see "Automatic Conflict Resolution" on page 63.
Maximum number of automatic conflict resolution retries	Sometimes, Oracle Workspace manager (particularly on Oracle 11g) doesn't return all conflicts when requested. To resolve this, auto conflict resolution can be tried a number of times to ensure that all conflicts have been resolved. This setting can be used to specify how many times to retry if conflicts aren't all resolved. A low number might result in unresolved conflicts. A high number might result in the job taking longer and using more resources before going to a conflicted state.
Automatically resolve conflicts using the latest changes for the following columns	<p>Enter the column names you want to apply this setting to, separated by line.</p> <div data-bbox="798 1294 1334 1417" style="border: 1px solid gray; padding: 5px;">  Note: This applies to non-spatial conflicts only. </div>
Job ID	
The column in the feature database tables that will be populated with the 1SMS Job ID on update	The name of the column populated by 1Exchange and used by 1Validate to indicate which features have been uploaded by the current job and therefore need validation. A database administrator must ensure that all tables that contain data to be extracted and submitted include this identically named column. The column must be an integer type and does not need to be indexed.

Setting	Description
Quarantine	
Jobs enter the Quarantine state prior to merging	This requires a job to be checked before completion, and stops the changes being immediately applied to the central database once validated.
Worklist Customisation	
URL of a custom logo to display in Worklist	Enter the URL of an image to use it as a custom logo in the top right corner of the Worklist interface.
Enable the light theme for all users	Activating will apply the light theme to all users instead of the default dark theme on next load of the Worklist.
Error Handling	
Clean up uploaded files on submission error	Toggle on or off to delete the uploaded files in the event of a submission error.
	<div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  Note: This is set to on by default </div>

Notifications

1Workflow has the ability to send email notifications on job state changes and in the event of errors. For details around these settings please refer to " " on the facing page

Services

Setting	Description
1Workflow host and port	The location of the 1Workflow host and port e.g: <code>http:// [hostname]: [port]</code>

Setting	Description
1Plan host and port	The location of the 1Plan host and port e.g: http:// [hostname]: [port]

Configuring Email Notifications

1Workflow has the ability to send email notifications on job state changes and in the event of errors. An example email can be seen below:


```
From: integration@domain.com
[mailto:integration@domain.com]
Sent: DATE TIME
To: Worker
Subject: [1Workflow] Info: Job 25 -> ACTIVATED
Job 25 (Digitise New Road) has transitioned from
ALLOCATED to ACTIVATED.
Job is assigned to: Job Worker

You can access this job in 1SMS Worklist by
clicking this link.

OPTIONAL: Table
OPTIONAL: Footer
```

In order to use this feature, you must first set up the User Messaging Email Driver and the Workflow Notification Properties within the Oracle Enterprise Manager page.


For more information about the configuration of email notifications, please see [Oracle documentation](#).

 **Note:** As well as configuring the workflow, the administrator will need to set up an E-Mail driver in the SOA Suite. This is documented further in [Oracle documentation](#).

Notifications Settings

The following settings are available in the Notifications tab of the 1Workflow Admin pages. For more information please refer to "1Workflow Administration" on page 50

Setting	Description
Send email notifications	Turn email notifications on.
Notification Recipient Configuration	Using the matrix, select which users receive notification upon a job reaching the relevant state or upon the completion of an action. For more information see "Notification Recipient Configuration " on the facing page.
Number of days before (-) or after (+) the planned completion date to send the late job alert as configured in the table above	Configure when a late email is sent: before, after or on the planned completion day. The planned completion day is represented by "0" and the email will always be sent at 00:00.
Notify an administrator on error	Controls whether or not a named administrator is emailed when a workflow contains an error. When it is switched on, the admin is prompted to select the username of the administrator from a dropdown box. The list is populated by users in your LDAP directory.
Notify the worker on error	This setting controls whether or not the Worker is sent an email to indicate that an error has occurred with one of their jobs. If this occurs, it is likely to happen after they have submitted a job and it may be useful to record that the issue has been handled by the administrator.

Setting	Description
Include a footer in all email notification	<p>You also configure emails to include a Notification Footer with fully customisable HTML.</p> <div data-bbox="866 409 1335 658" style="border: 1px solid black; padding: 5px;"> <p> Note: There is a maximum 4000 character limit for the HTML appended to notification emails.</p> </div>
Include a tabular representation of the current job metadata in all email notifications	This can be turned on to include job metadata as a table of values after an email, but above the footer.

Notification Recipient Configuration

The notification recipient configuration matrix allows for the selection of multiple users and job states or actions.

Notification recipient configuration


State Change	Worker	Planner	Worker Group	Embargo Reviewer	Quarantine Reviewer
Allocated	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Activated	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Prepared	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Downloaded	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Uploaded	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Committed	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Completed	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Validation Failure	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Conflict Resolution	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Revoked	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Relinquished	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Abandoned	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Failed	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Quarantined	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Quarantine Correction	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Approved	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Corrections Required	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Discarded	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Embargoed	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Late Completion Alert	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure 7-1: Notifications Configuration

Automatic Conflict Resolution

In addition to detecting conflicting edits, 1Exchange can also automatically correct potentially conflicting edits. For example, if two users try to change different attributes of the same feature at the same time or modify disparate sections of a long piece of linework.

Some organisations may want to always detect these changes for manual correction, or may allow the system to automatically correct these conflicts. The **Try to Automatically Resolve Any Conflicts** settings controls this behaviour.

 **Note:** If your deployment is using an FME Workspace to convert data formats from Oracle, and you want to perform manual conflict resolution, please contact 1Spatial support.

Reconfiguring 1Workflow

When 1Workflow is installed, it is linked to the other services in the system, including itself, by providing the service URLs to the installer.

If any URLs assigned at the point of installation need updating (either due to environmental changes or installation issues), the service URLs can be manually updated in the BPEL console.



Note: Due to an Oracle bug, Oracle patch 19635267 is a prerequisite (for 12.1.3).

Changing the service location:

1. Login to enterprise manager console e.g. `http://<server>:7001/console` (or `http://localhost:7001/console` to access from the local machine).
2. Navigate to the SMSMainWorkflow application: **SOA>soa-infra>default>SMSMainWorkflow**.
3. Under the **SOA Composite** drop down select **Service/Reference Properties**.
4. This will reveal a list of all services that 1Workflow communicates with (See table below for service mapping)

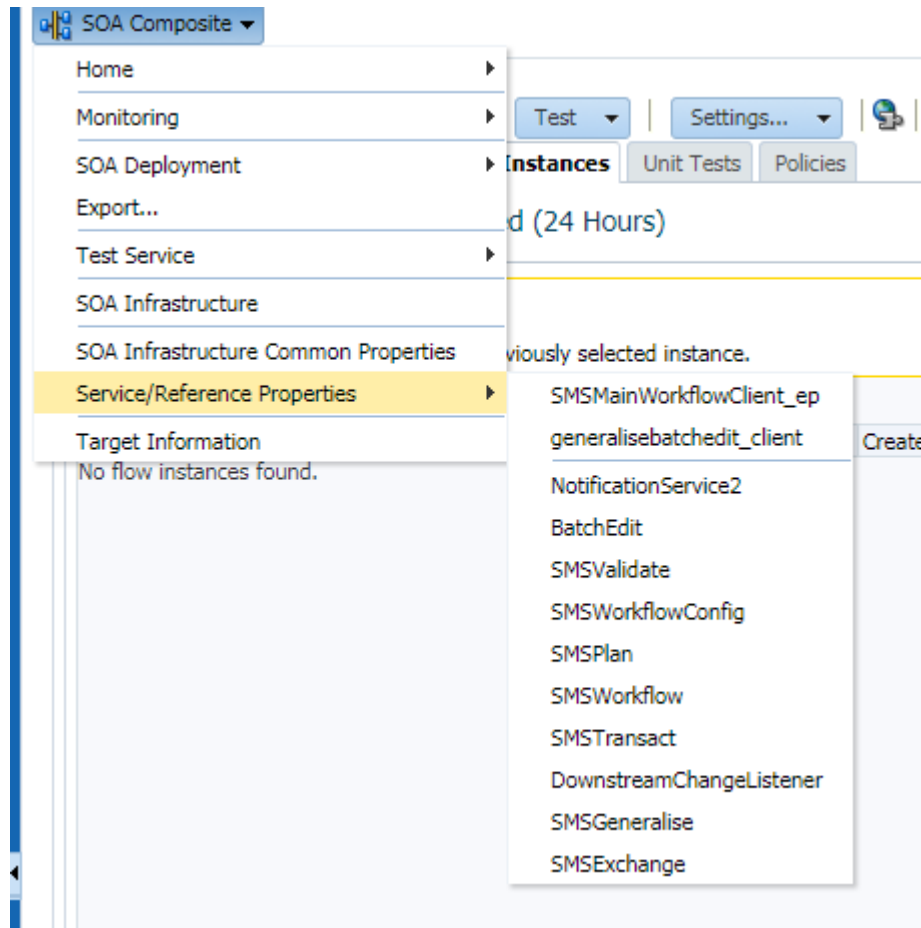


Figure 7-2: List of 1Workflow services

5. Select the service you wish to change.
6. Set the endpoint address to the new service URL.
7. Hit apply and the changes will immediately take effect.

The table details the Reference Names, Components and Address Formats for Service Mapping (any not in the table can be ignored)

Reference Name	Component	Address Format
SMSValidate	1Validate	http://<host>:<port>/soap/ValidationService
SMSWorkflow Config	Worklist	http://<host>:<port>/wf/WorkflowConfigSoapService
SMSPlan	1Plan	http://<host>:<port>/pcs/JobSoapService
SMSWorkflow	1Workflow	http://<host><port>/soa-infra/services/default/SMSMainWorkflow/SMSMainWorkflowCl

Reference Name	Component	Address Format
SMSTransact	1Transact	http://<host>:<port>/ms-transact-soapview/TransactService
SMSGeneralise	1Generalise	http://<host>:<port>/soapGene
SMSExchange	1Exchange	http://<host>:<port>/soapview/ExchangeService

8 1Generalise Administration

The 1Generalise administration interface can be used to install and upload flowlines, back up your data or delete your data.

The 1Generalise administration interface is available at: [http://\[server\]:8100/admin/](http://[server]:8100/admin/)

Within the administration interface, click the **Admin** tab and **Repository Administration** to access the administration functionality detailed here.

Installing System Flowlines

If you are running 1Generalise for the first time, you can install system flowlines to use with the 1Generalise user interface.

In the Install 1Generalise System Flowlines area, click **Install System Flowlines Now**.

To upload your own flowlines, in the Install Configuration area, click **Choose File** and browse to the file containing your flowlines.

Backing up Data

In the Download Configuration area, you can download and back up your 1Generalise configuration repository as a zip file.

Click the links to download individual file backups, or **Full Repository** to download all files.

Migrating Data

If you are upgrading from a previous version of 1Generalise, in the Database Migration area, click **Migrate From v[version number] Repository**.

Enter the host name, port number, Net Service name, user name, user password, and destination folder for the data. Click **Start Migration**.

Deleting Data

To delete all folders and keep repository data, in the Delete All Folders area click **Delete All Folders Now**. The 1Generalise Flowlines folder in the

Actions tab will be empty and all flowlines will be deleted.

To delete all data from the 1Generalise repository, in the Purge Repository area click **Purge Repository Now**.

Invalidating the repository cache

The **Invalidate JPA Cache** setting clears the internal cache. It is recommended that you contact 1Spatial Support for assistance before using this setting.

9 1Height Administration


1Height is a service that can be used either as a standalone application or integrated as part of the 1Spatial Management Suite. It can be used as part of 1Spatial Management Suite so that it is invoked automatically by 1Workflow, through the 1Height user interface, or from other software by calling its web services SOAP API (see "1Height SOAP Web Service" on page 77).

1Height allows height data that has been captured using techniques such as LIDAR scans to be used to improve the height accuracy of features. This is achieved by automatically setting the Z values stored in Oracle of any edited geometries using heights from a Triangulated Irregular Network (TIN) stored in Oracle.

Accessing 1Height

The user interface is accessed at: [http://\[server name\]:8200/#/height/](http://[server name]:8200/#/height/)

The user interface allows users to trigger a request to height geometries in the database and also allows them to request the height of the TIN at any specified x, y location.

 **Note:** This interface is expected to be used mostly as a testing interface to ensure that the configuration of 1Height is correct.

The administration interface is accessed at: [http://\[server name\]:8200/#/config](http://[server name]:8200/#/config),

or alternatively by clicking  in the user interface.

The administration interface allows administrators to specify the TIN blocks table name, manage the heighting policies, and manage the quality region metadata.

Configuring 1Height

The following steps outline the general steps required to configure 1Height ready for use.


1. Populate the Oracle TIN using Oracle or third-party tools.
2. For each geometry of a feature table that needs heighting, add a height quality column (of type NUMBER).

If there are multiple geometry columns in a table, you must enter one height quality column for each.




Note: If the table is version enabled, follow the Oracle Workspace Manage guide on how to add columns using the BeginDDL and CommitDDL functions.

3. The geometry quality columns on the data can be NULL, in which case features will always be heighted regardless of the quality of the TIN.
If the current height quality of the geometries is known, all the geometry quality columns can be updated with this value.
4. Install 1Height, specifying the locations of:
 - ▶ **JMS schema** - location to store security tokens. All 1Spatial Management Suite server-side components have a schema for this.
 - ▶ **Service Metadata schema** - This is where the QUALITY_REGION and HEIGHT_POLICY metadata tables will be created.
 - ▶ **Feature schema** - This must point to the schema that contains the data to be heighted. When a heighting policy names a feature table, the feature table must be in this schema.
 - ▶ **TIN schema** - the schema that contains the Oracle TIN Blocks table. This is where 1Height will read the TIN data from. When the administrator uses the interface to define the name of the TIN blocks table, the table must exist in this schema.
5. Populate the QUALITY_REGION table directly in Oracle with quality values and extent polygons for each height quality region in the TIN.
There must be either one quality region with an extent that covers the whole TIN or a number of quality regions whose extents tile together to cover the whole TIN.
6. Populate the HEIGHT_POLICY table either directly in Oracle or using the administration interface.
Typically you have a single height policy, although you can create more than one policy for different job types. Choose a height policy name and enter that name for every feature geometry profile in the table.

 **Note:** You need a feature geometry profile for every feature table's geometry that you want to be heighted. If a table or geometry is not listed in the table, it will not be heighted. You can consider scripting this step by inserting a row for every geometry column in the feature data schema.

7. Specify the TIN blocks table name in the administration interface.
8. Restart the 1Height service to pick up your changes.

 **Note:** You can modify the TIN blocks data, heighting policy table, or quality region table at any time. Restart the 1Height service to ensure that the latest values are used.

Triangulated Irregular Networks (TIN)

A TIN is a standard way of defining a surface of heights and is built by triangulating a set of points. Typically these points have been captured using LIDAR scans.

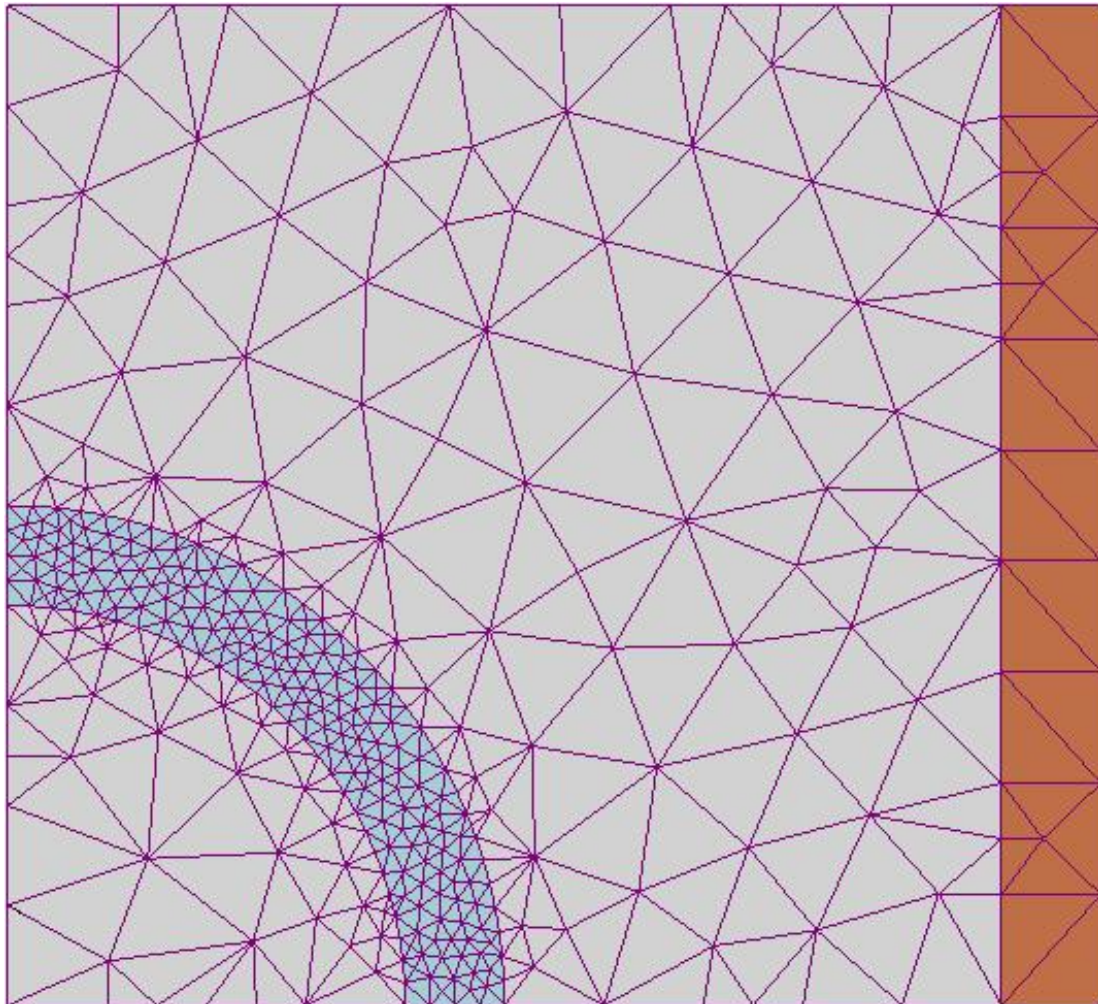


Figure 9-1: *Example TIN*

Oracle provides an `SDO_TIN` data structure for storing TINs as well as tools to build a TIN from a set of points. An Oracle TIN is made up of metadata about the TIN, plus a set of points, plus a set of triangles each defined by listing three of the points.


In Oracle, TINs are broken up into more manageable chunks called blocks. It is these blocks that contain the actual triangulation.

To "height" data, 1Height requires that a single Oracle TIN has already been created and administrators need to specify the Tin Block table that contains the data.

Adding a TIN

When you install 1Height you specify the database connection and schema that contains the TIN data, which can be anywhere on the network that is accessible to the application server.

Once installed, the administration interface allows administrators to specify the name of the TIN Blocks table within this schema.

 **Note:** You specify the TIN blocks table, not the TIN table itself, because the actual triangles are stored in the TIN blocks table. The TIN blocks table must contain only the blocks from the TIN to be used for heighting.

Height quality

When heighting data, 1Height needs to determine if the height quality of the TIN is better than the height quality of a feature's geometry. The height quality of the TIN is defined in the `QUALITY_REGION` table.

A TIN might contain points of varying height quality, for example in rural areas the heights may have been captured to within 1m height accuracy while in urban areas a more accurate height survey may have captured heights to within 0.2m accuracy.

While only a single TIN can be accessed by 1Height, this TIN can contain points with different height quality in different regions. The information about these height quality regions are defined in the `QUALITY_REGION` table, as follows:

- ▶ **ID** - (mandatory) The numeric Primary Key for this region.
- ▶ **QUALITY_VALUE** - (mandatory) A numeric value that defines the quality for this region. This is an accuracy number where smaller numbers indicate better quality. Features are only heighted if the feature's quality is equal or lower than this value.
- ▶ **DESCRIPTION** - A description to help administrators recognise the source of the quality patch (e.g. "2015 20cm LIDAR survey for North East").
- ▶ **EXTENT** - (mandatory) A 2D or 3D Oracle `SDO_GEOMETRY` polygon that defines the 2D spatial extent of this quality region.

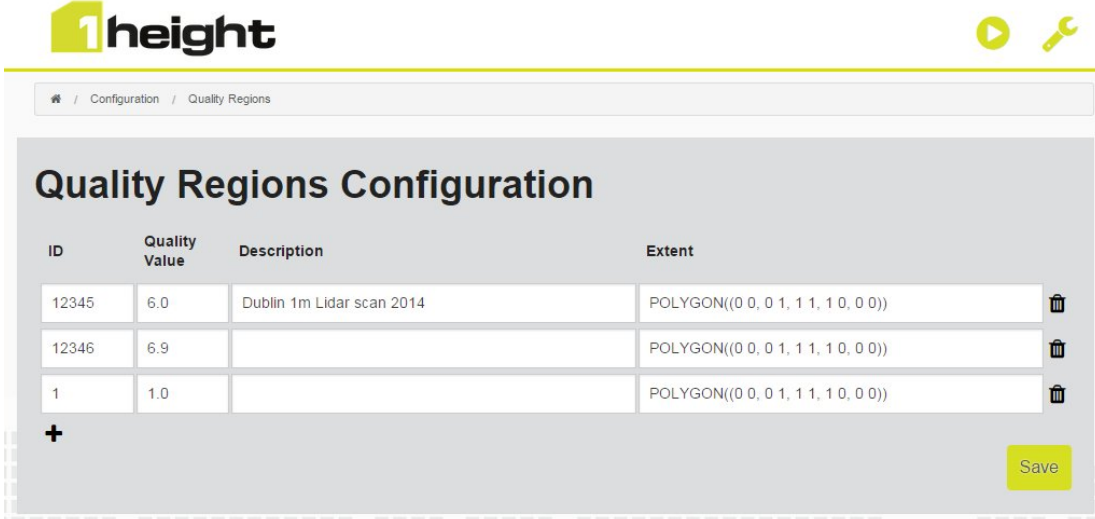
This table exists in the schema for 1Height service metadata specified during installation. Administrators must populate this table directly in Oracle and add either one row with an extent that covers the whole TIN or many rows, each with separate extent geometries that together cover the whole TIN.

This table can be populated in Oracle directly or by entering WKT polygons into the administration interface in the format: `SRID=27700;POLYGON((0 0, 1 0, 1 1, 0 1, 0 0))`

Note: The SRID of the extents must be the same as the SRID of the feature data to be heighted. Any features that are not within a quality region extent will not be heighted. If quality extents overlap, the "better" quality value is used in preference to the "worse" quality value.

Administrators can view the contents of this table from the 1Height administration interface.

Note: If the table is updated, the 1Height service should be restarted to guarantee that the changes are applied, as some values may be cached by the service.



The screenshot shows the 1Height administration interface. At the top, there is a navigation bar with the 1Height logo and a search icon. Below the navigation bar, the breadcrumb path is "Configuration / Quality Regions". The main content area is titled "Quality Regions Configuration" and contains a table with the following data:

ID	Quality Value	Description	Extent
12345	6.0	Dublin 1m Lidar scan 2014	POLYGON((0 0, 0 1, 1 1, 1 0, 0 0))
12346	6.9		POLYGON((0 0, 0 1, 1 1, 1 0, 0 0))
1	1.0		POLYGON((0 0, 0 1, 1 1, 1 0, 0 0))

Below the table, there is a plus sign (+) to add new entries and a yellow "Save" button.

Figure 9-2: `QUALITY_REGIONS` table in the 1Height administration interface

Heighting policies

1Height administrators define one or more heighting policies.

A heighting policy has a name and a list of feature geometry profiles. A feature geometry profile specifies the table name of the feature to height, the column name of a geometry from the feature and the name of the height quality column for that geometry.

Feature tables must exist in the feature schema defined during installation and must have the height quality column (a decimal number) added for each geometry in the table and the geometry column must be spatially indexed. If the geometry column is not spatially indexed, the heighting will not work.

A table with more than one geometry column that needs heighting will have a row added for each geometry and must have a different height quality column for each geometry. If the same quality column is used for the different geometries of a feature, some geometries might not get heighted because the quality has already been updated when heighting the first geometry.

The height quality column is used to decide if the feature needs to be heighted by comparing it with the quality region data for the TIN.

Policies are stored in a HEIGHT_POLICY table defined below and can be defined either through the administration interface or by updating the table directly:



Note: If the HEIGHT_POLICY table is updated directly, the 1Height service must be restarted to ensure that it uses the changed values.

- ▶ **ID** - (mandatory) The numeric Primary key for this policy.
- ▶ **POLICY_NAME** - (mandatory) Name of the heighting policy.
- ▶ **FEATURE_TABLE_NAME** - Name of the table that contains features to be heighted.
- ▶ **GEOM_COLUMN_NAME** - Name of the geometry column within the table to be heighted.
- ▶ **QUALITY_CONTROL_COLUMN_NAME** - Name of the quality column on the feature for the named geometry. This is used to decide if the feature needs heighting.
- ▶ **KEY_COLUMN_NAME** - The name of the identifier column for the table, used to update the correct feature's geometry. The data must have a single feature identifier column of any type. For version-enabled tables, the primary key will have been replaced with a compound key across the identifier and version columns, but it is only the unique identifier that should be specified here.
- ▶ **FILTER** - reserved for future use.

Heighting Geometry Features

When a feature's geometry is heighted, the whole geometry is checked to see if the quality regions that cover the geometry have equal or better quality than that registered for the feature's geometry.

If so, the vertices are heighted based on the heights from the TIN and the geometry's height quality is set to that of the poorest of the overlapping height quality regions.

This means that if half of the feature was heighted using quality 0.9 and half was heighted quality 0.5, the geometry's quality will be set to 0.9 which is the "worst common denominator" quality used.


Features can be heighted using any of three methods:

- ▶ Via 1Workflow (see "1Workflow Integration" on the facing page)
- ▶ Via the SOAP Web Service (see "1Height SOAP Web Service" on the facing page)
- ▶ Using the 1Height User Interface

Using the 1Height User Interface


A feature can be heighted by using the 1Height user interface to specify the following parameters:

- ▶ **Heighting Policy** - The name of the heighting policy to apply.
- ▶ **Workspace** - The name of the workspace in which the updates are made. Leave blank if data is not version enabled or to make the update into live.
- ▶ **Job ID** - if specified, only the features found by the heighting policy whose job ID is set to this value will be heighted.

 **Note:** If **Job ID** is specified, **Job ID Column** must also be specified.

- ▶ **Job ID Column** - defines the name of the existing job ID column on all feature tables that will be updated.
- ▶ **Region** - The spatial extent of the job. This allows the correct TIN blocks to be loaded for performance reasons.

Only geometries completely inside the job extent are heighted. This region is specified in the interface in a Well Known Text (WKT) polygon with an SRID, in the form: `SRID=1234;POLYGON((4 11, 4 13, 6 13, 6 11, 4 11))`

 **Note:** The SRID must be the same as the SRID of the feature data to be heighted.

Testing the TIN


As well as heighting a feature's geometry using the user interface or SOAP web service as described earlier, you can also request the height of any point within the TIN by specifying the x and y location from the user interface and asking 1Height to calculate the height of the TIN at that location.

1Workflow Integration

When integrated within the 1Spatial Management Suite, 1Height is invoked by 1Workflow during the job submission process before the job is validated.


In order for 1Height to be invoked by 1Workflow, heighting policies must be assigned to job types within the 1Workflow administration interface.

Administrators can define different heighting policies (or no heighting policy) for different job types.

 **Note:** When a feature is created or modified within a job, all the feature's geometries named in the heighting policy will be heighted, whether or not those specific geometries were edited.

A feature's geometry is heighted by 1Workflow when the following are true:

- ▶ The feature is created or updated in a job type that has a heighting policy defined via the administration interface.

 **Note:** 1Height knows if the feature has been created or updated if its job ID column has been set to the current job ID.

- ▶ The feature table and geometry column is listed in the heighting policy for that job type.
- ▶ The feature's geometry is completely inside the job extent (using INSIDE+COVEREDBY spatial relationship).
- ▶ The whole of the feature's geometry is covered by one or more quality regions.
- ▶ The feature geometry's height quality value is either NULL or is equal or greater (less accurate) than the covering height quality regions.
- ▶ The geometry's vertices are within the triangles defined by the TIN.

1Height SOAP Web Service

The SOAP WSDL location is: [http://\[server name\]:8200/soap/HeightService](http://[server name]:8200/soap/HeightService)

This displays an HTML page that lists the WSDL location, typically: [http://\[server name\]:8200/soap/HeightService?wsdl](http://[server name]:8200/soap/HeightService?wsdl)

The REST API is described by the WADL document which is usually accessed at: [http://\[server name\]:8200/api/application.wadl](http://[server name]:8200/api/application.wadl)

When 1Height is not integrated within 1Height, a feature is heighted either by calling the SOAP web service or using the 1Height user interface to specify the following parameters:

SOAP API Definition

The SOAP API has a single operation `heightFeatures` with the following parameters:

- ▶ **heightingPolicyName** - (mandatory) The name of an existing heighting policy.
- ▶ **Priority** - Unused value for future use. Omit the value or specify an arbitrary integer.
- ▶ **workspaceName** - Only relevant to version enabled tables. If specified, reads and writes the features within the specified workspace name. If not specified, or LIVE, then reads from the LIVE version.
- ▶ **jobId** - If specified, then this operation only heights features whose job ID column contains this job ID.



Note: If **jobId** is specified, **jobIdColumnName** must also be specified.

- ▶ **jobIdColumnName** - Tells 1Height the name of the feature table column that stores the job ID. This column name must be the same for all feature tables.
- ▶ **Geometry** - Specifies the extent of the job and is used for performance to reduce the potential set of TIN blocks that need to be analysed to height the features.

This geometry must be a 2D polygon and can be specified in one of three ways:

- ▶ *MBR (minimum bounding rectangle)* - By specifying the minimum and maximum x and y values to define a bounding box.
- ▶ *WKT (Well Known Text)* - An ISO/OGC defined format for specifying a geometry in a simple text-based format.
- ▶ *OGC GML MultiSurface* - An ISO/OGC defined format for specifying a polygon geometry in an XML format.