FME® Server Training Manual

FME Server 2015.1 Edition

SAFE SOFTWARE

Document and Copyright Information

Safe Software Inc. makes no warranty either expressed or implied, including, but not limited to, any implied warranties of merchantability, non-infringement, or fitness for a particular purpose regarding these Tutorials, and makes such Tutorials available solely on an "as-is" basis. In no event shall Safe Software Inc. be liable to anyone for direct, indirect, special, collateral, incidental, or consequential damages in connection with or arising out of the use, modification or distribution of these Tutorials.

This manual describes the functionality and use of the software at the time of publication. The software described herein, and the descriptions themselves, are subject to change without notice.

Data Sources

City of Vancouver

Unless otherwise stated, the data used here originates from open data made available by the City of Vancouver, British Columbia (data.vancouver.ca). It contains information licensed under the Open Government License - Vancouver.

Others

Forward Sortation Areas: Statistics Canada, 2011 Census Digital Boundary Files, 2013. Reproduced and distributed on an "as is" basis with the permission of Statistics Canada. © This data includes information copied with permission from Canada Post Corporation.

Digital Elevation Model: GeoBase®

Fire Hall Data: Some attribute data adapted from content © 2013 by Wikipedia (http://en.wikipedia.org/wiki/Vancouver_Fire_and_Rescue_Services), used under a Creative Commons Attribution-ShareAlike license

Stanley Park GPS Trail: Used with kind permission of VancouverTrails.com

Copyright

Revisions

Every effort has been made to ensure the accuracy of this document. Safe Software Inc. regrets any errors and omissions that may occur and would appreciate being informed of any errors found. Safe Software Inc. will correct any such errors and omissions in a subsequent version, as feasible. Please contact us at:

Safe Software Inc., Suite 2017, 7445 – 132nd Street, Surrey, BC, Canada, V3W1J8
e-mail: services@safe.com

Safe Software Inc. assumes no responsibility for any errors in this document or their consequences, and reserves the right to make improvements and changes to this document without notice.

Trademarks

FME® and SpatialDirect® are registered trademarks of Safe Software Inc. All brand or product names are trademarks or registered trademarks of their respective holders and should be noted as such.

Document Information

Document Name:	FME Server Training Manual 2015
Version:	FME Server 2015.1
Updated:	February 2015

Welcome to Safe Software

Introduction

Introduction

FME Version
Sample Data

FME Server Basic Training 2015

This document is the training manual for FME Server 2015

FME Version

This training material is designed specifically for use with FME2015. You may not have some of the functionality described if you use an older version of FME.

Sample Data

This sample data required for the exercises in this document can be obtained from:

www.safe.com/fmedata

Course Details

Course Details

Course Structure
Training Philosophy
Prerequisites

FME Server Basic Training 2015

This is the workspace authoring training course for Safe Software's FME Server application.

Course Structure

This training material covers solely how to create FME translation models for use on FME Server.

It is comprised of the following sections:

1. Introduction to FME Server
2. Running Workspaces on FME Server
3. Self-Serve with FME Server
4. Real-Time with FME Server
5. Automation with FME Server

The content may be spread over several days, or your instructor may choose particular sections to be covered.

Training Philosophy

This training provides a framework for authoring workspaces for FME Server. We hope that you will learn all the tools upon which to base your work, and go home with many new FME ideas!

The training will introduce basic concepts and terminology, help you become an efficient FME Server user, and direct you to resources to help apply the product to your own needs.

Prerequisites

This training material is intended for persons with some prior experience of using FME Workbench. The content assumes a basic familiarity with the concepts and practices covered of FME Desktop; at least to the extent covered by the FME Desktop Tutorial.

In particular it would be helpful to be familiar with:

- Parameters and published parameters
- Managing Readers, Writers and feature types in a workspace
- Using transformers similar in complexity to the Clipper and Joiner

About the Manual

About the Manual

Icons
Troubleshooting

CONNECT.
TRANSFORM.
AUTOMATE.

2015.0

FME Server Basic Training 2015

The FME Server training manual is yours to keep. It includes detailed material to help you remember your training.

During training we suggest you pay close attention to your instructor; they may cover the course content in a different order than the manuals, and will skip or add new content to better customize the course to your needs.

You'll find most of the instructor's PowerPoint slides reproduced in your manuals, as above. This will help you relate the manual content to the topic the instructor is discussing.

Icons

In the training manual you may see the following icons…

Tip: Additional advice to help apply the knowledge you have learned.

Caution: A warning where misuse of FME could lead to difficulties.

Exercise: An exercise that the instructor will use to demonstrate a particular point or feature of FME. You may work through the exercise with the instructor or on your own.

Advanced Exercise: Extra challenges for students who are quick to finish and want to take an exercise a little bit further.

New: A feature new to, or significantly changed in, the most recent version of FME.

Q&A: Questions about the course content and how it is applied.

Important hints and tips appear in a box like this, to separate them from ordinary content.

Troubleshooting

Each chapter has a short appendix with troubleshooting suggestions; general hints to follow in case FME Server causes problems for you.

Course Resources

Course Resources

On Your Training Computer
Amenities

FME Server Basic Training 2015

A number of sample datasets and workspaces will be used in this course.

On Your Training Computer

The following applications may already be installed, licensed, and located on your training computer (real or virtual):

- Java Virtual Machine
- Apache Tomcat
- FME Desktop Version 2015
- FME Server Version 2015
- FLEXlm License Manager
- PostgreSQL

A collection of sample datasets and workspaces are available in **C:\FMEData2015**

The data used in this training course is based on open data from the City of Vancouver, Canada.

Most exercises ask you to assume the role of an analyst at an air photo reseller and to solve a particular problem using this data.

Throughout the manual people from the city of Interopolis pop up to give you advice and to dispense FME-related wisdom.

Amenities

For in-person training, your instructor will explain the facilities and amenities available to you during the course.

Please respect other students' needs by keeping noise to a minimum when using a mobile phone or checking e-mail.

Introductions

Introductions

Who am I?
Who are you?
What do you plan to use FME Server for?

FME Server Basic Training 2015

Welcome to this FME Server training course.

Who Am I?

This is your instructor's chance to introduce her or himself.

Who Are You?

This is your chance to introduce yourself, and tell us about the following:

- Your organization
- Your planned use of FME Server

Introduction to FME Server

What is FME Server?

What is FME Server?

FME Server Summarized
FME Server vs. FME Desktop

FME Server is a powerful product for handling large volumes of data at the enterprise level.

FME Server Summarized

FME Server is a powerful product for handling large volumes of data at the enterprise level. It has three core capabilities: Self-Serve, Real-Time, and Automation.

Self-Serve

Self-Serve is the ability for the end-user to select and download the data they require, in the format and structure they require; or to upload data for validation. It eliminates the need for a data manager/analyst to carry out manual data distribution tasks.

Real-Time

Real-Time is the ability to react to real-time events and sensors, to carry out immediate updates, and to deliver instant notifications. It allows subscribers to have the most up-to-date information for their business decision making.

Automation

Automation is the ability to carry out data processing at a specific schedule, and to spontaneously move data through different systems and web services – even onto mobile platforms and devices. It allows data to move from anywhere, to anywhere, without manual intervention.

FME Server vs FME Desktop

FME Server expands on the local processing tools of FME Desktop by providing the same core data translations and transformations, but at an enterprise level, and with the above enterprise-level capabilities.

With these capabilities, FME Server can take advantage of a huge variety of different communication technologies. This means it can provide many different ways to:

- Start a workspace
- Structure a translation
- Deliver the output

FME Server does not include an authoring tool, so FME Desktop is used to create these workspaces.

The focus of this course is how to create these workspaces in ways that will take full advantage of FME Server's enterprise capabilities.

FME Server Roles

FME Server Roles

Author
User
Developer
Administrator

FME Server Basic Training 2015

Different roles have different interests in FME Server

When using a product with the range and scope of FME Server, it's no surprise that there are a number of different roles available to different users.

Author

An author is someone who creates translations (workspaces) using FME Desktop and publishes them to FME Server for use by end-users. This particular role is the focus of these training materials.

Typically an author would work at the analyst level, and would be an experienced FME Desktop user with a good understanding of Readers, Writers, transformers, and other FME Workbench functionality.

User

A user is defined as a person who accesses data using an FME Server Service.

It is not expected that a user in this sense has, or needs to have, any experience of FME and does not even need to be aware of FME Desktop or FME Server.

The user might be someone like a CAD operator checking out spatial data, a business manager looking at corporate data in an FME Server client, or members of the public who want to download data for personal use.

Developer

FME Server can be used as the back-end to power other software applications.

A developer (in FME Server terms) is someone who creates applications that submit jobs to FME Server and then handle the results. This role is somewhat covered by these training materials.

Web developers may choose to include FME Server Services within their own web applications or create their own services using the FME Server API.

Administrator

An FME Server administrator is the person responsible for installing and maintaining FME Server and its related services.

The administrator's tasks include:

- Planning system architecture
- Installing prerequisite applications
- Installing and licensing FME Server
- Setting up FME Server services
- Monitoring FME Server services
- Troubleshooting
- Scaling FME Server

FME Server Architecture

FME Server Architecture

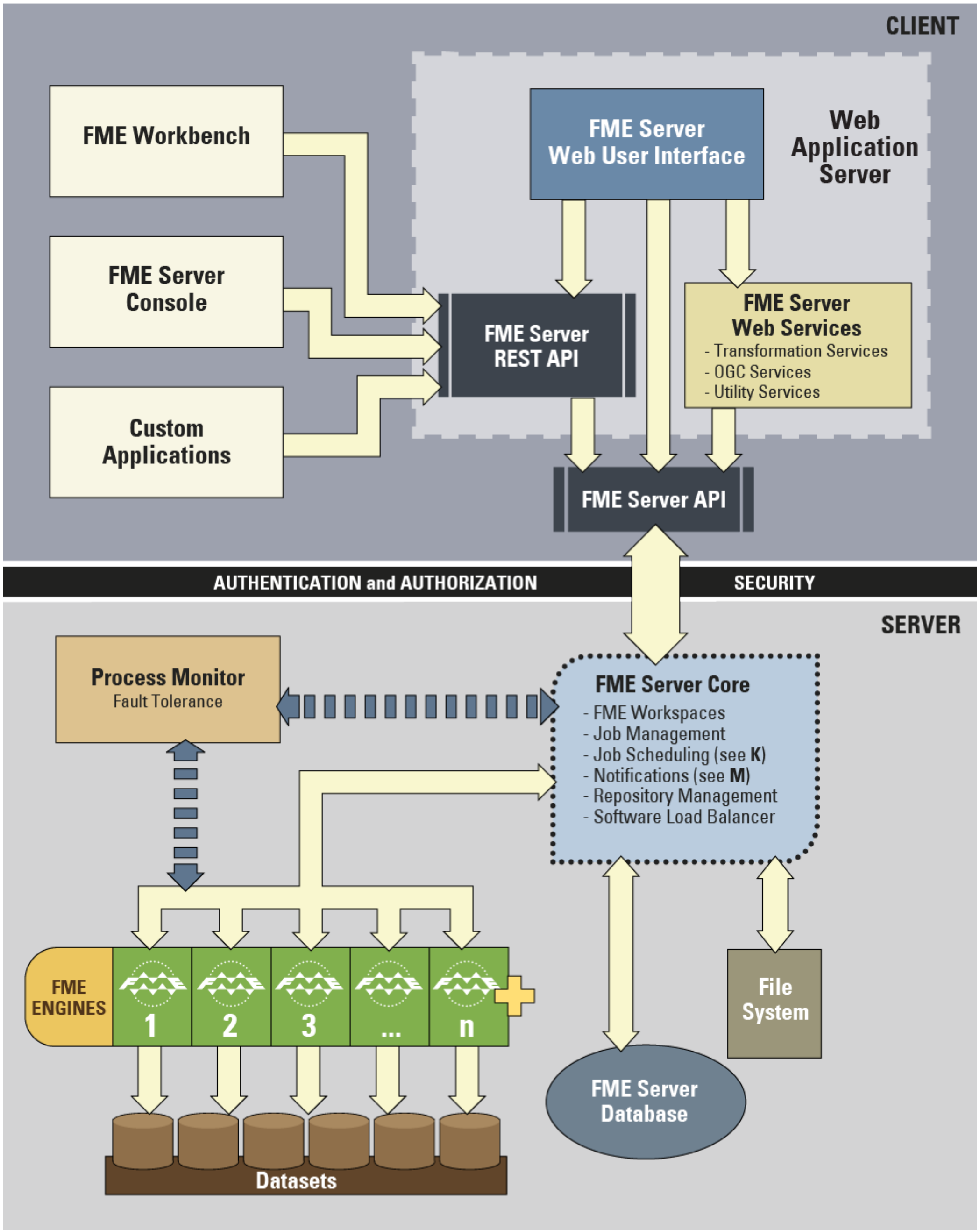
FME Server Components
- Engine
- Server Core
- Web Services

FME Server Basic Training 2015

FME Server consists of a number of different components.

The architecture of FME Server looks something like this:

FME SERVER ARCHITECTURE



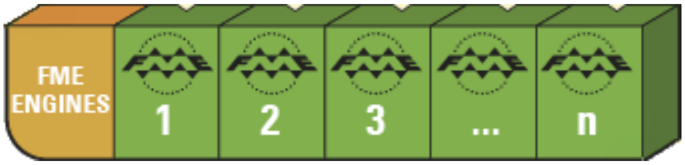
FME Server Components

Don’t worry about trying to understand the whole structure, just be aware of the main components of FME Server:

- **FME Engines**: To carry out data transformation processing
- **Server Core**: To queue jobs, handle scheduling, and manage load balancing
- **Web Services**: To handle networking capabilities

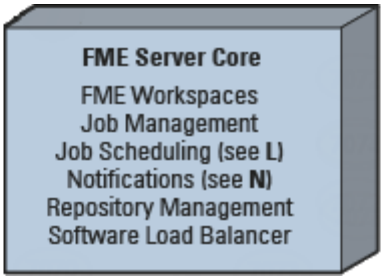
FME Engines

FME Engines process job requests by running FME Workspaces. This is the same core engine, carrying out the same processing, that is used by FME Desktop.



Each FME Engine processes a single request at a time.

FME Server processing can be scaled by adding FME Engines to the same computer or to separate computers within a distributed FME Server environment.



Server Core

The FME Server Core manages and distributes job requests (queuing, request routing, scheduling), the repository contents (workspaces, custom formats, custom transformers, data), and notification requests.

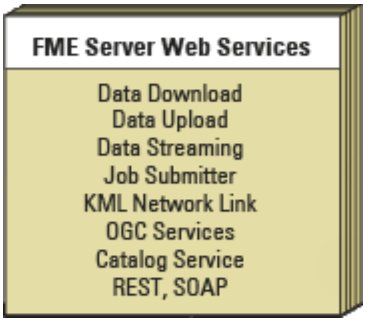
The FME Server Core contains a Software Load Balancer (SLB) that distributes jobs to FME Engines.

Web Services

Much of the FME Server networking capabilities are handled using what we call “Services”. Services are software whose interface provides communication between server and clients.

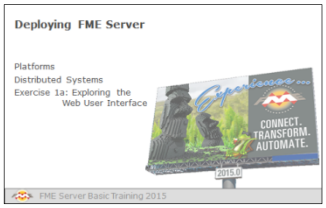
FME Server has a number of services:

- Data Download
- Data Upload
- Data Streaming
- Job Submitter
- KML Network Link
- OGC Services (WFS and WMS)
- Catalog
- Token Security
- SOAP
- REST
- Notification



Some services (for example, Data Download) are “transformation” services that carry out data transformation, whereas others (for example, Catalog) are non-transforming “helper” services.

Deploying FME Server



Deploying FME Server

Platforms
Distributed Systems
Exercise 1a: Exploring the Web User Interface

FME Server Basic Training 2015

FME Server can be deployed in a number of configurations

Platforms

FME Server can be deployed on a variety of platforms:

- ***Local Infrastructure (Physical Hardware)***
  This is the traditional platform where you purchase FME Server and install it on your own hardware systems.

- ***Infrastructure as a Service (Virtual Hardware)***
  This is where you purchase FME Server and install it on virtual hardware that is provided as service by a company such as Amazon.

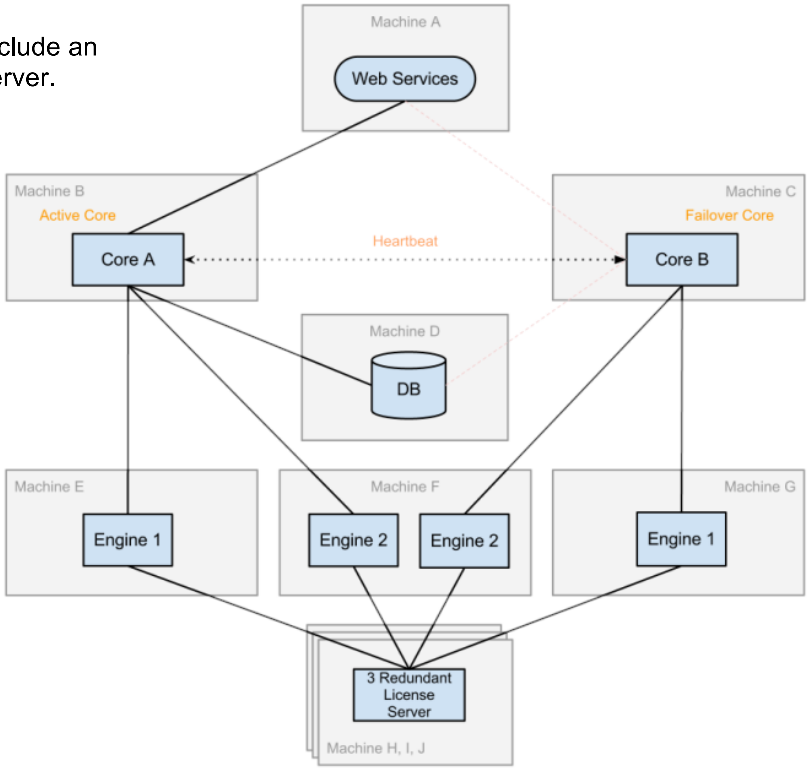
- ***Platform as a Service (FME Cloud)***
  This is where FME Server is delivered pre-installed on an Amazon virtual computer, with the whole platform provided by Safe Software on a pay-as-you-go basis.

Distributed Systems

A distributed system is when different components of a system are located on separate, networked, computers.

In a distributed environment, FME Engines run on a computer or computers that are separate from the FME Server host. Administrators can configure the FME Engines to register with a failover FME Server host, which acts as a backup if the primary FME Server host fails.

Other failover mechanisms include an additional core and license server.



Exercise 1a: Exploring the Web User Interface	
Scenario	FME Server author, administrator, or end-user
Data	N/A
Overall Goal	Explore the FME Server web user interface
Demonstrates	Using the FME Server web user interface
Starting Workspace	None
Finished Workspace	None

Follow these steps to log into the FME Server web user interface and examine the available functionality.

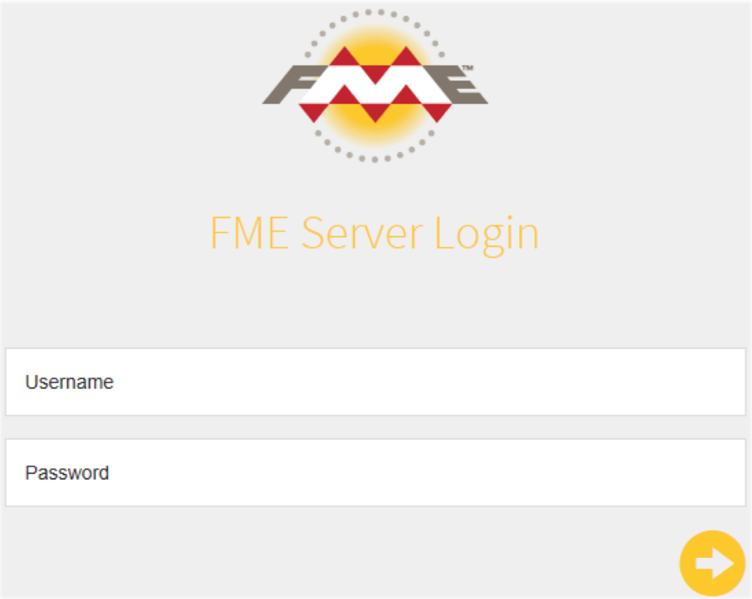
1) Connect to Server
In your web browser, enter the address to your FME Server.

> *When FME Server is installed on either physical or virtual hardware, the address is http://<servername>/fmeserver*
>
> *If you are using FME Cloud, then the address is: http://<server name>.fmecloud.com/fmeserver*

This will open the web user interface login screen for the FME Server being used. Bookmark this web address, since you will use this link quite often.

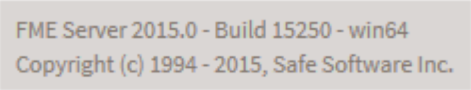
2) Log In to Server
In the User Login dialog, enter a username and password for your FME Server account. A common username/password combination for a training installation is admin/admin

Click the Arrow button.



3) Examine User Interface.

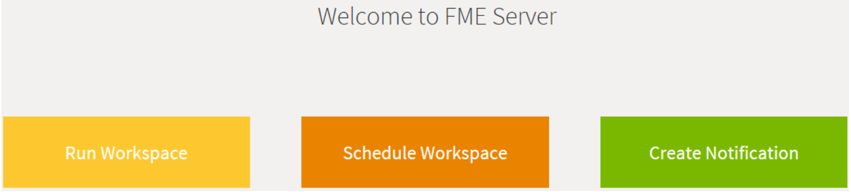
Examine the user interface. This is your primary method for interacting with FME Server. Notice that the bottom-left corner shows which version and build of FME Server is being used by this instance:



FME Server 2015.0 - Build 15250 - win64
Copyright (c) 1994 - 2015, Safe Software Inc.

4) Examine Repositories

Let's look first at the system's repositories. Click the Run Workspace button.

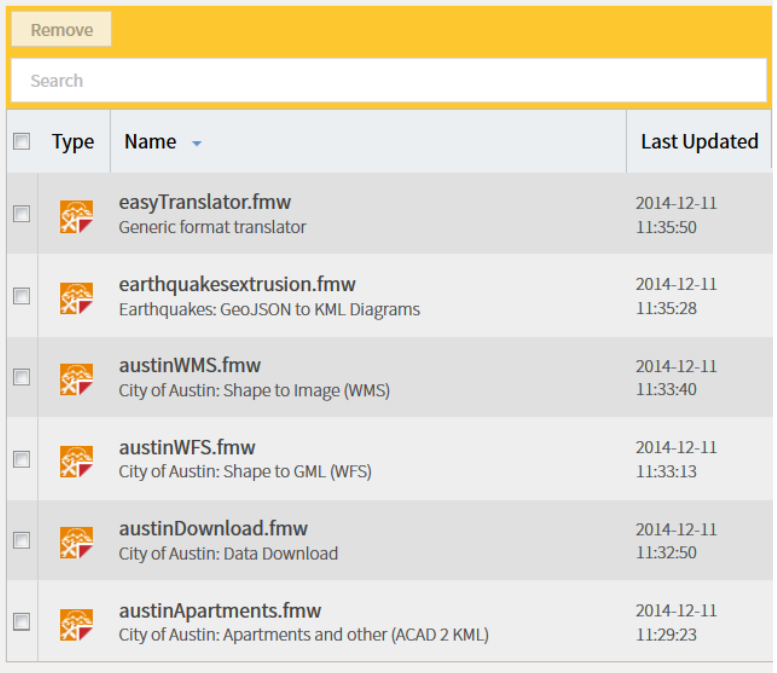


Welcome to FME Server

This will open a page that shows a list of the available FME Server workspace repositories. A repository is a method for storing and categorizing workspaces. It holds a number of workspaces in the same way that a folder holds a number of files.

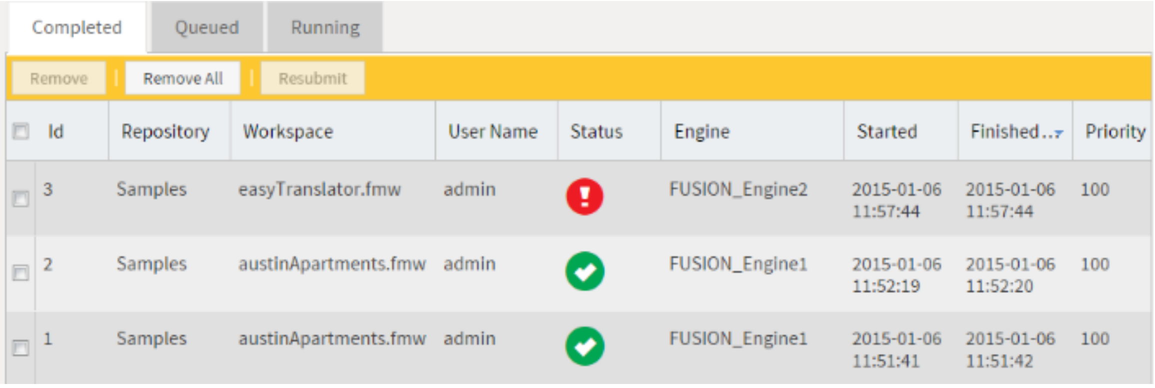
In this case there will be a repository for Samples and one for Utilities.

Click on the Samples repository. A list of workspaces in that repository is shown. These are workspaces that have been loaded into FME Server and are available to run.



☐	Type	Name	Last Updated
☐		easyTranslator.fmw Generic format translator	2014-12-11 11:35:50
☐		earthquakesextrusion.fmw Earthquakes: GeoJSON to KML Diagrams	2014-12-11 11:35:28
☐		austinWMS.fmw City of Austin: Shape to Image (WMS)	2014-12-11 11:33:40
☐		austinWFS.fmw City of Austin: Shape to GML (WFS)	2014-12-11 11:33:13
☐		austinDownload.fmw City of Austin: Data Download	2014-12-11 11:32:50
☐		austinApartments.fmw City of Austin: Apartments and other (ACAD 2 KML)	2014-12-11 11:29:23

5) Examine Jobs
Now click Manage, then Jobs. This tool shows you a list of jobs that have been run on the system, as well as jobs that are currently queued or currently running.



□	Id	Repository	Workspace	User Name	Status	Engine	Started	Finished	Priority
□	3	Samples	easyTranslator.fmw	admin	!	FUSION_Engine2	2015-01-06 11:57:44	2015-01-06 11:57:44	100
□	2	Samples	austinApartments.fmw	admin	✓	FUSION_Engine1	2015-01-06 11:52:19	2015-01-06 11:52:20	100
□	1	Samples	austinApartments.fmw	admin	✓	FUSION_Engine1	2015-01-06 11:51:41	2015-01-06 11:51:42	100

Jobs and Repositories are the two items that you will use for basic FME Server use and administration.

Sister Intuitive says…

“You might have noticed that the start menu has a tool for adding sample workspaces; but if you used the express installation then you don’t need to use this – everything is installed automatically.

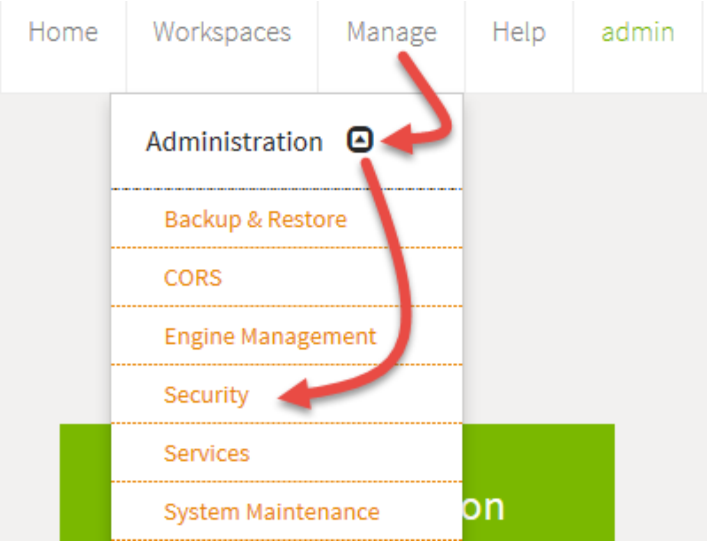
FME Server Security

FME Server Security

Roles and Users
Security Policies
Other Security Functions
Exercise 1b: Exploring Security Settings

FME Server Basic Training 2015

FME Server includes security to prevent unauthorized users from accessing services and other resources.



All security information for FME Server is stored in the repository database and is managed using the FME Server Web User Interface.

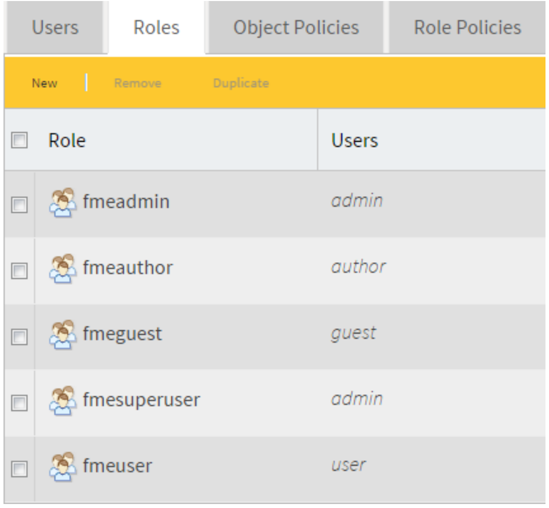
The key to setting up FME Server security is to have a clear plan on which services and accounts will be provided with open access, and which need to be restricted.

These policies are then set up using the Security section of the Web User Interface.

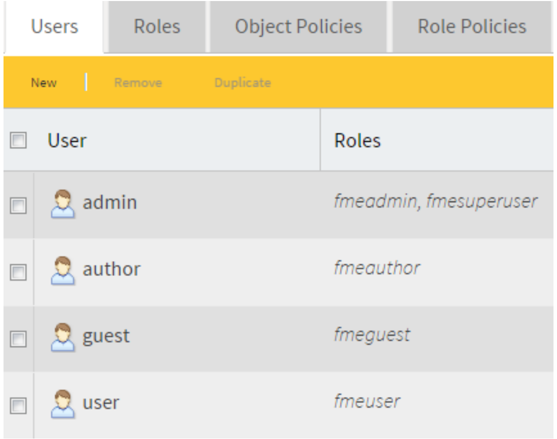
Roles and Users

Security is implemented using a system of roles and users. A role is best described as a function of the person; for example user, author, or administrator. Each user account is associated with one or more roles.

A default set of roles and user accounts is defined when FME Server is installed:



Role	Users
fmeadmin	admin
fmeauthor	author
fmeguest	guest
fmesuperuser	admin
fmeuser	user

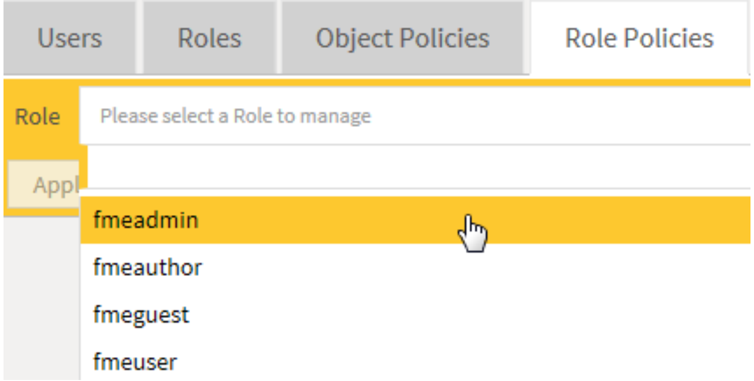


User	Roles
admin	fmeadmin, fmesuperuser
author	fmeauthor
guest	fmeguest
user	fmeuser

For example, in the previous example, if you logged in using the *admin* account, then you have both the role of *fmeadmin* and *fmesuperuser*.

Rather than apply security settings to individual users, FME Server allows you to apply them to specific roles. That way the settings can be changed for multiple users at once.

The role is selected under the Role Policies tab:

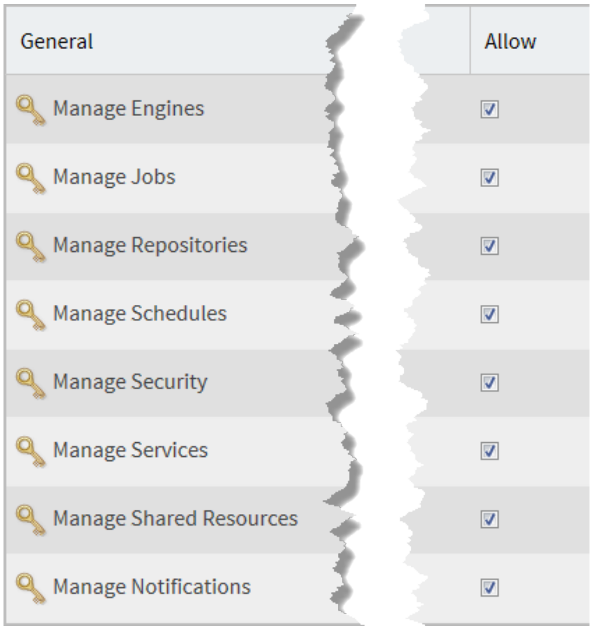


Security Policies
There are various sections of security policies that can be set for each role.

General
These are general policies such as the ability to manage jobs, services, and security. For example, if you don't have the ability to manage security, that item will be missing from the main menu of the user interface.

Topics
Topics are related to Notifications, which we'll cover later. Different capabilities – read, write, publish, remove – can be assigned for each topic you create.

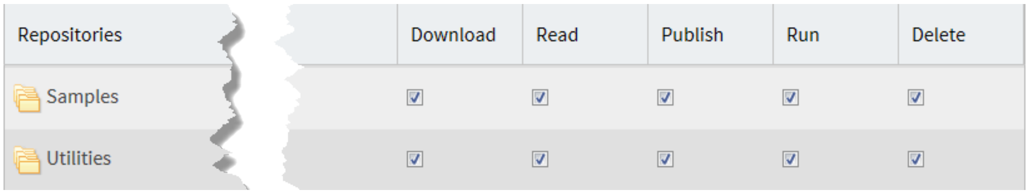
Resources
Resources are files and datasets stored on FME Server. Different permissions – read, write, upload, delete – can be assigned to each resource.



General	Allow
Manage Engines	☑
Manage Jobs	☑
Manage Repositories	☑
Manage Schedules	☑
Manage Security	☑
Manage Services	☑
Manage Shared Resources	☑
Manage Notifications	☑

Services
Services are key items of functionality on FME Server. They are the different methods by which a workspace can be run and output data delivered. Each role can be allowed – or not – to use a particular service.

Repositories
As you know, repositories are a place to store and categorize workspaces. Each role can be given different permissions for each and every repository.



Repositories	Download	Read	Publish	Run	Delete
Samples	☑	☑	☑	☑	☑
Utilities	☑	☑	☑	☑	☑

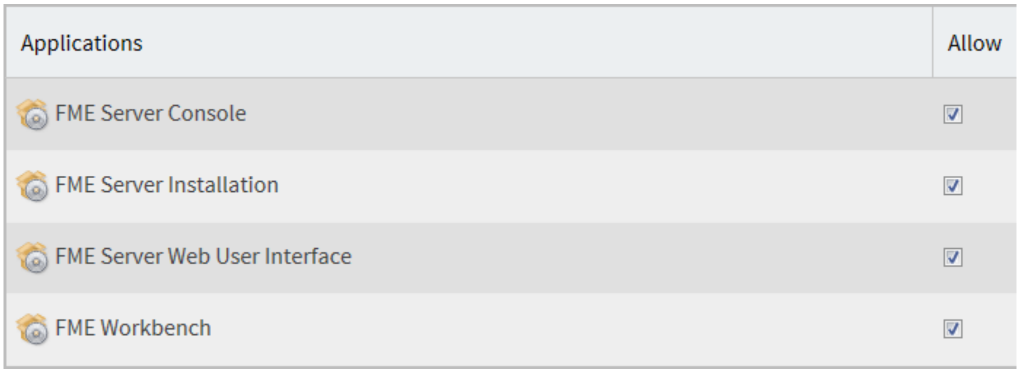
Components

Components are different pieces of workspace functionality – Readers, Writers, and Transformers – that can be controlled through security settings.

For example, you may wish to prevent a particular role from running workspaces on Server using the FMEServerJobSubmitter transformer.

Applications

Applications are the different ways in which FME Server can be accessed. For example, it's possible to control access to the FME Server Web Interface for a specific role.



Applications	Allow
FME Server Console	☑
FME Server Installation	☑
FME Server Web User Interface	☑
FME Workbench	☑

Each security policy can be set for a particular role, meaning it is applied to all users who are assigned to that role. If one user needs a specific set of security settings, different to any other user in that role, then you simply create a new role to apply to them.

Other Security Functions

FME Security can be fully integrated with both Active Directory and Integrated Windows Authentication (IWA). The FME Server Administrators Guide explains how this can be achieved.

Encryption

FME Server also allows Secure Sockets Layer (SSL) connections in order to encrypt transaction data. This is most important for ensuring sensitive log in information is not exposed and is particularly important when you are using Active Directory.

Again, the FME Server Administrators Guide explains the steps needed to implement this. In brief the process consists of two steps:

- Modify web service URLs to use HTTPS instead of HTTP
- Enable SSL on the web and/or application server.

Judge GIS says...
'I am the law! The FMESuperUser role is the high court of FME Server and is granted all permissions on all security settings. What's more, these permissions cannot be revoked, unset, or appealed against!

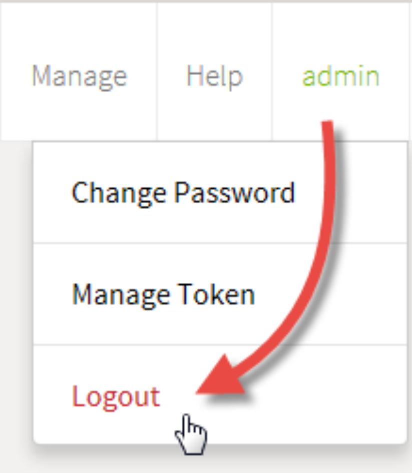
So, be sure not to assign accounts to the FMESuperUser role unless you really, really mean for them to be given that degree of power.'

Exercise 1b: Exploring Security Settings	
Scenario	FME Server administrator
Data	N/A
Overall Goal	Explore the FME Server Security Settings
Demonstrates	Using the FME Server web user interface Security Menu
Starting Workspace	None
Finished Workspace	None

Follow these steps to inspect different security settings.

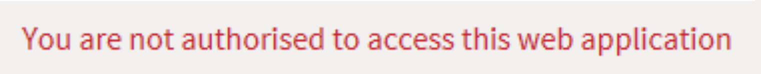
1) Connect to Server – 1
As in Exercise 1a, connect to the FME Server web user interface in your web browser.

If you are already logged in then log out using the tool in the top-right of the interface:



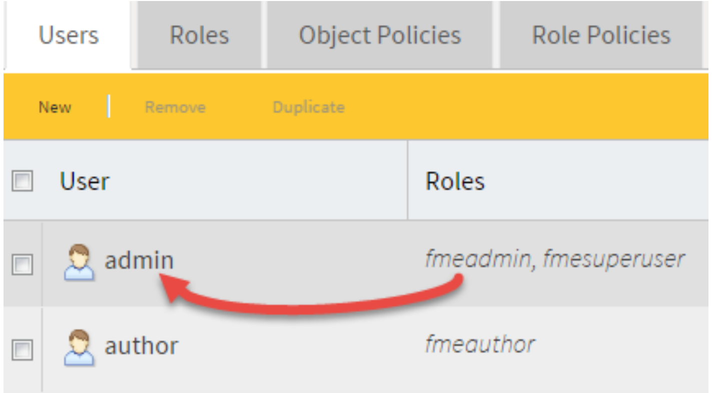
Now try to log in using the default guest account (Username:guest, Password: guest).

The log in will fail because – by default – this account does not have permission to access the web user interface:


You are not authorised to access this web application

2) Connect to Server – 2
Now log in using the default administrator account (Username: admin, Password: admin).

For the purposes of this exercise the account needs to be assigned to a role with permission to manage security. By default only the *fmeadmin* role has this capability, and the only account assigned to that role is *admin*.



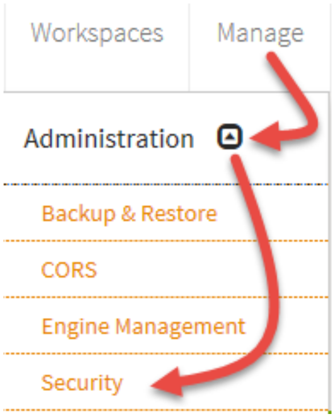
User	Roles
admin	*fmeadmin, fmesuperuser*
author	*fmeauthor*

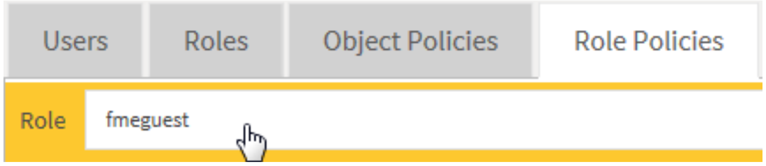
3) Open Security Settings

Click Manage on the top right side, and select Administration then Security:

If you don't see an option for Security then the account you have chosen does not have permissions to access this functionality.

Click on the Role Policies tab and choose the role *fmeguest*



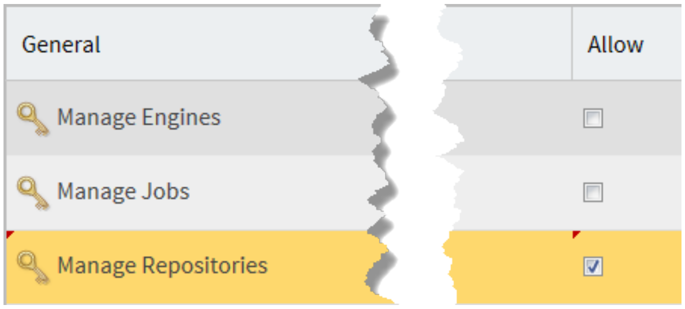


Role: fmeguest

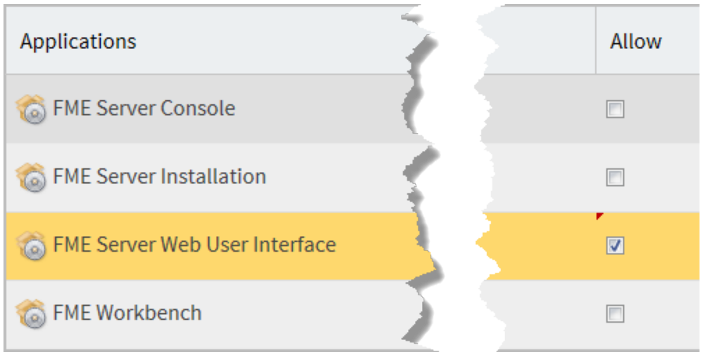
4) Examine Security

Examine the different security options for the fmeguest role. Notice, for example, that it can read the Samples repository, and run a workspace within it, but cannot publish a new workspace and cannot delete an existing one.

At the bottom of the list of security settings, locate Manage Repositories. Put a checkmark to allow the guest role to carry out this function.



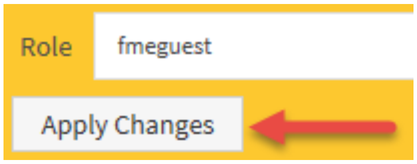
General	Allow
Manage Engines	□
Manage Jobs	□
Manage Repositories	☑



Applications	Allow
FME Server Console	□
FME Server Installation	□
FME Server Web User Interface	☑
FME Workbench	□

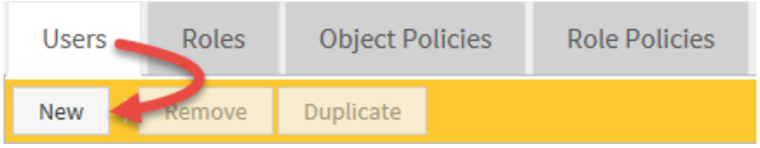
Scroll to the top of the list of security settings. Put a checkmark to allow the guest role to use the FME Server Web User Interface. **Also put a checkmark to allow the guest role to use 'Web Connection, REST Service'.**

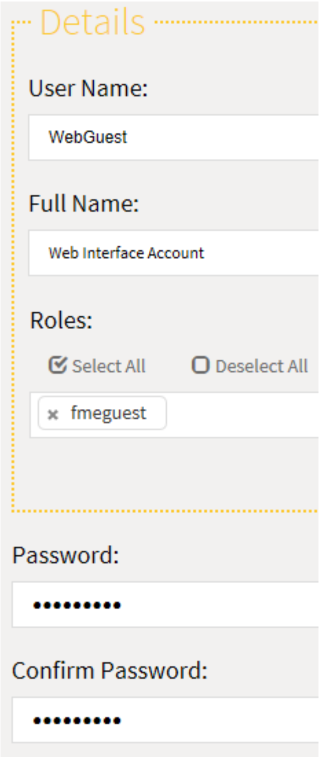
Click the Apply Changes button to apply the new security policies!



Role: fmeguest

5) Create New User
Now let's create a new user account. Click on the Users tab and then on the New button:





User Name: WebGuest

Full Name: Web Interface Account

Roles: × fmeguest

Password: •••••••••

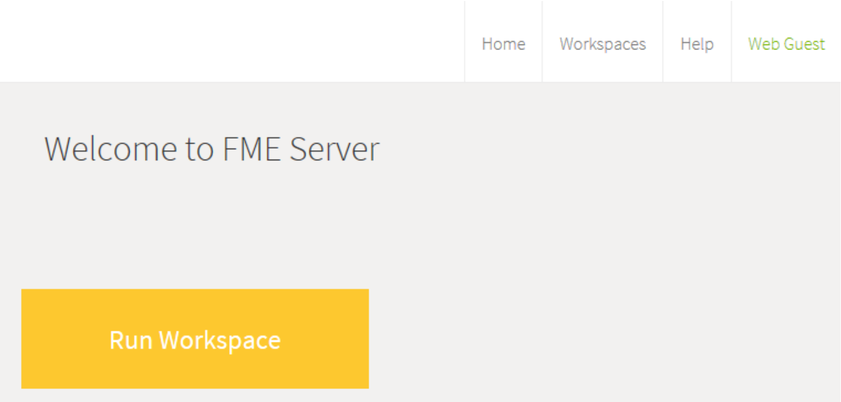
Confirm Password: •••••••••

When prompted, create a new user with the following parameters:

User Name:	WebGuest
Full Name:	Web Interface Account
Roles:	fmeguest
Password:	WebGuest1

6) Log In as New User
Now log out of the web user interface and log in again using the WebGuest account. You should be able to log in, because the setting was changed to allow guest users access to the web user interface.

Notice that the menu options will be severely restricted for this user! Only the Home, Workspaces and Help items will be available, because repositories are the only feature this role is allowed to manage.



Troubleshooting for Administrators

Troubleshooting for Administrators

Web User Interface

FME Server Basic Training 2015

This section shows a few basic troubleshooting techniques in case of emergency

Web User Interface

If you are unable to access the Web User Interface then the following suggestions may be of help.

- Confirm that FME Server is installed and running!

  The easiest way to be sure is to restart FME Server using *Start* > *Programs* > *FME Server* > *Windows Service* > *Restart*

- Check whether FME Server was installed using an application server port other than 80. For example, if port 80 was already being using the installer might have used a different port; 8080 is most common.

  To check, try entering the URL with this syntax *http://<host>:<port>/fmeserver* - for example *http://localhost:8080/fmeserver*

Module Review

Module Review

What you should have learned from this module

FME Server Basic Training 2015

This module introduced you to FME Server and the Web user Interface used to access most of its functionality

What You Should Have Learned from this Module

The following are key points to be learned from this module.

Theory

- FME Server is a powerful product for handling large volumes of data at the **enterprise** level. It has three core capabilities: Self-Serve, Real-Time, and Automation.
- FME Server provides the **same processing core** as FME Desktop, but on the enterprise level. It does not include an authoring tool, for which FME Workbench is required.
- Someone who creates translations for use on FME Server is called an **author**.
- The three main components of FME Server are **FME Engines,** the **Server Core**, and **Web Services**
- FME Server can be implemented on **Physical hardware**, **Virtual hardware**, or under a product called **FME Cloud**.
- FME Server security is implemented using a system of **roles** and **users**.

FME Skills

- The ability to open the FME Server Web User Interface
- The ability to confirm the version of FME Server and check that FME core engines are available
- The ability to browse repositories
- The ability to check job status
- The ability to check and manage security settings

Running Workspaces on FME Server

FME Workbench and FME Server

FME Workbench and FME Server

FME Desktop as an Authoring Tool
The Workspace Authoring Workflow

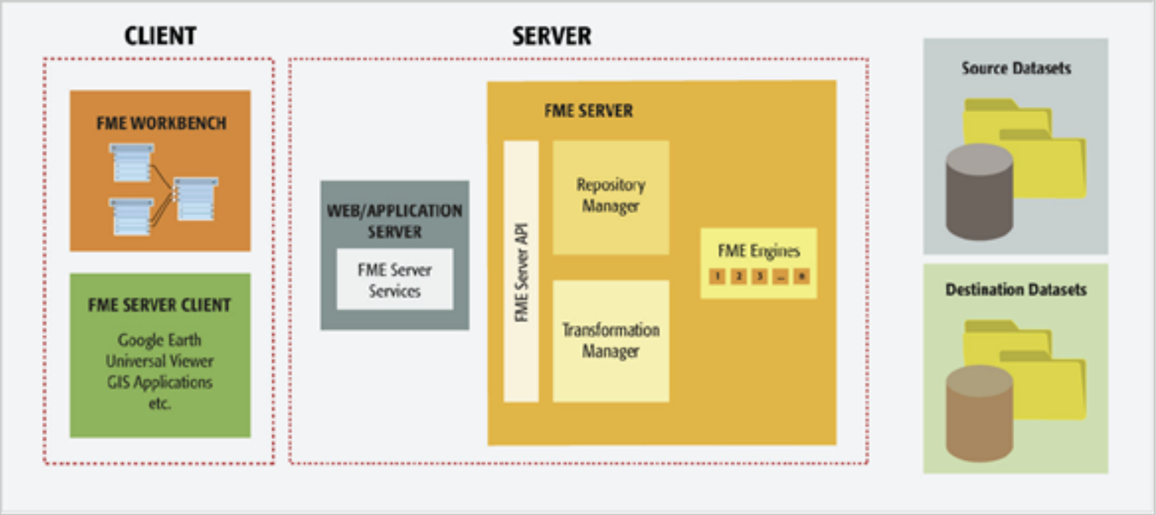
FME Server Basic Training 2015

FME Workbench is the authoring environment for FME Server translations. It has functionality specifically for FME Server.

FME Desktop as an FME Server Authoring Tool

FME Server can be called a model-driven architecture, because the specification for a translation is expressed as a model. In FME, these models are better known as **Workspaces**.

Workspaces are created – we call it "authored" – using FME Desktop. In particular, the **FME Workbench** application is used. FME Workbench is a client of FME Server, and so they form a *client-server pair*. However, both share the same core engine and process data the same way.



The Workspace Authoring Workflow

The four basic steps to creating and using a workspace on FME Server are:

- Author the workspace in FME Workbench.
- Publish the workspace from FME Workbench to FME Server.
- Run the workspace on FME Server.
- Maintain the workspace by downloading it from FME Server, making any required updates in FME Workbench, and re-publishing it back to FME Server.

Ms. Analyst says…

"Be aware that FME Workbench is part of the FME Desktop product and is not included as part of FME Server."

Workspace Management

Workspace Management

Transferring Workspaces
Publishing a Workspace
Re-Publishing a Workspace
Downloading a Workspace
Repositories
Exercise 2a: Publishing a Workspace

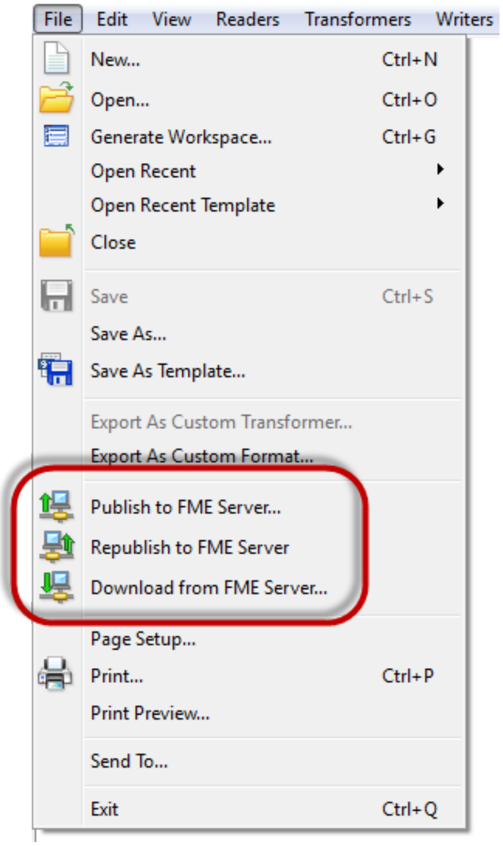
Because Workbench is a client of FME Server, workspaces may be transferred to and from an FME Server repository.

Transferring Workspaces

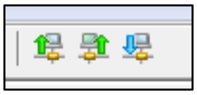
FME Workbench has the ability to:

- Publish a workspace to FME Server
- Republish a workspace to FME Server
- Download a workspace from FME Server

These three capabilities are accessed in FME Workbench either through the menubar…



…or the toolbar:



Professor Lynn Guistic says…

“Published workspaces are held on FME Server in a file-based ***repository****. Each FME Server may have multiple repositories, but any workspace can only belong to one of them.*

All metadata related to the workspace is held separately in FME Server’s repository database. This metadata includes information about source and destination datasets, workspace feature types, and published parameters.”

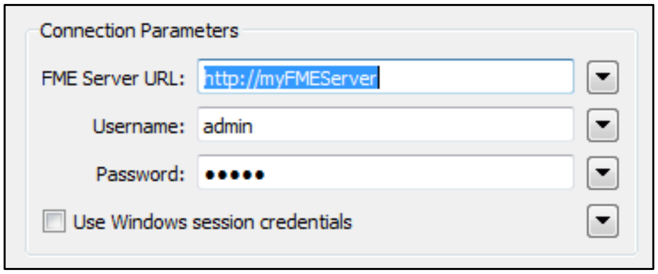
Publishing a Workspace

Once a workspace has been authored it is “published” to FME Server. A workspace is published using a simple wizard interface that connects using the FME Server REST API.

Connecting to Server

The first dialog in the publishing wizard is where a connection is defined.

Workbench can connect to FME Server by web connection (using HTTP and the REST API).



Connection Parameters

FME Server URL: http://myFMEServer

Username: admin

Password: •••••

□ Use Windows session credentials

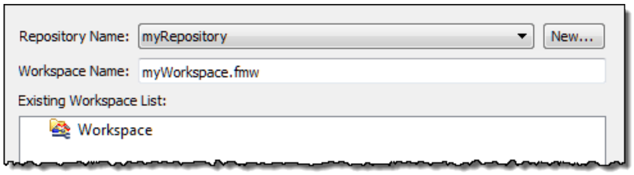
The web connection requires the URL of the Server.

HTTP is used when the FME Server is not on the same Local Area Network, but on a web-accessible server.

The dialog also provides fields in which to define connection credentials. Note the ability to save time by reusing Windows session credentials

Repository Selection

The next dialog defines the repository for the workspace:



Repository Name: myRepository

Workspace Name: myWorkspace.fmw

Existing Workspace List:

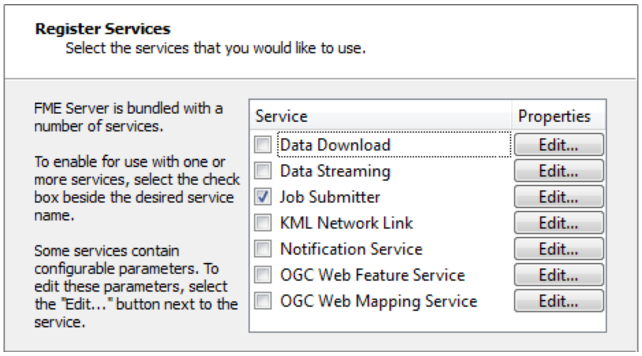
Workspace

Either an existing repository can be used, or a new one created. The workspace name can also be edited, though it must always conclude with “.fmw”.

Workspace Registration

The final dialog defines which services the workspace is to be registered against.

The Job Submitter service allows FME Server to run a workspace as-is. This is the closest to running a workspace on FME Desktop. All inputs and outputs are defined in the workspace so data is simply written out and not streamed or delivered in any other manner.



Register Services

Select the services that you would like to use.

FME Server is bundled with a number of services.

To enable for use with one or more services, select the check box beside the desired service name.

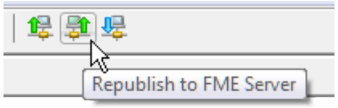
Some services contain configurable parameters. To edit these parameters, select the "Edit..." button next to the service.

Service	Properties
□ Data Download	Edit...
□ Data Streaming	Edit...
☑ Job Submitter	Edit...
□ KML Network Link	Edit...
□ Notification Service	Edit...
□ OGC Web Feature Service	Edit...
□ OGC Web Mapping Service	Edit...

Job submission is ideal for testing workspaces, and for running large-scale and batch translations that make use of the server process queue.

Republishing a Workspace

Once a workspace has been published, the republish tool becomes active.
Further updates to the workspace (within the same session) can then be uploaded with a single click:

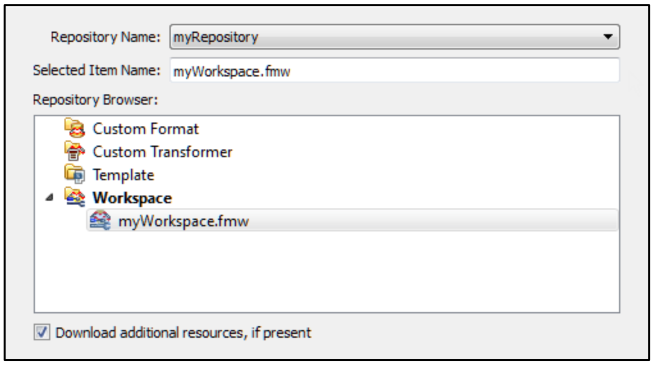


The same parameters are used as before. If changes need to be made to these parameters, then the full publishing wizard should be used.

Downloading a Workspace

Workbench can also "download" a workspace held in an FME Server repository. This is usually done in order to make edits to the workspace. Note that downloaded workspaces are copies of the original, which remains in the FME Server repository.

The downloading wizard begins with the same connection dialog as the publishing wizard. From there, the second – and final – dialog page is a repository and workspace selection tool:



Repository Name: myRepository

Selected Item Name: myWorkspace.fmw

Repository Browser:

- Custom Format
- Custom Transformer
- Template
- Workspace
  - myWorkspace.fmw

☑ Download additional resources, if present

The user is then prompted for a location to save the workspace. The default is *<user>/FME/My FME Server Workspaces*. The workspace – and resources – are then downloaded.

Once downloaded, the workspace is automatically opened within Workbench for editing.

To avoid entering the same parameters every time a connection is made, set the parameters, click the Defaults button, and select 'Save as My Defaults'. Now these parameter values will be used automatically every time the connection dialog is opened.

Ms. Analyst says…

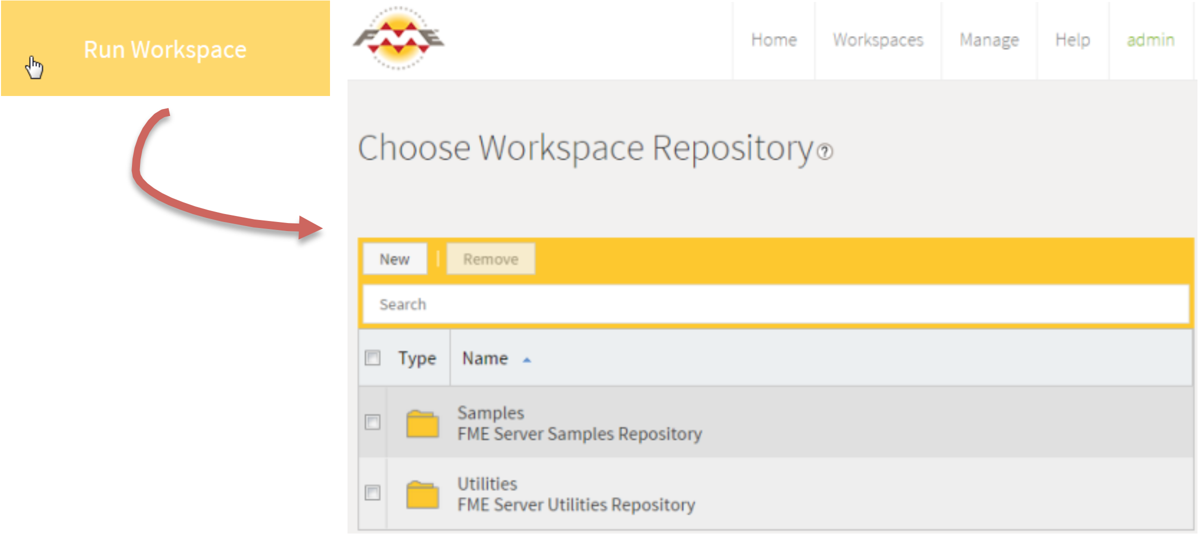
"Besides workspaces, it's also possible to publish/download FME custom transformers and custom formats to/from a server repository."

Repositories

Repositories are storage facilities for FME Server workspaces.

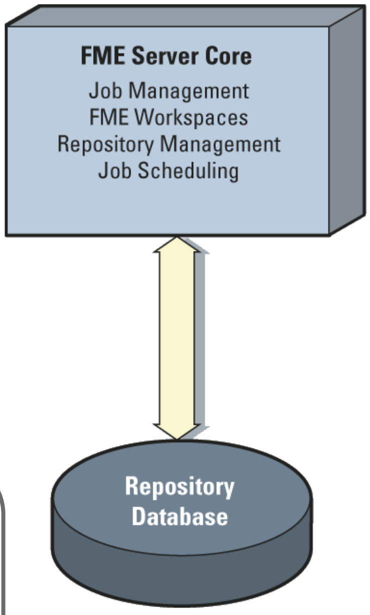
You can think of each repository as being like a folder on a file-system. Like a folder, it stores workspace files. Also like a folder, access rights can be granted to individuals and groups.

Repositories can be accessed through the web user interface using the Run Workspace button on the main menu:



Choose Workspace Repository

□	Type	Name
□		Samples FME Server Samples Repository
□		Utilities FME Server Utilities Repository



Although the workspaces are stored on the file-system (C:\apps\FMEServer\Server\repository) information about each workspace is stored in an internal database within FME Server.

For interested experts, the FME Server core communicates with the database via JDBC over TCP/IP.

The contents of this database should **never** be edited directly.

Querying the database is permissible, although extracting repository information is more commonly achieved using the REST API.

Police Chief Webb-Mapp says…

"Security in FME Server is very important, and never more so than for repositories.

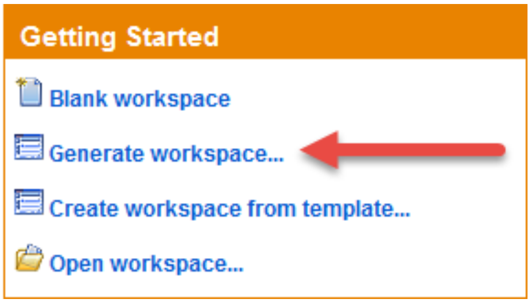
For each repository you create, check the security settings. If you don't then end-users may not get access to the repository!"

Exercise 2a: Publishing a Workspace	
Scenario	Airphoto Data Vendor
Data	GeoTiff Orthophotos
Overall Goal	Create a workspace, publish it to Server
Demonstrates	Publishing a workspace to FME Server
Starting Workspace	None
Finished Workspace	C:\ FMEData2015\Workspaces\ServerAuthoring\Exercise2a-Complete.fmw

This simple exercise will refresh your memory on FME Workbench and demonstrate how to publish a workspace. In this scenario you are an FME user working for a company producing and selling airphotos. You want to publish a workspace to FME Server to translate GeoTiff datasets to JPEG.

1) Generate Workspace
In the FME Workbench start screen, select the option to Generate Workspace

Alternatively you can select File > New > Generate from the menubar or use the shortcut Crtrl + G.

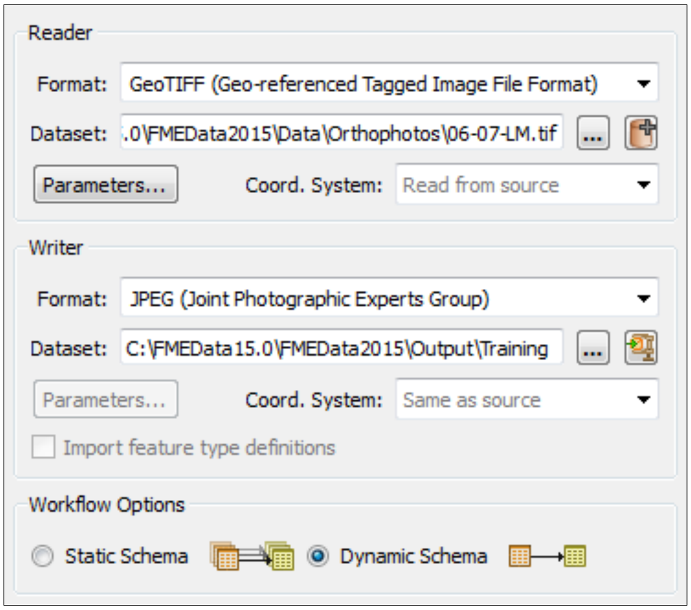


When prompted, set up the workspace using the following parameters:

Reader Format	GeoTIFF (Geo-referenced Tagged Image File Format)
Reader Dataset	C:\FMEData2015\Data\Orthophotos\06-07-LM.tif
Writer Format	JPEG (Joint Photographic Experts Group)
Writer Dataset	C:\FMEData2015\Output\Training

Before clicking OK, select the option for Dynamic Schema at the bottom of the dialog.

This will allow the workspace to process any GeoTiff file, not just the one originally chosen.



Reader

Format: GeoTIFF (Geo-referenced Tagged Image File Format)

Dataset: .0\FMEData2015\Data\Orthophotos\06-07-LM.tif

Coord. System: Read from source

Writer

Format: JPEG (Joint Photographic Experts Group)

Dataset: C:\FMEData15.0\FMEData2015\Output\Training

Coord. System: Same as source

☐ Import feature type definitions

Workflow Options

○ Static Schema ● Dynamic Schema

2. Save and Run Workspace
Save the workspace as Exercise2a.fmw (using File > Save from the menubar) and then run it (using Run > Run Workspace). Browse through the file system to confirm that you have written a JPEG file to the output directory.

Professor Spatial says…

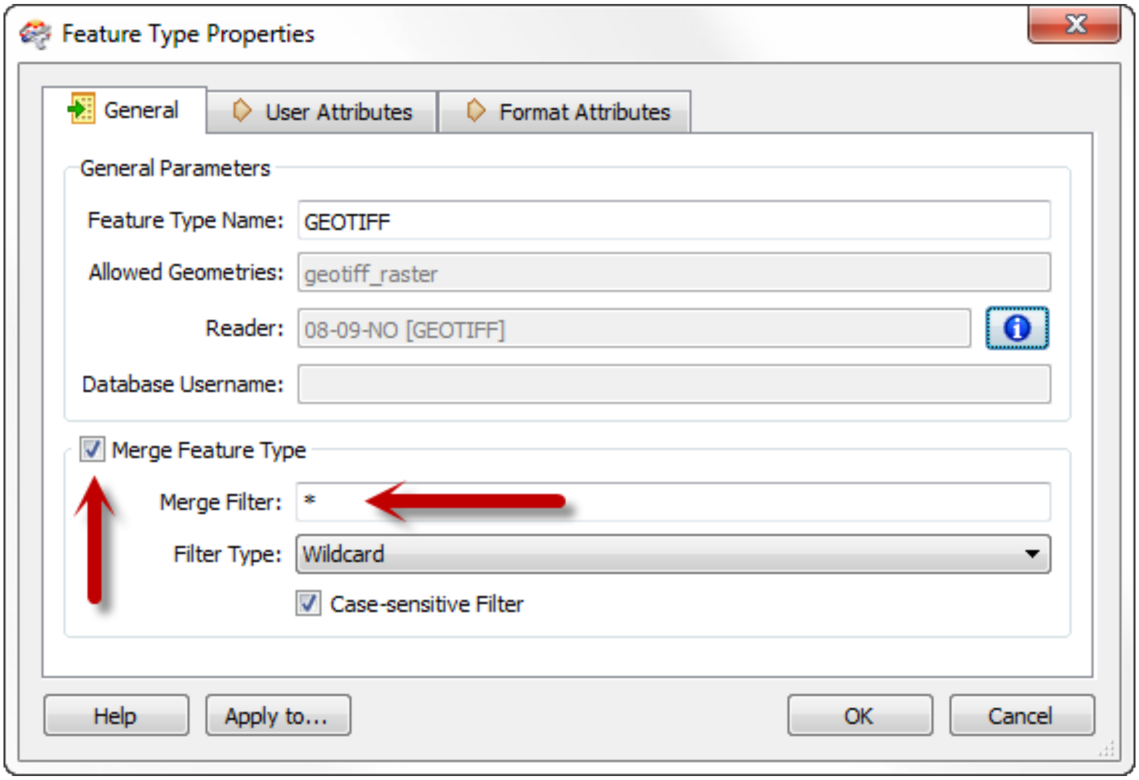
"You should always run your workspace first in Workbench, before publishing it to FME Server.

This is the easiest way to ensure the workspace functions as expected. If it fails in Workbench, it won't suddenly work on FME Server!"

As a further test, run the workspace a second time but this time using Run > Prompt and Run. This action prompts you to select the source and destination datasets, instead of simply using those defined in the workspace.

Select a different file this time (say, 08-09-NO.tif) and open the output to prove that the contents are different to before.

NB: *If you missed setting a Dynamic Schema in step 1, you can either recreate the workspace (it will only take a minute) or – if you know how – open the Reader feature type properties dialog and set a merge filter.*



General Parameters

Feature Type Name: GEOTIFF

Allowed Geometries: geotiff_raster

Reader: 08-09-NO [GEOTIFF]

Database Username:

☑ Merge Feature Type

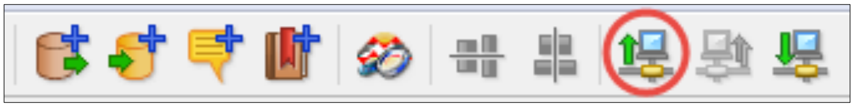
Merge Filter: *

Filter Type: Wildcard

☑ Case-sensitive Filter

3. Publish Wizard (Step 1)
Workspaces are published to FME Server using a wizard.

Choose File > Publish to FME Server from the menubar (or select the same tool on the toolbar).



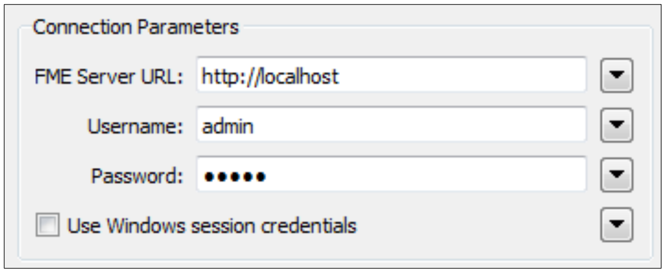
In the first dialog of the wizard you are prompted to enter connection parameters to FME Server.

Enter the parameters provided by your training instructor.

In most cases the parameters will be as follows:

FME Server URL: http://localhost

Username: admin
Password: admin



Connection Parameters

FME Server URL: http://localhost

Username: admin

Password: •••••

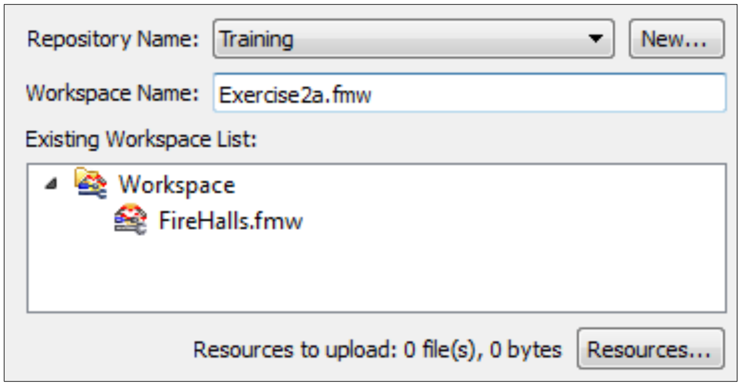
□ Use Windows session credentials

Click **Next** to continue. If the credentials are correct a connection will be made and you will move on to the next dialog in the wizard.

4. Publish Workspace (Step 2)
The next dialog prompts you to choose a repository in which to store the workspace.

For this exercise we'll create a new repository by clicking the New button. When prompted enter the name Training.

Click **OK** to close the Create New Repository dialog. Uncheck the box labeled Upload Data Files, and then click **Next** to continue the wizard.



Repository Name: Training

Workspace Name: Exercise2a.fmw

Existing Workspace List:

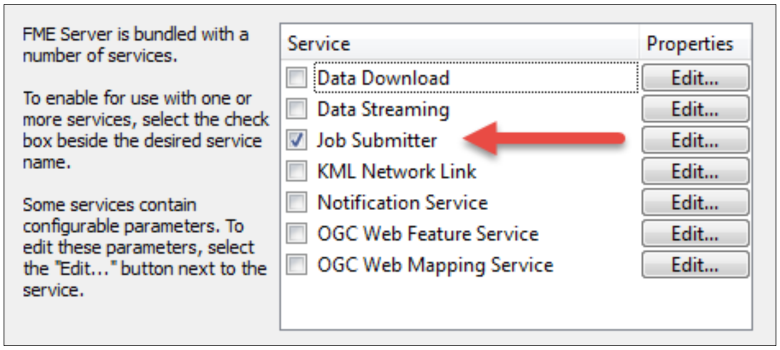
- Workspace
  - FireHalls.fmw

Resources to upload: 0 file(s), 0 bytes

5. Publish Workspace (Step 3)

In the final screen of the wizard we can register the workspace for use with various services.

Select the Job Submitter service as this is the only service we are using for now and click **Publish** to complete publishing the workspace.



FME Server is bundled with a number of services.

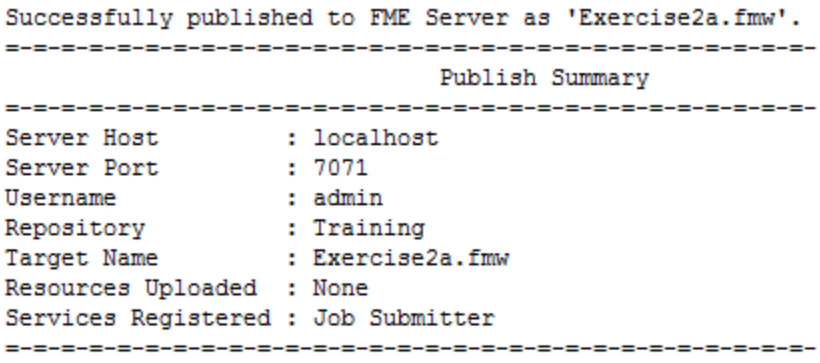
To enable for use with one or more services, select the check box beside the desired service name.

Some services contain configurable parameters. To edit these parameters, select the "Edit..." button next to the service.

Service	Properties
□ Data Download	Edit...
□ Data Streaming	Edit...
☑ Job Submitter	Edit...
□ KML Network Link	Edit...
□ Notification Service	Edit...
□ OGC Web Feature Service	Edit...
□ OGC Web Mapping Service	Edit...

After a workspace is transferred to Server, the log window displays a message reporting which workspace has been published to which repository and for which services.

In this exercise, the workspace Exercise2a.fmw has been published to the Training repository for use in the Job Submitter service.



```
Successfully published to FME Server as 'Exercise2a.fmw'.
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
                               Publish Summary
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
Server Host         : localhost
Server Port         : 7071
Username            : admin
Repository          : Training
Target Name         : Exercise2a.fmw
Resources Uploaded  : None
Services Registered : Job Submitter
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
```

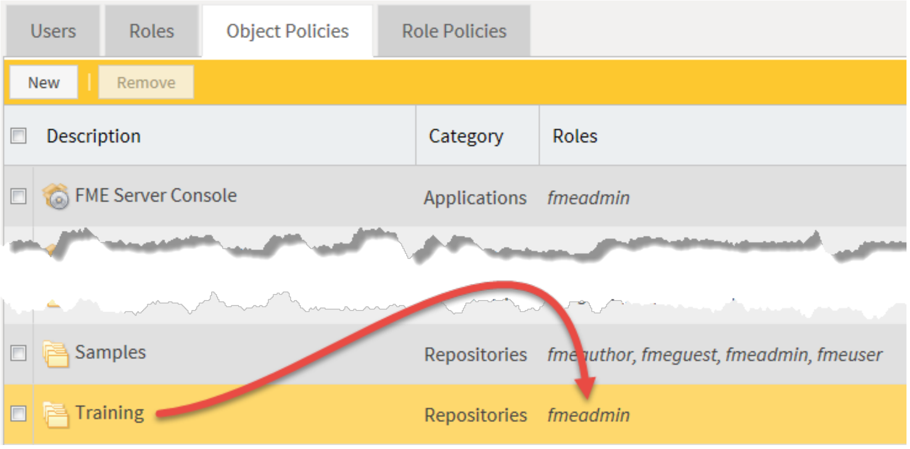
Congratulations! You have now created a workspace and published it to FME Server.

Advanced Task
Open the FME Server Web User Interface in a web browser and log in using the same connection parameters as you published the workspace with.

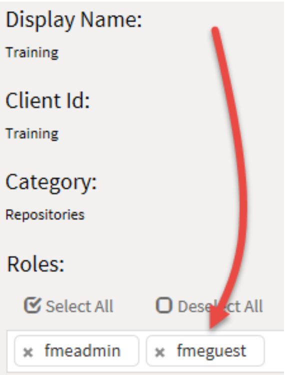
Click Security on the main menu to open security settings, and then click the Object Policies tab.

Locate the Training repository in the list of security objects. You should see that the roles allowed to make use of it is restricted to the one(s) to which the publisher's account belongs.

In this case the user was *admin*, which is associated with the *fmeadmin* role. Therefore only an administrator will be able to make use of the repository that we have just created.



□ Description	Category	Roles
□ FME Server Console	Applications	*fmeadmin*
□ Samples	Repositories	*fmeauthor, fmeguest, fmeadmin, fmeuser*
□ Training	Repositories	*fmeadmin*



Display Name:
Training

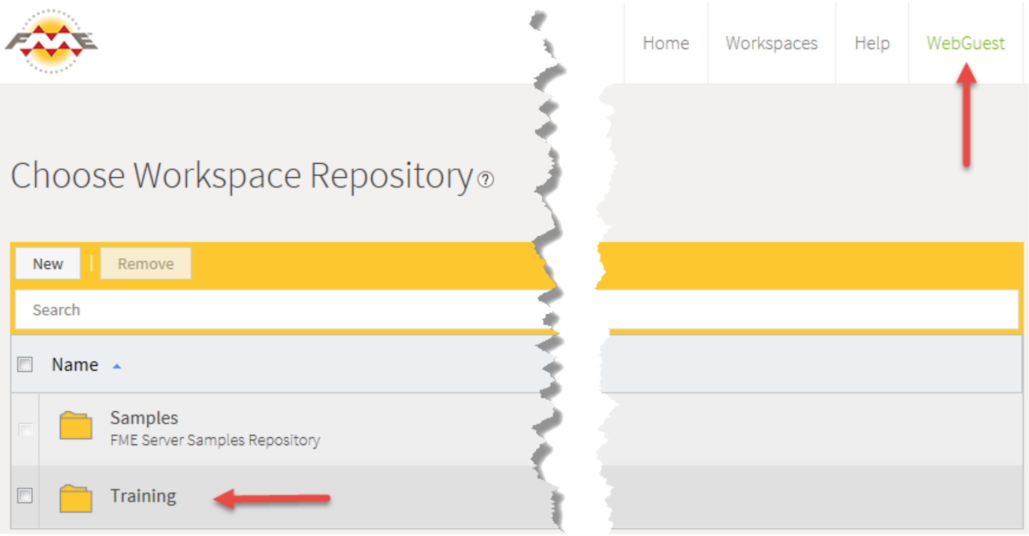
Client Id:
Training

Category:
Repositories

Roles:

Click on the repository object to open its security policies. Add the *fmeguest* role so that users with a guest account can access this repository.

If you created a guest account (WebGuest) in Exercise 1b, then log out and log in again using that account, to prove that guests now have access to the repository.



Samples
FME Server Samples Repository

Training

Source Data Management

Source Data Management

Accessing Resources
Resource Functionality
Exercise 2b: Uploading to a Resources Folder

Source data may be published to FME Server and stored in a "Resources" folder, where it is available to any workspace

FME Server has several mechanisms for uploading and storing resources for workspaces to use in a translation; for example the source data for a translation.

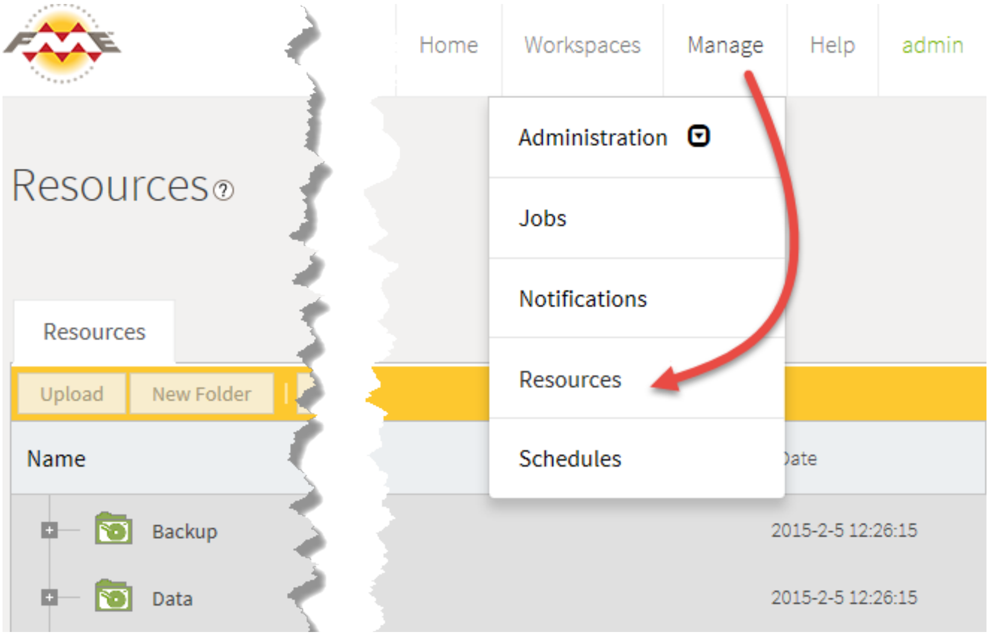
One such mechanism is called "Resources" or "Resource Management", and is accessed through the Web User Interface.

The advantage of these Resources, compared to other methods, is that data can be uploaded and referenced by any workspace, without having to upload it every time. This is particularly useful when access to the file system is restricted (like on an FME Cloud system).

If the workspace author does have access to the file system, then Resources are still useful, as a Resources folder can be mapped and shared among many users as a physical drive.

Accessing Resources

Resources are accessed through the Manage, then Resources on the main menu.



Resource Functionality

The various resource functions are:

- Upload	- Download	- Copy
- New Folder	- Delete	- Move

More information on this functionality is proved in Chapter 3 under **Data Upload**.

Exercise 2b: Uploading to a Resources Folder	
Scenario	Airphoto Data Vendor
Data	GeoTiff Orthophotos
Overall Goal	Uploading data to be used by the previous workspace
Demonstrates	FME Server resource management
Starting Workspace	n/a
Finished Workspace	n/a

In this exercise we'll upload some GeoTiff files as Resources on FME Server. This will make them available for use by a workspace, including the one published in Exercise 2a.

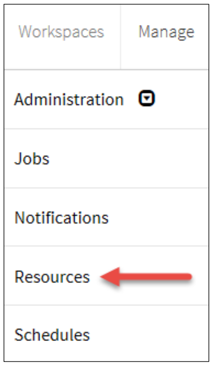
1. Open Web User Interface

In a web browser open the FME Server Web User Interface, as we did in Chapter 1, by entering http://<servername>/fmeserver

Alternatively use the Windows shortcut Start > Programs > FME Server 2015.1 > Web User Interface.

Login in using the credentials provided by the instructor, by default these will be

User	admin
Password	admin

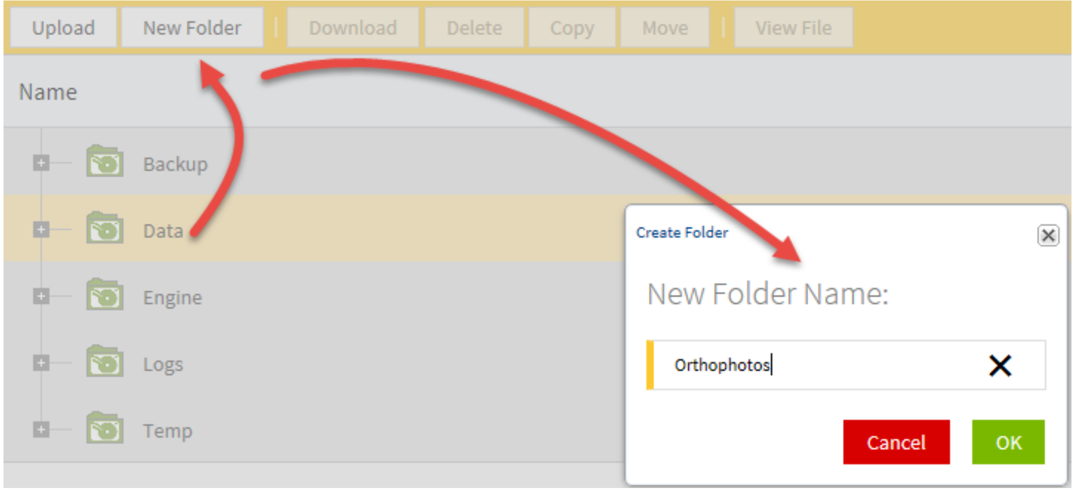


2. Create New Resources Folder

Under Manage, click Resources, on the upper right menu.

Click on the Data Folder in the Resource Management dialog.

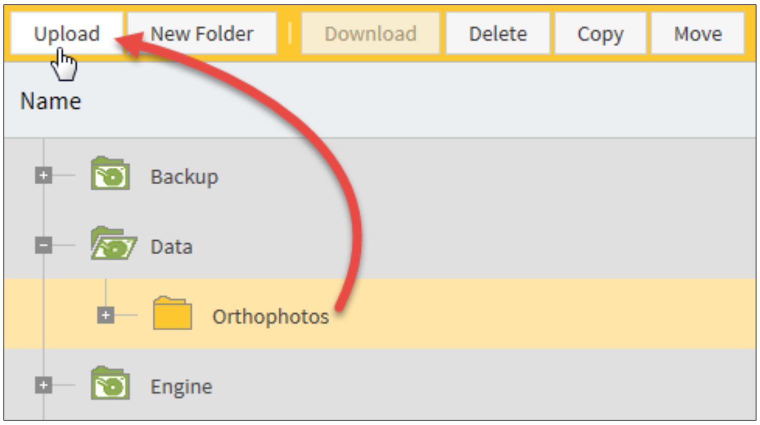
Click New Folder to create a new subfolder inside Data. Name the folder Orthophotos.



Orthophotos

3. Upload Image Data

Still in the Resources Management dialog, select the newly created Orthophotos folder. Click Upload.

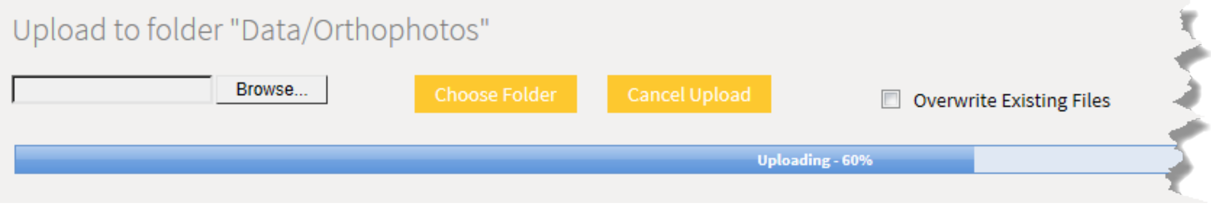


Click Browse. A file browser window will open. Browse to the folder C:\FMEData2015\Data\Orthophotos and select the following GeoTiff files for upload.

08-09-NO.tif
08-09-LM.tif
06-07-NO.tif
06-07-LM.tif

NB: *If you wish, you can choose **all** of the files; but they total 184mb in size, so uploads and downloads will take longer.*

Click Open and the selected files will be uploaded.


Upload to folder "Data/Orthophotos"
Browse...
Choose Folder
Cancel Upload
□ Overwrite Existing Files
Uploading - 60%

Congratulations! You have uploaded a series of source datasets, which are now available for use in any FME Server Workspace.

Running a Workspace on FME Server

Running a Workspace on FME Server

- Running the Job Submitter using the Web Interface
- Configuring a Translation
- Running the Job Submitter using a URL
- Exercise 2c: Running a Workspace

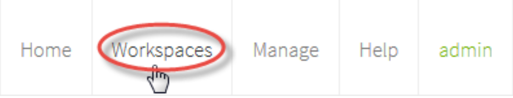
This section introduces running a workspace on FME Server

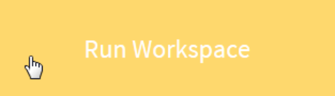
Once a workspace is uploaded to FME Server, it can be run on that server.
There are several ways to run a workspace, the simplest being using the Job Submitter service.
This service can be accessed either through the Web User Interface or via a URL

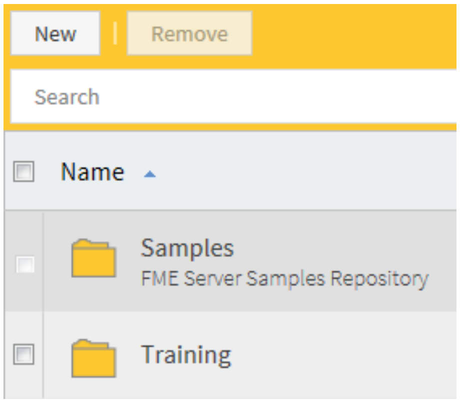
Running the Job Submitter using the Web Interface

Using the web interface to run a workspace is very similar to running it in FME Workbench; you simply locate the workspace and click a Run button.

To locate a workspace first click on Workspaces or the Run Workspace entry in the main menu of the web interface.

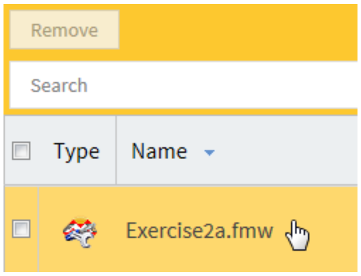


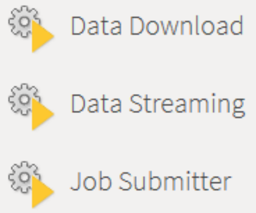




Secondly, click on the repository that contains the workspace to be run.

Next, click on the workspace itself.



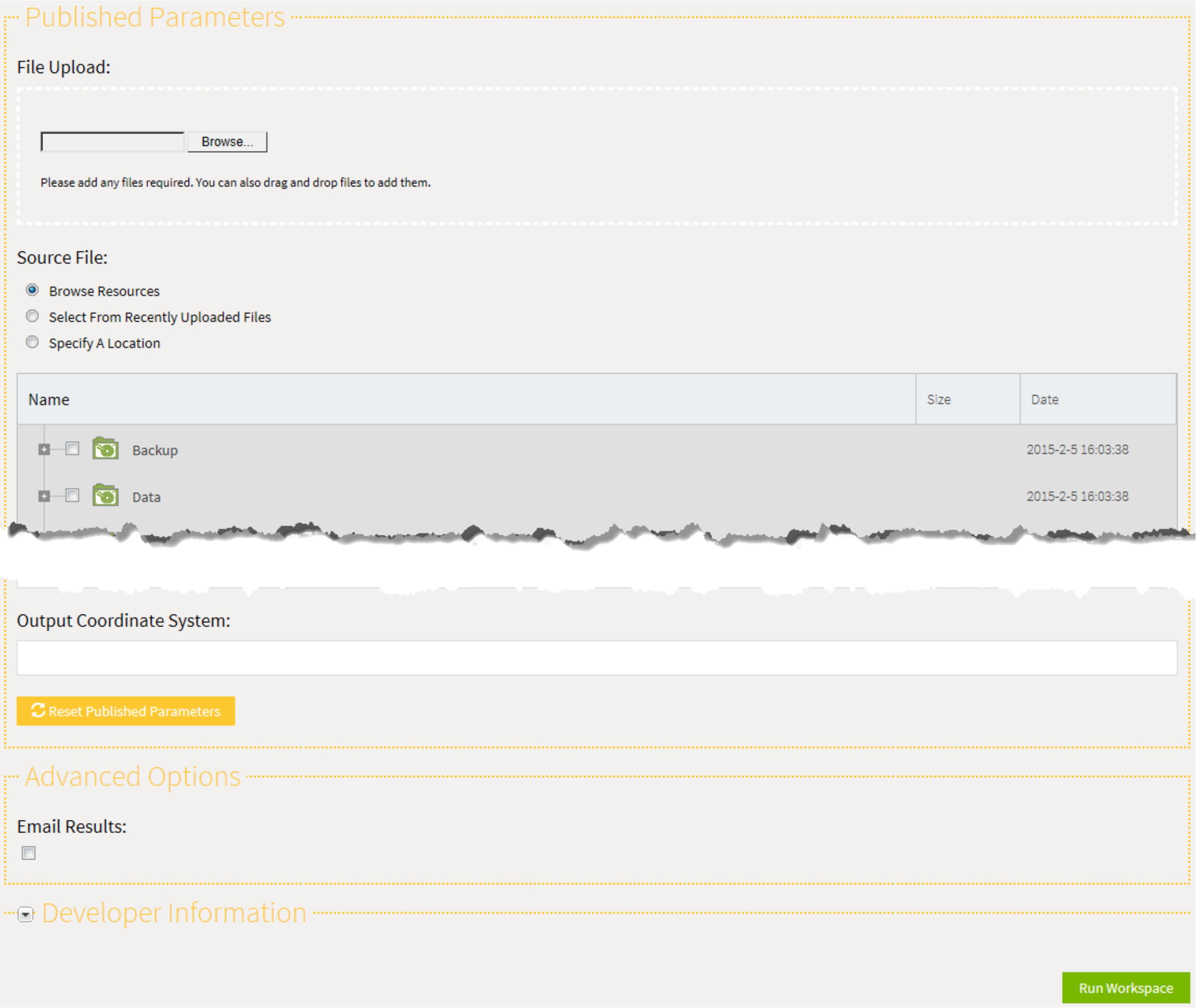


Finally you have the option to click which service to configure or run.

Configuring a Translation

Clicking on a service for a workspace is the equivalent to the Prompt and Run option in FME Workbench. You get to define what parameters are going to be set for the translation, such as File to Upload, Source File and Output Coordinate System.

For example, here the user can specify where the source data is available:



Published Parameters

File Upload:

Browse...

Please add any files required. You can also drag and drop files to add them.

Source File:

● Browse Resources
○ Select From Recently Uploaded Files
○ Specify A Location

Name	Size	Date
Backup		2015-2-5 16:03:38
Data		2015-2-5 16:03:38

Output Coordinate System:

Reset Published Parameters

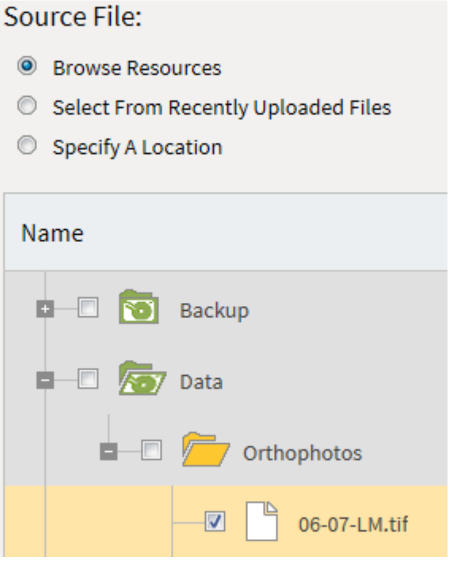
Advanced Options

Email Results:

□

Developer Information

Run Workspace



Source File:

● Browse Resources
○ Select From Recently Uploaded Files
○ Specify A Location

Name

□ Backup
□ Data
□ Orthophotos
☑ 06-07-LM.tif

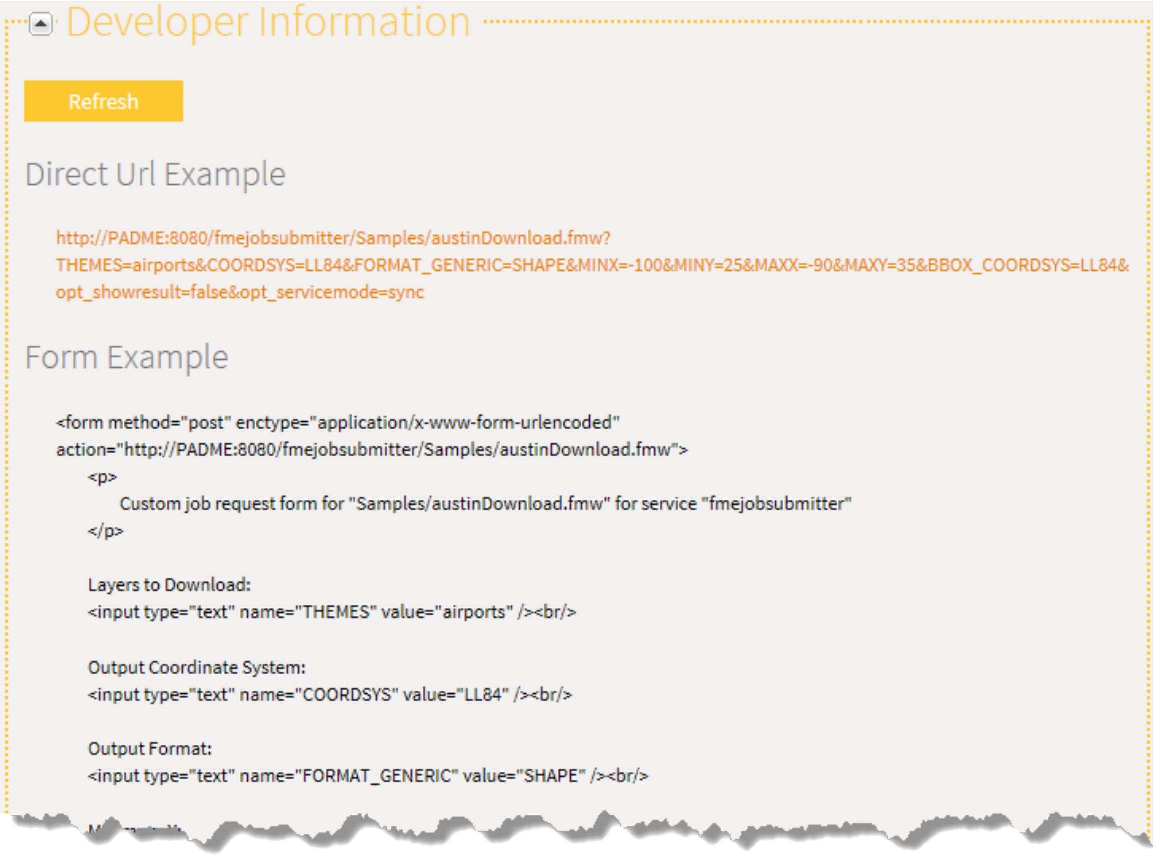
If data was made available using Resources functionality, then the user could choose the Browse option to search for and select it:

Once the parameters are set, click the **Run Workspace** button to complete the process.

Running the Job Submitter using a URL
All job requests to an FME Server are a variation on an HTTP request. This makes running a workspace via a URL very simple, provided you know what form the request will take.

The easiest way to find that URL is through the workspace services in the FME Server web interface.

Notice that the service configuration dialog includes a section labelled “Developer Information”. Once expanded, an example URL and example form are displayed.



Direct Url Example

```
http://PADME:8080/fmejobsubmitter/Samples/austinDownload.fmw?
THEMES=airports&COORDSYS=LL84&FORMAT_GENERIC=SHAPE&MINX=-100&MINY=25&MAXX=-90&MAXY=35&BBOX_COORDSYS=LL84&
opt_showresult=false&opt_servicemode=sync
```

Form Example

```
<form method="post" enctype="application/x-www-form-urlencoded"
action="http://PADME:8080/fmejobsubmitter/Samples/austinDownload.fmw">
    <p>
        Custom job request form for "Samples/austinDownload.fmw" for service "fmejobsubmitter"
    </p>

    Layers to Download:
    <input type="text" name="THEMES" value="airports" /><br/>

    Output Coordinate System:
    <input type="text" name="COORDSYS" value="LL84" /><br/>

    Output Format:
    <input type="text" name="FORMAT_GENERIC" value="SHAPE" /><br/>
```

The URL shown uses a HTTP GET request whereas the HTML form uses a HTTP POST request.

This information is a useful tool for building your own web applications that access FME Server services, because you can copy the HTTP request and embed it on your own website or application.

You could also embed the URL or form into an email, or paste the URL directly into a web browser.

t!p

There are limits to the amount of data you can send in a GET request because URLs have length restrictions that vary depending on the browser being used. If you anticipate that your request parameters may include very long strings, use a POST request.

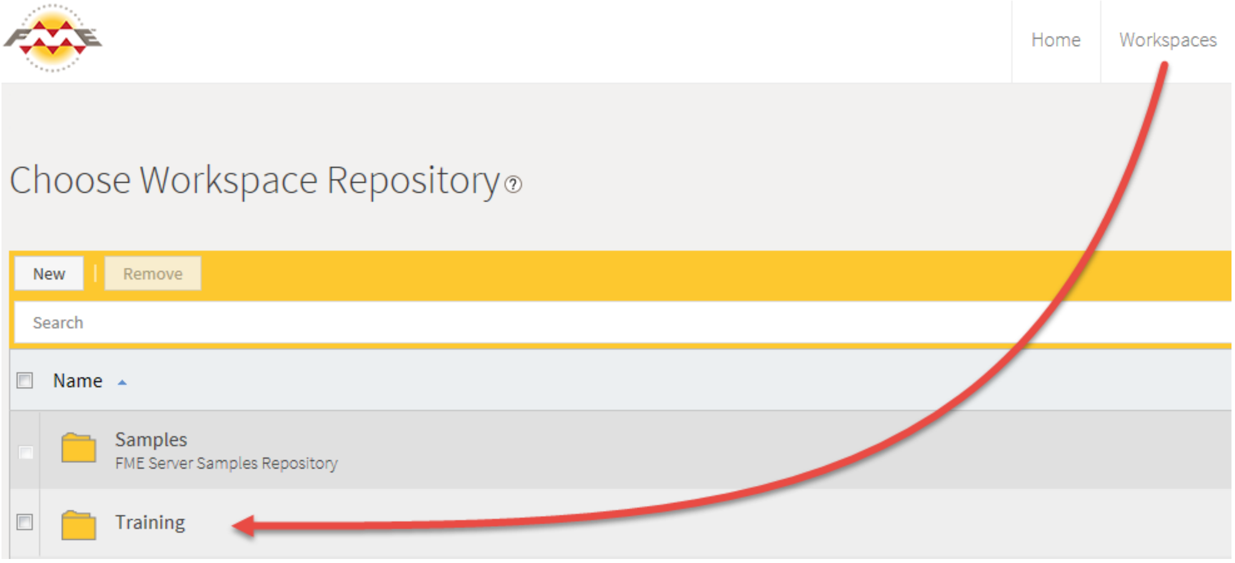
Exercise 2c: Running a Workspace	
Scenario	Airphoto Data Vendor
Data	GeoTiff Orthophotos
Overall Goal	Run the previously published workspace
Demonstrates	Running a workspace on FME Server
Starting Workspace	Exercise2a.fmw (in the Training repository on FME Server)
Finished Workspace	Exercise2a.fmw (in the Training repository on FME Server)

Now we have both a workspace and data on the server, we can run the workspace to translate the uploaded data.

1. Browse to the Workspace

Open the Web User Interface in a web browser (if it isn't already) and click **Workspaces** on the upper right-hand menu. A list of available repositories now appears.

Click on the Training repository in the list; this was the repository we created when we published the workspace. This will cause a list of available workspaces in that repository to be displayed.



Now click on your workspace, Exercise2a.fmw, and then Job Submitter. The Job Submitter parameters will open, which will allow the selection of specific source files.

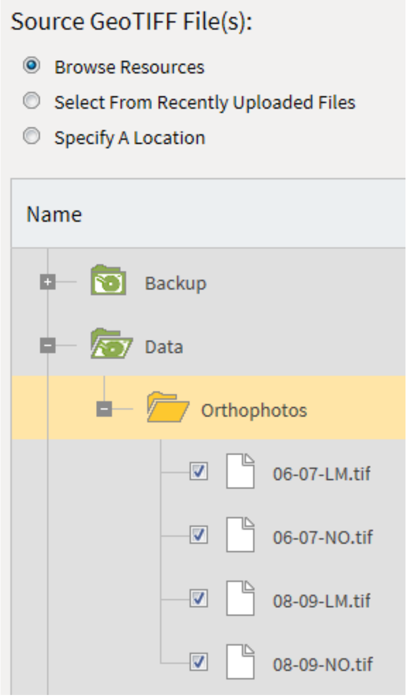
2. Select Source Data from Resources

The Configure page allows you to set any published parameters for a workspace, in the same way that using “Prompt and Run” would in FME Workbench.

Parameters which represent file sources are given special treatment to allow you to either *Browse* to a Resource dataset, *Upload* data or *Specify Location* of the data as a path on the file system.

Having already uploaded data to a Resources folder, we will browse to those files.

Select Browse Resources and expand the Data and Orthophotos folders. Select all or some of your files. The other parameters can be left with their defaults.



Source GeoTIFF File(s):

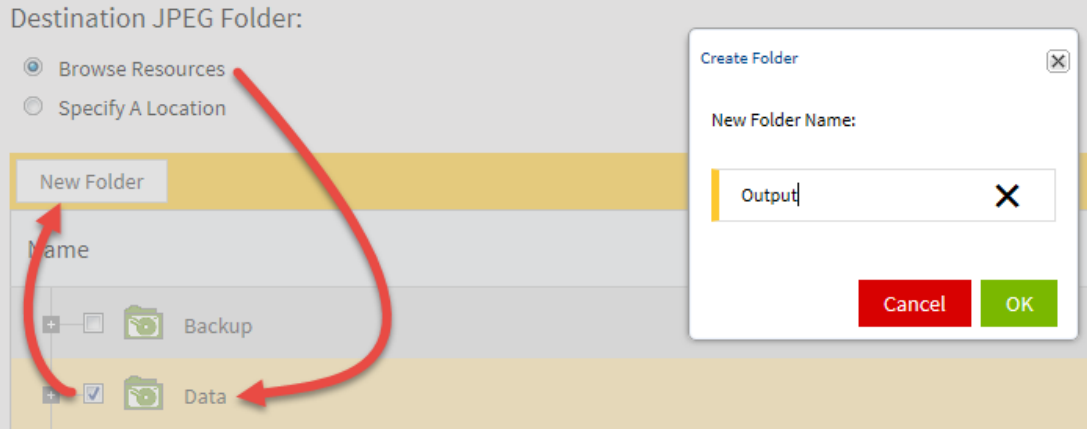
- ● Browse Resources
- ○ Select From Recently Uploaded Files
- ○ Specify A Location

Name
- Backup
- Data
  - Orthophotos
    - ☑ 06-07-LM.tif
    - ☑ 06-07-NO.tif
    - ☑ 08-09-LM.tif
    - ☑ 08-09-NO.tif

3. Select Destination Folder

Once the source data is selected, we can now select an output location in the Resources folder.

Locate the Destination Dataset parameter and click the Browse tab if it is not already highlighted. Select the Data folder, click New Folder and create a new folder called Output.



Destination JPEG Folder:

- ● Browse Resources
- ○ Specify A Location

New Folder

Name
- □ Backup
- ☑ Data

Create Folder

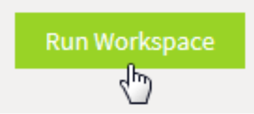
New Folder Name: Output

Ensure that the Output folder is selected as the output location.

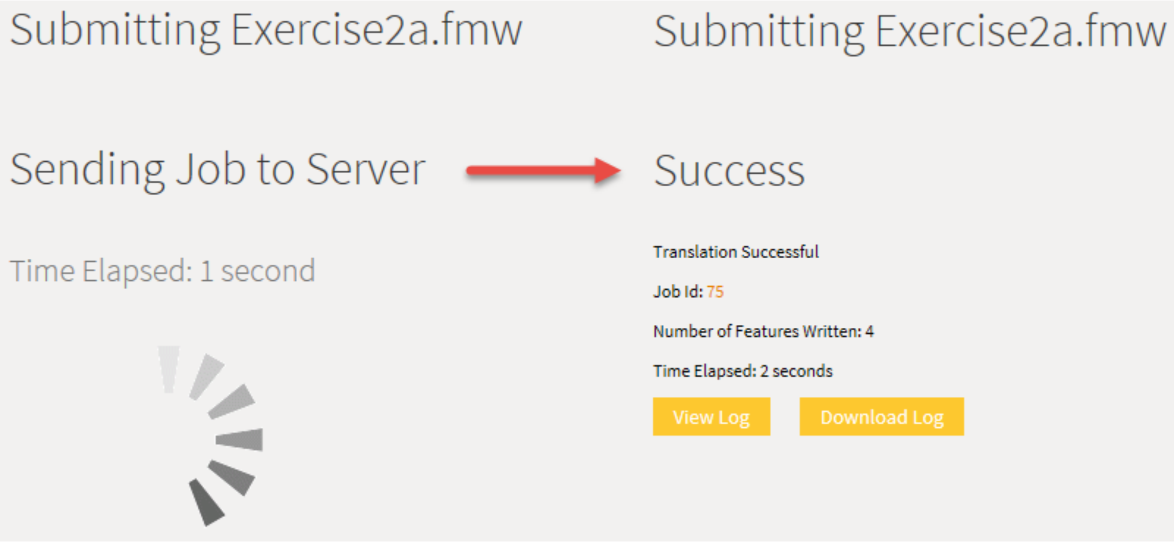
NB: *Writing to a resource folder like this is important for when there is no local or network path for FME to write the output to.*

4. Run the Workspace

Once you have selected some image files, click Run Workspace at the bottom-right of the configure page. This will submit the job with your parameters and an FME Engine will run the workspace.



The browser will wait until the job completes because it was submitted in mode known as synchronous. After a short time a Success message will appear with links to view or download the FME log file.



Translation Successful

Job Id: 75

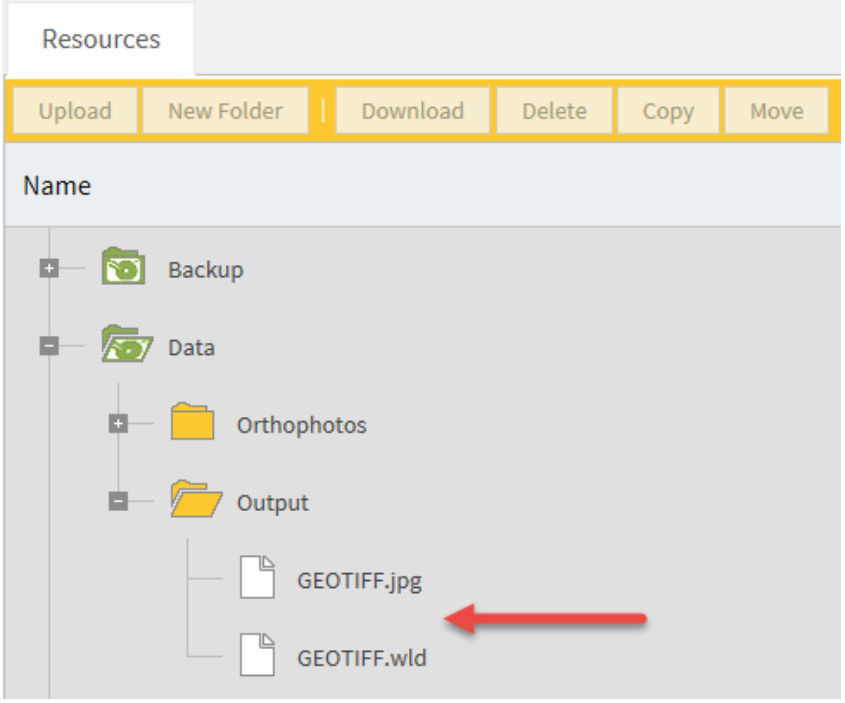
Number of Features Written: 4

Time Elapsed: 2 seconds

5. Locate Output

When complete, browse to the Resources (Manage > Resources) section of the interface and look in the Output folder. You should see the result of the translation written to this folder.

Congratulations!
You have now run a workspace with the Job Submitter service on FME Server.



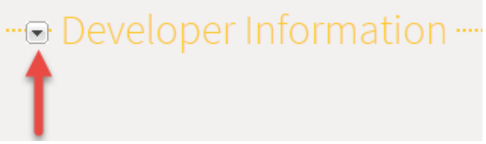
Advanced Task

You can submit a job to the FME Server Job Submitter Service or any other service using a URL or by posting a form. The syntax is provided in the Configure page for your workspace, so let's give it a try.

1. Browse to the Workspace

Browse to your workspace again in the Web User Interface using Workspaces > Training > Exercise2a.fmw > Job Submitter.

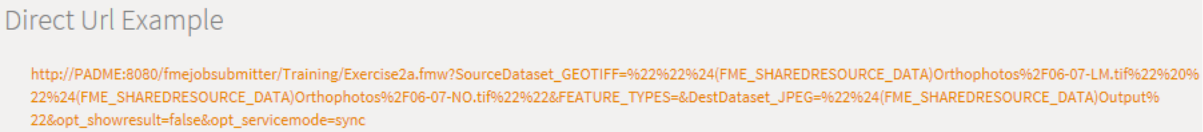
At the bottom-left corner of the page, expand Developer Information



2. Run the Workspace

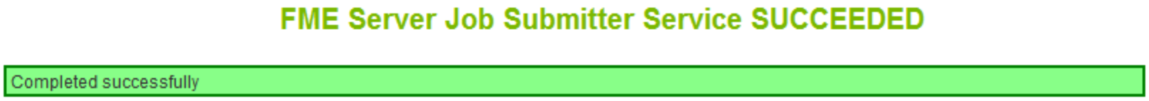
The Developer Info button opens up a section that shows how to run the workspace using a Direct URL (as a HTTP GET) and a web form (using HTTP POST).

Copy the Direct URL and paste it into the address bar of a new tab in your browser. Press enter to activate the URL and run the workspace.


Direct Url Example

```
http://PADME:8080/fmejobsubmitter/Training/Exercise2a.fmw?SourceDataset_GEOTIFF=%22%22%24(FME_SHAREDRESOURCE_DATA)Orthophotos%2F06-07-LM.tif%22%20%
22%24(FME_SHAREDRESOURCE_DATA)Orthophotos%2F06-07-NO.tif%22%22&FEATURE_TYPES=&DestDataset_JPEG=%22%24(FME_SHAREDRESOURCE_DATA)Output%
22&opt_showresult=false&opt_servicemode=sync
```

You may be prompted to authenticate again in which case enter the same user and password as you did in Exercise 2a. The job will be submitted synchronously again although the results page will look different since it was not submitted via the web interface.


FME Server Job Submitter Service SUCCEEDED

Completed successfully

3. Change Parameters

Examine how the published parameters are set in the URL and try to change one of the source GeoTiff file names (to different ones in your Resources) before running the workspace again.

Translation Results

Translation Results

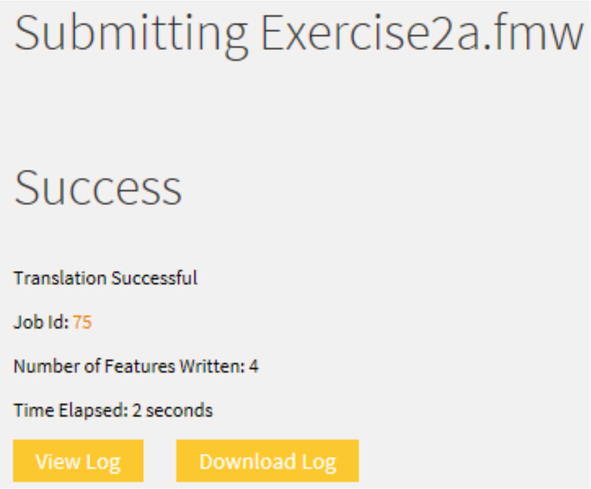
Translation Result
Job Management
Log File
Exercise 2d: Checking the Results

It's easy to check whether a workspace ran correctly or not

Translation Result

When the workspace has been run through the FME Server web user interface, you'll get a result that you can see immediately:

Here, for example, a workspace has been run and completed successfully.



Submitting Exercise2a.fmw

Success

Translation Successful

Job Id: 75

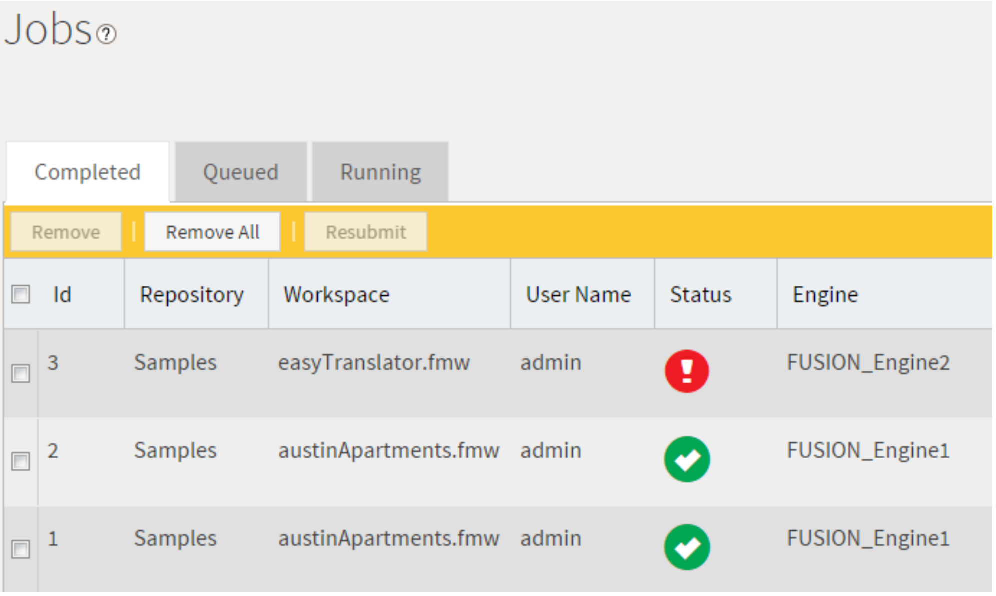
Number of Features Written: 4

Time Elapsed: 2 seconds

Job Management

When you come back to check on a previous translation you can check the result by clicking on the Jobs item on the menu. There are tabs for Queued, Running, and Completed translations.

Here you can see that two workspaces have completed successfully, and one failed:

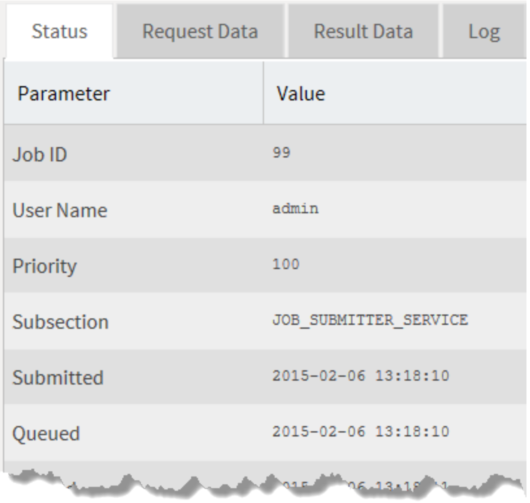


Jobs

☐ Id	Repository	Workspace	User Name	Status	Engine
☐ 3	Samples	easyTranslator.fmw	admin	!	FUSION_Engine2
☐ 2	Samples	austinApartments.fmw	admin	✓	FUSION_Engine1
☐ 1	Samples	austinApartments.fmw	admin	✓	FUSION_Engine1

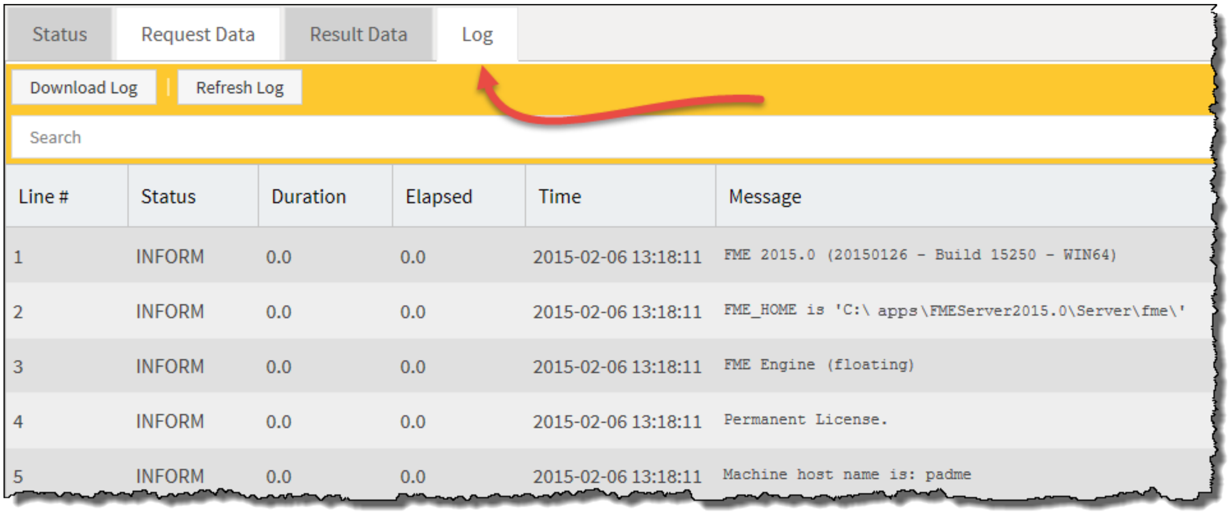
Log File

When you click on a job in the job queue it opens a dialog with more detailed information about the results of the job:



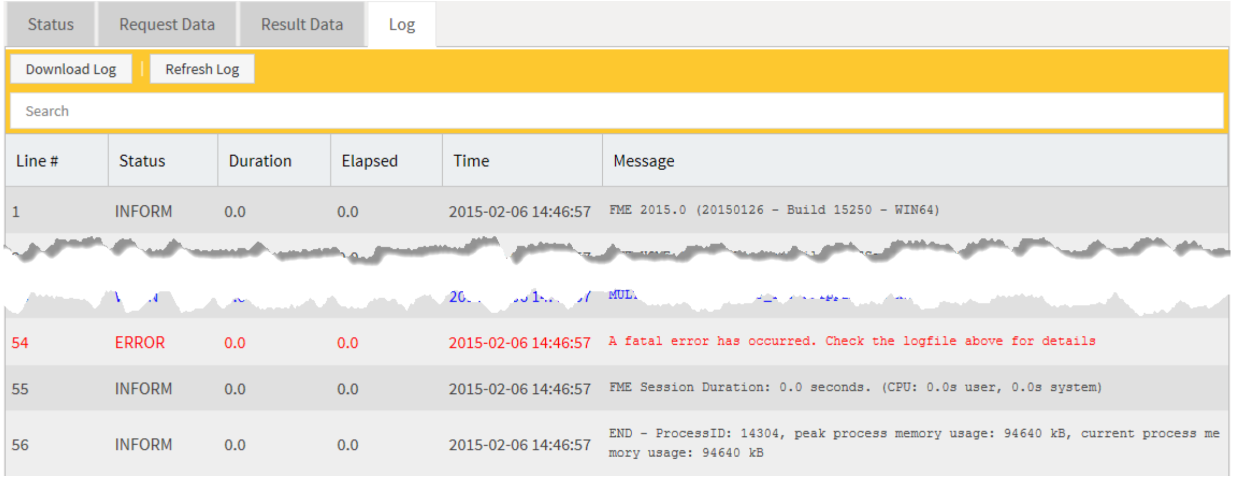
Parameter	Value
Job ID	99
User Name	admin
Priority	100
Subsection	JOB_SUBMITTER_SERVICE
Submitted	2015-02-06 13:18:10
Queued	2015-02-06 13:18:10

You can then click the Log tab to check the log file and find even more information about the translation itself. This log is equivalent to the log shown in the FME Workbench log window:



Line #	Status	Duration	Elapsed	Time	Message
1	INFORM	0.0	0.0	2015-02-06 13:18:11	FME 2015.0 (20150126 - Build 15250 - WIN64)
2	INFORM	0.0	0.0	2015-02-06 13:18:11	FME_HOME is 'C:\ apps\FMEServer2015.0\Server\fme\'
3	INFORM	0.0	0.0	2015-02-06 13:18:11	FME Engine (floating)
4	INFORM	0.0	0.0	2015-02-06 13:18:11	Permanent License.
5	INFORM	0.0	0.0	2015-02-06 13:18:11	Machine host name is: padme

The log is obviously of most use where the translation has been flagged as failed:



Line #	Status	Duration	Elapsed	Time	Message
1	INFORM	0.0	0.0	2015-02-06 14:46:57	FME 2015.0 (20150126 - Build 15250 - WIN64)
54	ERROR	0.0	0.0	2015-02-06 14:46:57	A fatal error has occurred. Check the logfile above for details
55	INFORM	0.0	0.0	2015-02-06 14:46:57	FME Session Duration: 0.0 seconds. (CPU: 0.0s user, 0.0s system)
56	INFORM	0.0	0.0	2015-02-06 14:46:57	END - ProcessID: 14304, peak process memory usage: 94640 kB, current process memory usage: 94640 kB

There is also a button labelled **Download Log** where you can download the file if necessary.

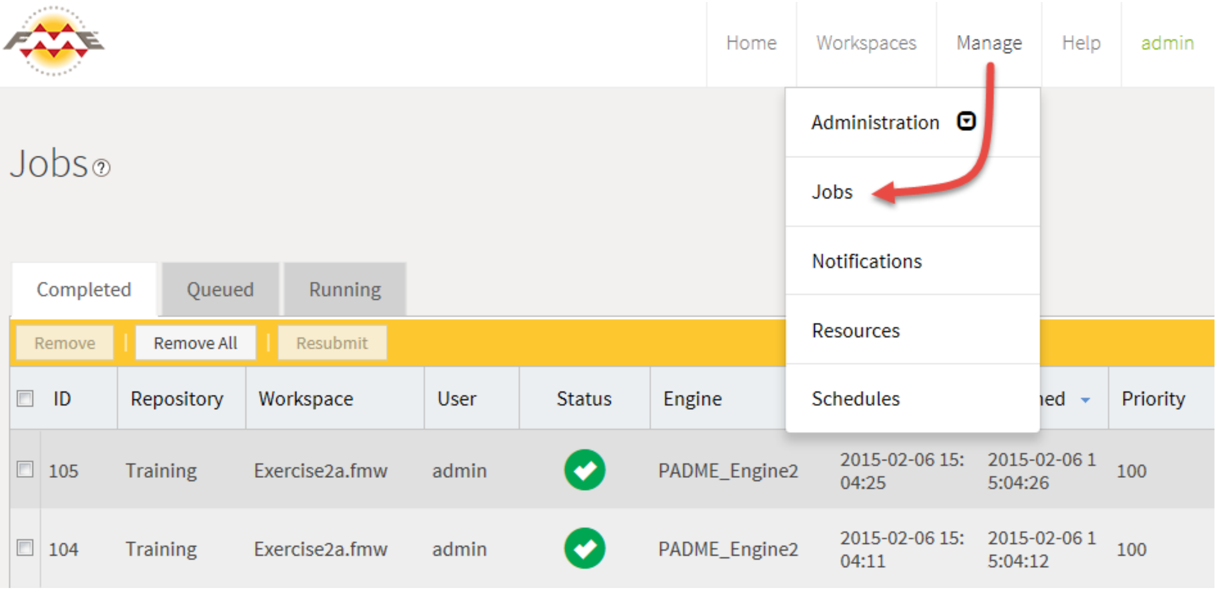
Exercise 2d: Checking the Result	
Scenario	Airphoto Data Vendor
Data	GeoTiff Orthophotos
Overall Goal	Checking the previous translation
Demonstrates	FME Server translation results
Starting Workspace	n/a
Finished Workspace	n/a

Having run a workspace, now let's look at the job history details. This will confirm the workspace ran to completion and produced the correct results.

1) Examine the Job History
Click on Manage then Jobs in the Web User Interface.

This will open a page with three tabs: Completed, Queued, and Running.

There should not be any queued or running jobs so click on the Completed tab to see a view of the latest completed jobs. Each job item represents one run of one workspace and includes details such a Status, Workspace Name etc.



ID	Repository	Workspace	User	Status	Engine		ed	Priority
105	Training	Exercise2a.fmw	admin	✓	PADME_Engine2	2015-02-06 15: 04:25	2015-02-06 1 5:04:26	100
104	Training	Exercise2a.fmw	admin	✓	PADME_Engine2	2015-02-06 15: 04:11	2015-02-06 1 5:04:12	100

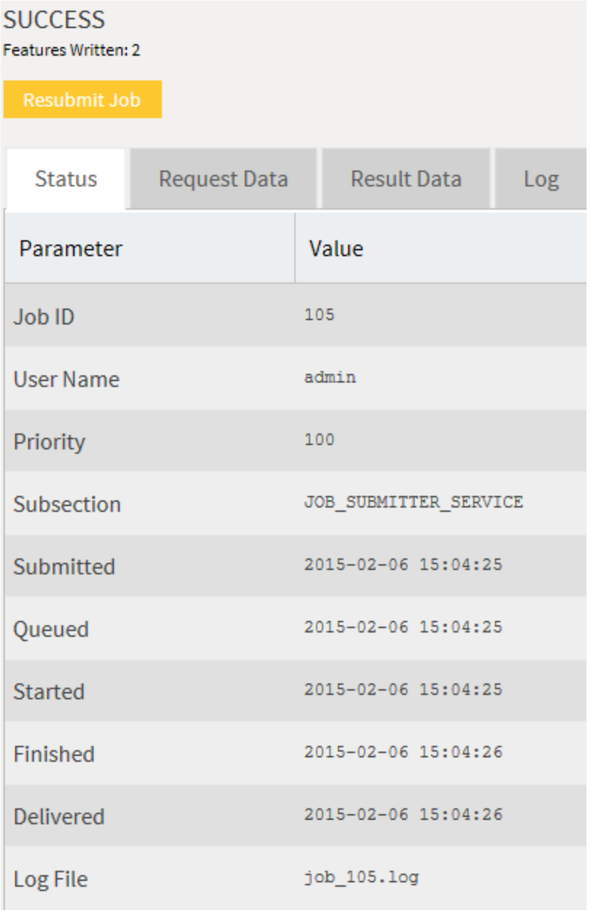
2. Examine Last Job
Click on the latest job which will be at the top of the list and should be the workspace you ran using the Job Submitter Service in Exercise 2c.

Information is available under one of a number of tabs: Status, Request Data, Result Data, Log.

Click on each tab in turn to inspect the information.

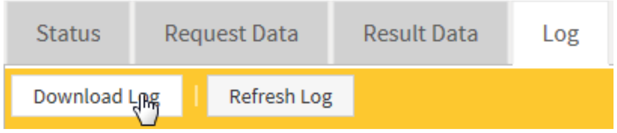
The Request Data tab shows the values used for each published parameter when the workspace was run.

Look at the values for the parameters DestDataset_JPEG and SourceDataset_GEOTIFF, both of which should point to the resource directory represented by an environment variable such as `$(FME_SHAREDRESOURCE_DATE)`



SUCCESS
Features Written: 2

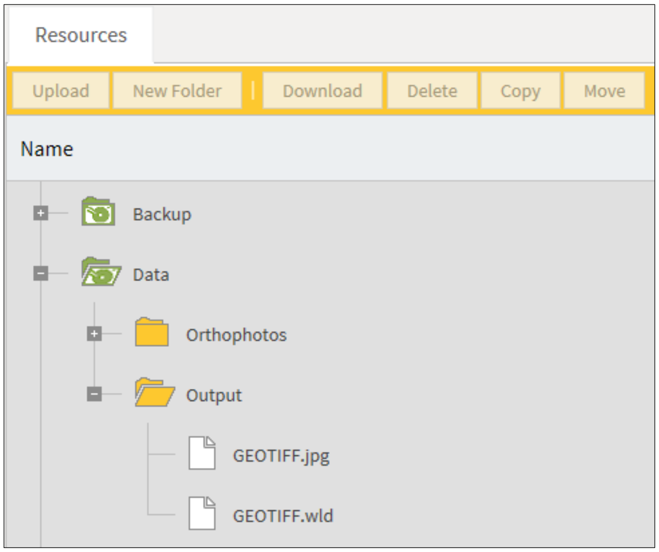
Parameter	Value
Job ID	105
User Name	admin
Priority	100
Subsection	JOB_SUBMITTER_SERVICE
Submitted	2015-02-06 15:04:25
Queued	2015-02-06 15:04:25
Started	2015-02-06 15:04:25
Finished	2015-02-06 15:04:26
Delivered	2015-02-06 15:04:26
Log File	job_105.log



3. Examine the FME Log File
Under the Log tab, click the **Download Log** button. This will download the FME log to a file.

Examine the log to see how it compares to a log file generated by FME Workbench. It should look very similar. The main differences are the Command line arguments, which include parameters specific to FME Server.

4. Examine Output Folder
Under the Resources menu browse to the Output folder and confirm the data has been written.



Troubleshooting for Administrators

Troubleshooting for Administrators

Workbench-Server Connection
Workspaces are Queued but not Run
Workspaces Fail when Run

FME Server Basic Training 2015

This section shows a few basic troubleshooting techniques in case of emergency

FME Workbench-FME Server Connection

If you are unable to connect from FME Workbench to FME Server then the following suggestions may be of help.

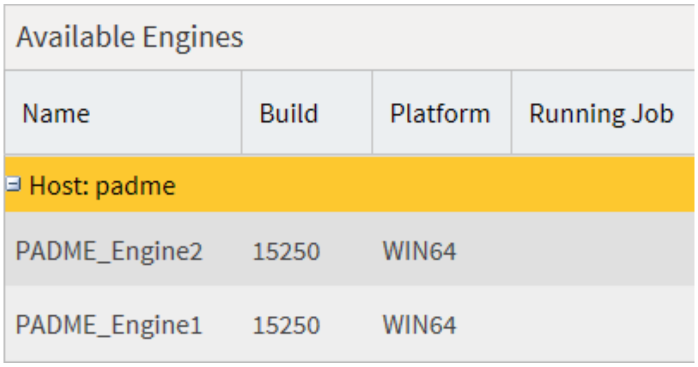
- Check if there is a firewall running on either your computer or the FME Server. If so, you must open port 7071 to use Direct Connection or port 80 (or 8080) to use the Web Connection.

- Restart the FME Server Services. On Windows, go to Start > Programs > FME Server > Windows Service > right-click Restart and select Run as administrator.

Workspaces are Queued but not Run

If a workspace appears in the FME Server queue, but is never executed, then it may be because no engines are running.

- Check the Web User Interface (Manage > Engine Management) to confirm engine status. If no engines are available check licensing and update as necessary. Restart FME Server once a proper licence is in place.



Available Engines

Name	Build	Platform	Running Job
⊟ Host: padme			
PADME_Engine2	15250	WIN64	
PADME_Engine1	15250	WIN64	

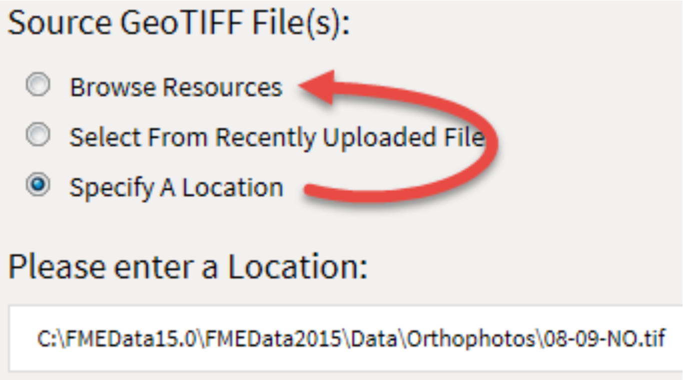
Workspaces Fail when Run
If a workspace fails when it is being run, then the following suggestions may help

- Run the workspace first on FME Desktop. If it does not work there, it will not work on FME Server.

- Check the FME log file using Manage > Jobs > Completed in the Web User Interface. This may help to explain why there is a problem.

- Commonly data paths cause problems when moving from a local Desktop machine to a Server environment. Check the dataset parameters (Reader and Writer) to ensure they are not referring to a local path that does not exist on the Server.

  You may need to change the parameter to use the Resources dialog (Browse) and not a file path dialog (Specify Location).



Source GeoTIFF File(s):

○ Browse Resources
○ Select From Recently Uploaded File
● Specify A Location

Please enter a Location:

C:\FMEData15.0\FMEData2015\Data\Orthophotos\08-09-NO.tif

- If you are trying to read data with a UNC path, ensure that the FME Server Windows Service is being run by a user with the proper domain access permissions.

Module Review

Module Review

What you should have learned from this module

FME Server Basic Training 2015

This module introduced you to running workspaces on FME Server.

What You Should Have Learned from this Module

The following are key points to be learned from this module.

Theory

- FME Server is a Model Driven Architecture – where workspaces are the models
- FME Desktop (particularly FME Workbench) is the authoring tool for FME Server workspaces
- FME workspaces are published from Desktop to Server and stored on FME Server in repositories
- Data (and other related files) can be uploaded and stored on FME Server as “Resources”. The output from a translation can be written to a Resources folder too.
- The Job Submitter service is for running a workspace “as-is”

FME Skills

- Publish a workspace from FME Workbench to FME Server, creating a new repository if necessary
- Upload source data to a Resources folder on FME Server
- Run a workspace on FME Server – either through the web interface or via a URL – that makes use of datasets in a Resource folder
- Examine the results of the translation to see if it was successful

Self-Serve with FME Server

Self-Serve with FME Server

Self Serve with FME Server

What is Self-Serve?
Self-Serve Types
Self-Serve and Services
Self-Serve and Parameters

FME Server Basic Training 2015

Self-Serve takes the tedious tasks away from staff and leaves them to concentrate on more important things.

What is Self-Serve?

Self-Serve is the term for a system set up to allow the end user to carry out his own data translations and transformations. In this way, routine data management tasks are offloaded from staff to the user, who is empowered to carry out processes at their own convenience.

Usually the system is set up in such a way that the end user needs no prior FME experience or training to carry out their goals; for example, they can access the functionality through a web interface customized to their particular needs. In fact the user does not even need to know of FME, or that FME is the engine driving their applications!

Self-Serve Types

In general there are two types of self-serve systems.

Data Upload systems are where the user is able to upload their data to be processed on FME Server. A typical application would be a user uploading data to be validated. The data is run against a number of tests in an FME workspace and the results sent back to the user.

Data Download systems are where the user is able to serve themselves with data. A typical application would be an organization that frequently provides data to either staff or customers. With a Data Download system the user can fetch their own data, rather than having to be provided with it in a more manual way.

Self-Serve and Services

Self-Serve is implemented through a number of Services on FME Server. A Service is a particular method of communication between client and server. Different services allow different forms of data distribution to take place.

Self-Serve and Parameters

FME translations and transformations are controlled by Parameters. To enable users to select the requirements for their translation, FME includes functionality called Published Parameters. A parameter that is published becomes available for the end-user to set.

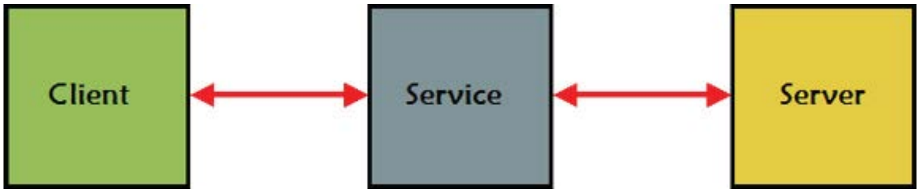
FME Server Services

FME Server Services

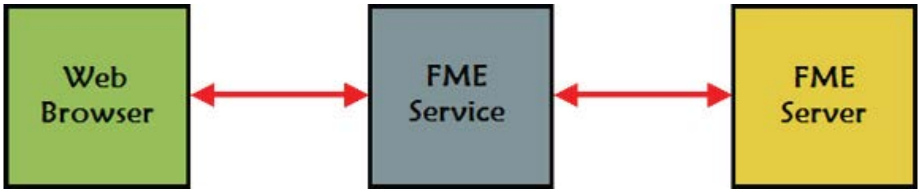
What is a Service?
Available Services
Workspaces and Services

Many standard operations using FME Server take place using something called a 'service'. FME Server provides a wide range of services to carry out common tasks.

What is a Service?
In the simplest of terms, a service is a piece of software that handles communications between a client and a server. In other words, it's a tool that allows users to access complex functionality through a simplified interface.



In terms of FME Server, the client is often—but not always—a web browser that passes requests to FME Server using a service.



For FME Server, a service lets you send specific types of requests to FME Server and provide results to client applications in a specific way.

For example, instead of just running a workspace, you can have a web page ask for the results of the workspace as a package of data compressed in a zip file.

Professor Spatial F.M.E., E.T.L. says...
'Although the concept sounds complicated, a service is a simpler way of communicating requests to FME Server than using the API.

Also, FME Server includes a number of predefined services that cover a lot of the functionality you are likely to need.'

Available Services

FME Server includes the following services:

OGC Web Feature Service (WFS)	OGC Services
OGC Web Mapping Service (WMS)	
Data Download Service	Transformation Services
Data Streaming Service	
Job Submitter Service	
KML Network Link Service	
Data Upload Service	Utility Services
Token Service	
Web Connection (SOAP) Service	
REST Service	
Notification Service	Notification Services

Remember that services can communicate in both directions. Transformation services – for example Data Download – are primarily Self-Serve tools for Server to deliver data to the end user.

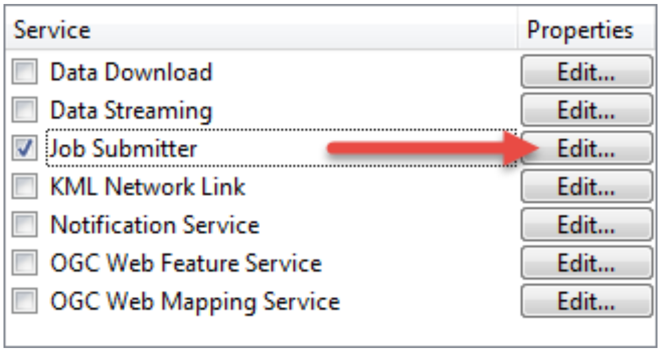
Utility services can be described as “helper” services. They interact with FME Server to assist in menial tasks such as uploading data or providing token security. In most cases these are facilities that an author or developer will be using in a way that’s hidden from the user.

The Notification Service is used for passing short messages into and out of FME Server. Incoming messages notify FME Server to take some action, whereas outgoing messages alert an end-user (or system) that some sort of event has occurred.

Workspaces and Services

When a workspace is published to FME Server it can be registered with a particular service.

The Job Submitter service is automatically selected in the FME Server publishing wizard, whenever a workspace is published, but many other services are available too.



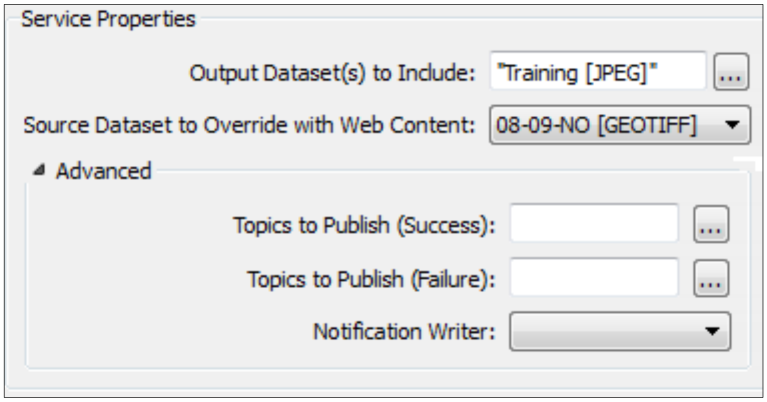
Service	Properties
□ Data Download	Edit...
□ Data Streaming	Edit...
☑ Job Submitter	Edit...
□ KML Network Link	Edit...
□ Notification Service	Edit...
□ OGC Web Feature Service	Edit...
□ OGC Web Mapping Service	Edit...

Registering a workspace with a service makes the workspace available for use in that service although, as you'll discover, not every workspace is capable of being used by every service.

Be aware of the Edit button to the right of each service.

Every service has a set of parameters available that determine how a workspace will be run with that service.

These parameters include ones for notification topics to trigger on completion of the workspace.



Service Properties

Output Dataset(s) to Include: "Training [JPEG]" ...

Source Dataset to Override with Web Content: 08-09-NO [GEOTIFF]

Advanced

Topics to Publish (Success): ...

Topics to Publish (Failure): ...

Notification Writer:

It's important to understand that a workspace may be registered against one service, many services, or no services at all.

When a workspace is not registered against a service, you can run it as a regular translation using the FME Server Console or the *FMEServerJobSubmitter* transformer in Workbench.

When a workspace is registered against a service, it can be used by that service and it's still available for use as a regular translation using FME Server Console.

Self-Serve and Parameters

Self-Serve and Parameters

What are Parameters?
Published Parameters
Parameter Uses in Self-Serve

FME Server Basic Training 2015

Workspace parameters are the key to controlling Self-Serve and the Data Download service

What are Parameters?

Parameters are what control a translation.

Chef Bimm says…

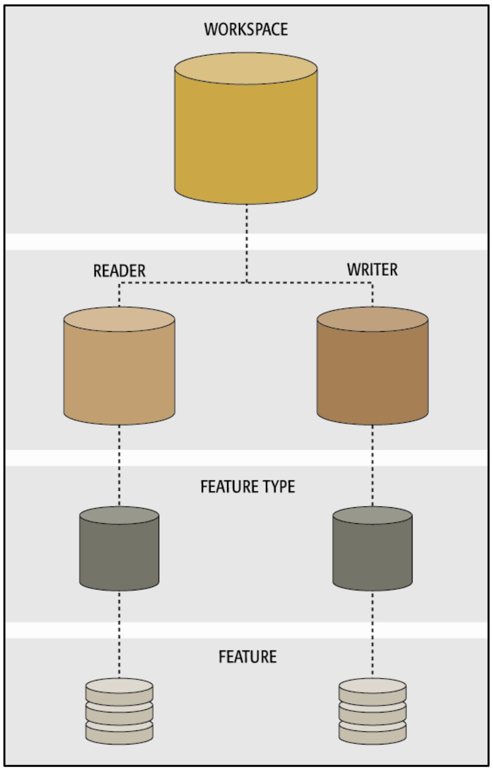
"Parameters in a translation are like the options when making coffee.

The machine can be set with the ingredients to be used, how they are combined, and what the end result of the process will be like."

In the hierarchy of different translation components, each different level of the hierarchy has a set of parameters that belong to it.

So there are:

- Workspace Parameters
- Reader Parameters
- Writer Parameters
- Feature Type Parameters



Published Parameters

Published Parameters are those that have been exposed to the end-user, so they can set the values themselves. They are also called "User Parameters".

In a self-serve application, published parameters are important to let the end user control how the data is served in terms of style and structure.

Chef Bimm says…

"Published Parameters are like when you are given control of the machine to make your own coffee."

Parameter Uses in FME Server

For self-serve systems, parameters can be used to set:

- Coordinate System Selection
- Feature Types (Layers) to Read
- Geographic Area (Bounding Box or Area of Interest)
- Any other Reader, Writer, or Transformer parameters

With FME Server, the key to successful workspace authoring is flexibility. Workspaces need to be flexible to allow end-users to make choices without seeing all of the complexity of the workspace or the data behind it. Parameters are one way to accomplish this.

Data Uploads

Data Uploads

What is Data Upload?
Implementing Data Upload
The Data Upload Service
Resource Folders
Exercise 3a: Data Upload

FME Server Basic Training 2015

Data Upload is a good way for a user to provide data to be processed

What is Data Upload?

As the name suggests, Data Upload allows a user to deliver data to FME Server for processing.

Data Upload can be used from any FME Server client, including:

- The Web User Interface
- An FME workspace
- A custom web application

Data Upload is often used for submitting data to an organization, for example a property developer submits a planning application containing a DWG dataset to a municipal planning department.

It is also often used for publishing data to be processed on FME Server, for example where FME Server is set up as a data validation web service and an end-user is uploading a set of data to be checked.

It's worth noting that data upload also includes other resources that may be required for a translation to run; for example custom transformers or text-file lookup tables may also be uploaded.

Implementing Data Upload

In FME, there are two forms of data upload: the **Data Upload Service** and the **Resource Folders**.

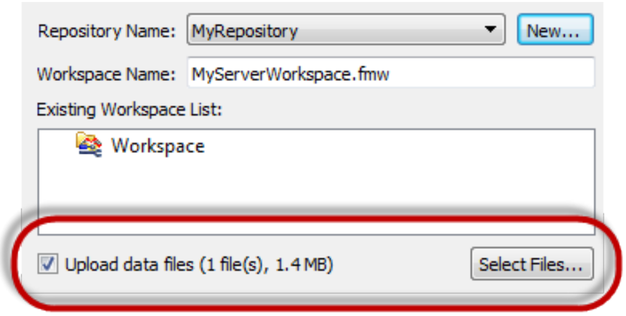
The Data Upload Service allows data to be uploaded to a particular workspace in a particular repository. The data is held temporarily for the workspace to be run.

The Resource Folders allow data to be uploaded to a folder for use by any workspace in any repository. This upload is persistent and the data held there for as long as it required.

The Data Upload Service

The Data Upload Service allows data and other resource files to be uploaded to FME Server.

Files can be uploaded through the FME Server Publishing Wizard, when a workspace is published from Workbench to FME Server:


Repository Name: MyRepository

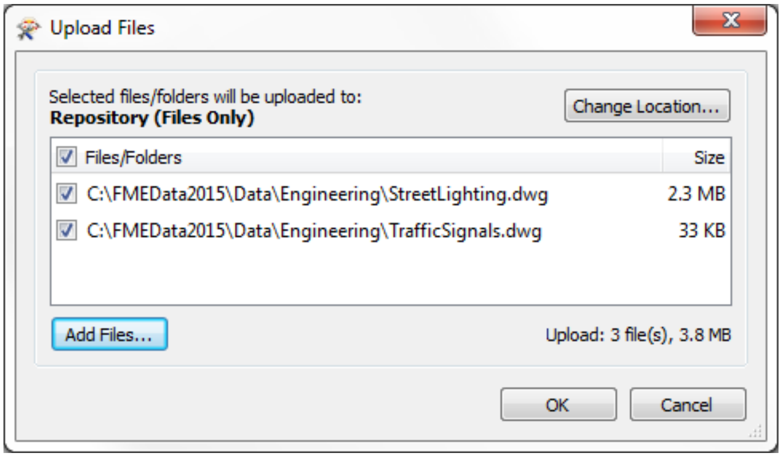
Workspace Name: MyServerWorkspace.fmw

Existing Workspace List:

Workspace

☑ Upload data files (1 file(s), 1.4 MB)

Clicking the Select Files button opens a dialog that shows what data is being uploaded.



Selected files/folders will be uploaded to:
Repository (Files Only)

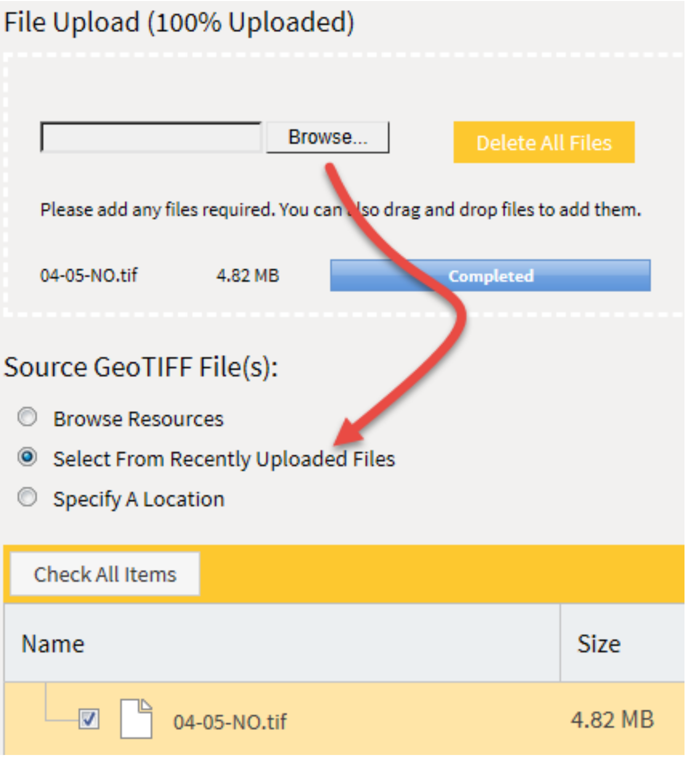
☑ Files/Folders	Size
☑ C:\FMEData2015\Data\Engineering\StreetLighting.dwg	2.3 MB
☑ C:\FMEData2015\Data\Engineering\TrafficSignals.dwg	33 KB

Upload: 3 file(s), 3.8 MB

The upload of these files can be turned on and off as required, and you can also change the location to which data is uploaded – either to a repository (Data Upload) or to a Resource Folder – by clicking the Change Location button.

The other way to upload data with this service is through the configure dialog when running a workspace.

Use the File Upload section to upload the data, then choose Select From Recently Uploaded Files to select which files should be assigned to which published parameter:


File Upload (100% Uploaded)

Please add any files required. You can also drag and drop files to add them.

04-05-NO.tif 4.82 MB Completed

Source GeoTIFF File(s):

○ Browse Resources

● Select From Recently Uploaded Files

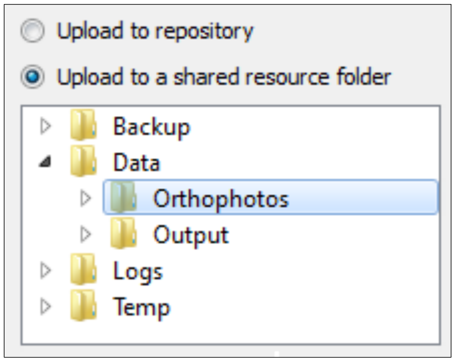
○ Specify A Location

Name	Size
☑ 04-05-NO.tif	4.82 MB

Be aware that data uploading with this service is only available with the chosen workspace. Also, the data is stored only temporarily and will be deleted in – by default – 30 minutes time.

Resources Folders

Resource Folders are a more permanent and flexible way to upload data and resources to FME Server. Similarly to the Data Upload Service, files can be uploaded at the same time as a workspace.



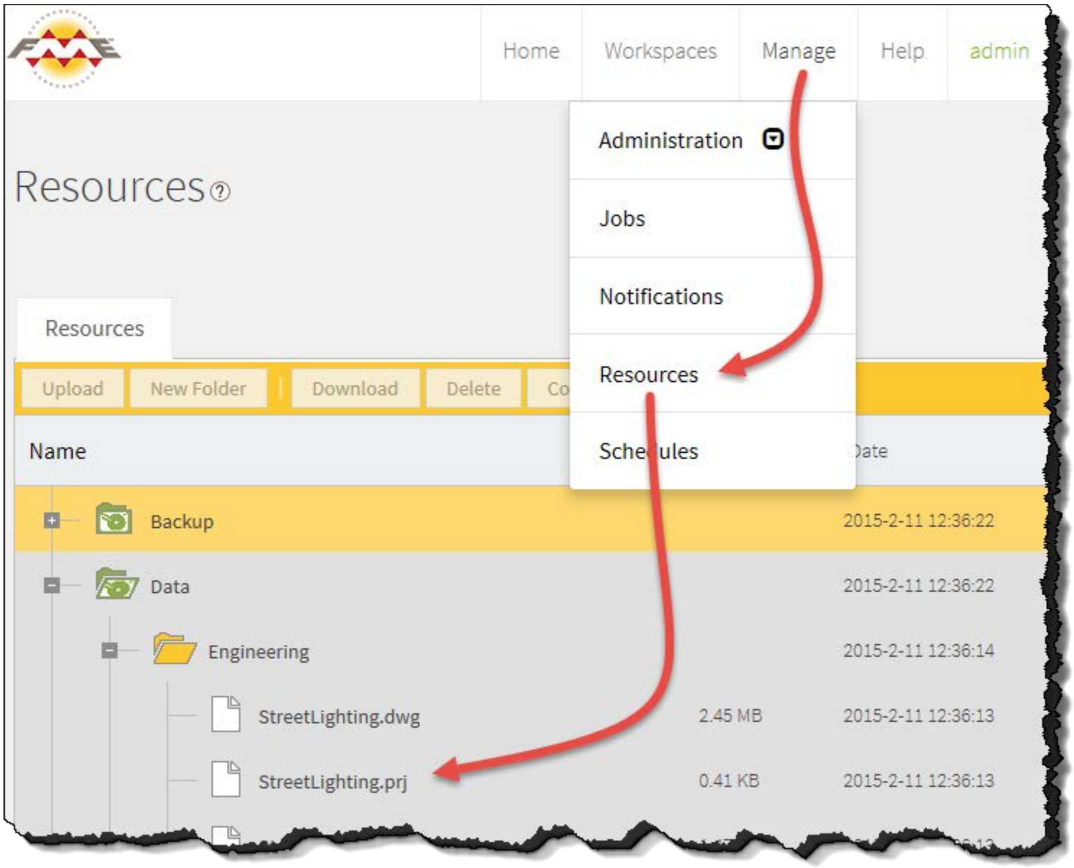
○ Upload to repository

● Upload to a shared resource folder

- Backup
- Data
  - Orthophotos
  - Output
- Logs
- Temp

When editing the resource info (in the publishing wizard, under Select Files, Change Location) you would simply choose to upload the files as a **shared resource** *(left)*. One other requirement is to select a location on the resources file system.

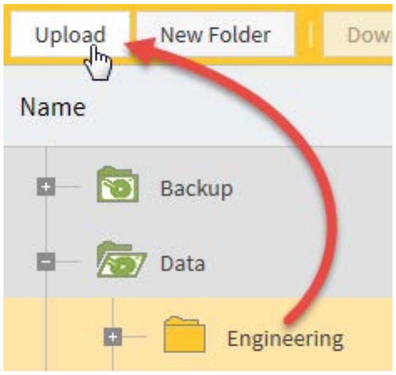
Another way to use this functionality is to use the Resources tool in the FME Server Web User Interface *(below)*. Here you can upload, download, copy, move, and delete files; plus manage folders too.



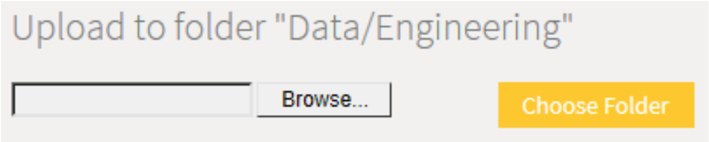
Name		Date
Backup		2015-2-11 12:36:22
Data		2015-2-11 12:36:22
Engineering		2015-2-11 12:36:14
StreetLighting.dwg	2.45 MB	2015-2-11 12:36:13
StreetLighting.prj	0.41 KB	2015-2-11 12:36:13

Resource Upload

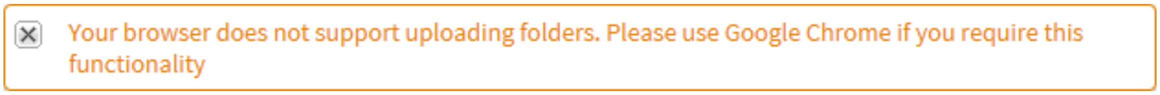
To upload data simply click on a folder and then choose the Upload option:



The upload tool gives you the option to either upload a set of files, or an entire folder of data:



To upload a folder of data requires use of a browser – for example Google Chrome – that has folder upload capabilities. At the time of writing, neither Firefox nor IE has this ability.



☒ Your browser does not support uploading folders. Please use Google Chrome if you require this functionality

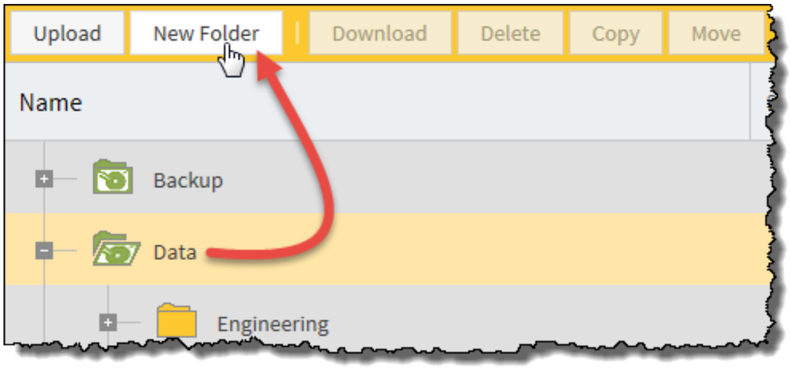
New Folder

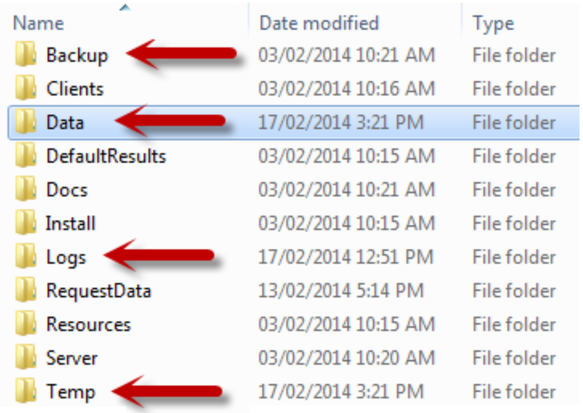
The New Folder tool creates a new folder or sub-folder in the Resources mechanism.

For example, to create a sub-folder in the Data folder simply click on Data and then click New Folder:

You are prompted to enter a name for the new folder.

Enter a name, click OK, and the folder is created.





Name	Date modified	Type
Backup	03/02/2014 10:21 AM	File folder
Clients	03/02/2014 10:16 AM	File folder
Data	17/02/2014 3:21 PM	File folder
DefaultResults	03/02/2014 10:15 AM	File folder
Docs	03/02/2014 10:21 AM	File folder
Install	03/02/2014 10:15 AM	File folder
Logs	17/02/2014 12:51 PM	File folder
RequestData	13/02/2014 5:14 PM	File folder
Resources	03/02/2014 10:15 AM	File folder
Server	03/02/2014 10:20 AM	File folder
Temp	17/02/2014 3:21 PM	File folder

Besides those methods, folders in the Resources mechanism can be accessed – or created – in the server's file system.

For example, here are folders in C:\apps\FMEServer on a Windows file system.

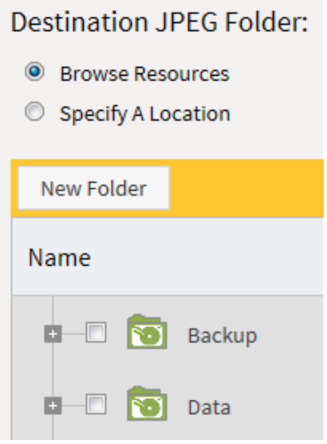
Benefits
Resource Folders are more permanent because the data is not automatically deleted after a set time. Instead all uploaded files reside there until removed by a user/administrator.

Additionally, Resource Folders are more flexible for a number of reasons.

Firstly, you can access the files and manage them through a file system on the server. This is useful in itself, but it also allows the data to be accessed through UNC paths, which the Data Upload service does not.

Secondly, Resource Folders can be used for more than data upload.

It’s also possible to write the results of a translation to a resource folder, as in this example. You would just set the output dataset parameter to a resource folder.



Destination JPEG Folder:

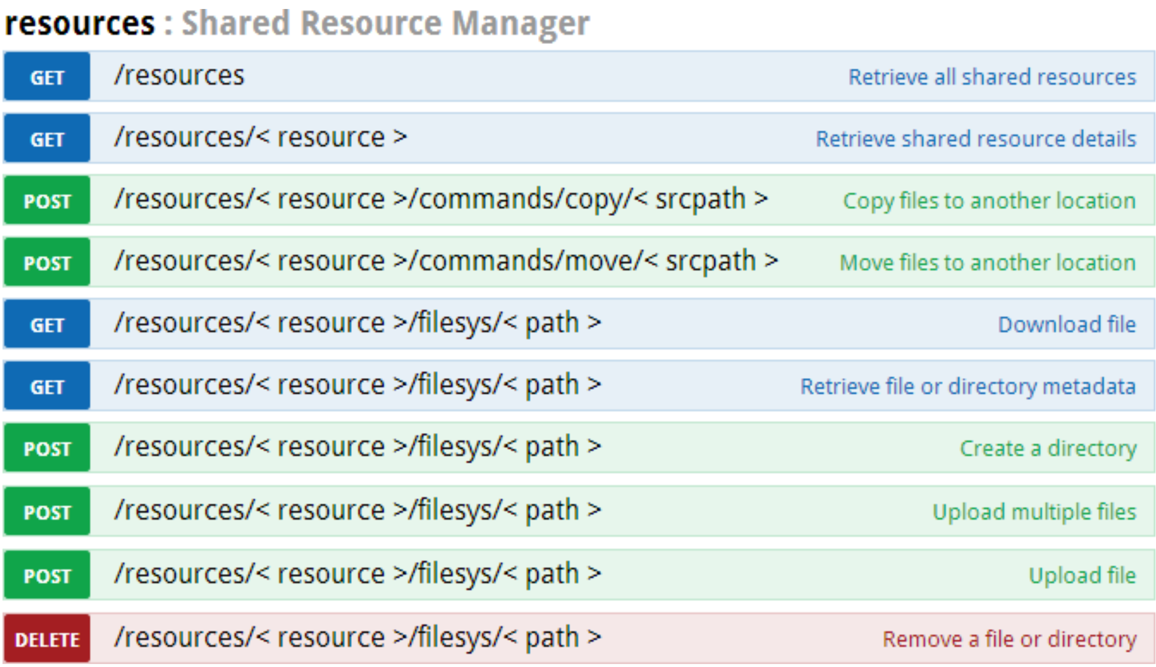
● Browse Resources
○ Specify A Location

New Folder

Name

□ Backup
□ Data

Also, files inside a Resource Folder can be managed using the REST API too, with POST and GET functions for downloading files, creating folders, and copying files (for example). This gives control over these resources to any developer who wishes to create their own web applications.



resources : Shared Resource Manager

GET	/resources	Retrieve all shared resources
GET	/resources/< resource >	Retrieve shared resource details
POST	/resources/< resource >/commands/copy/< srcpath >	Copy files to another location
POST	/resources/< resource >/commands/move/< srcpath >	Move files to another location
GET	/resources/< resource >/filesys/< path >	Download file
GET	/resources/< resource >/filesys/< path >	Retrieve file or directory metadata
POST	/resources/< resource >/filesys/< path >	Create a directory
POST	/resources/< resource >/filesys/< path >	Upload multiple files
POST	/resources/< resource >/filesys/< path >	Upload file
DELETE	/resources/< resource >/filesys/< path >	Remove a file or directory

Exercise 3a: Data Upload	
Scenario	Airphoto Data Vendor
Data	GeoTiff Orthophotos
Overall Goal	Create a data upload service for validating spatial data
Demonstrates	Authoring for and Using the Data Upload Service
Starting Workspace	n/a
Finished Workspace	C:\FMEData2015\Workspaces\ServerAuthoring\Exercise3a-Complete - DataUpload.fmw

This time you wish to create a solution for data providers to automatically upload data to your system. Naturally you will wish to test data quality – in this case resolution – and reject any data that does not meet your standards.

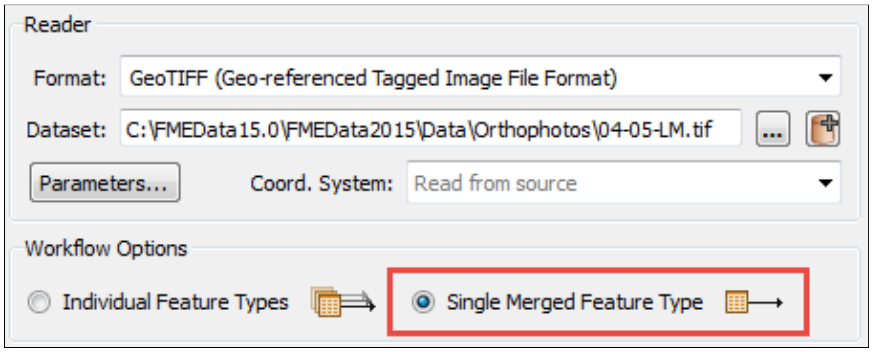
1. Start Workbench

Start Workbench (if necessary).

2. Add Reader

The first component required for Data Upload is a Reader, so select Readers > Add Reader from the menubar. When prompted add a Reader with the following parameters:

Reader Format	GeoTIFF (Geo-referenced Tagged Image File Format)
Reader Dataset	Any TIF file in C:\FMEData2015\Data\Orthophotos\
Workflow Options	Single Merged Feature Type



Reader

Format: GeoTIFF (Geo-referenced Tagged Image File Format)

Dataset: C:\FMEData15.0\FMEData2015\Data\Orthophotos\04-05-LM.tif

Parameters...

Coord. System: Read from source

Workflow Options

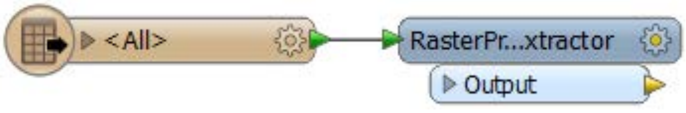
○ Individual Feature Types ● Single Merged Feature Type

The workflow option is important because setting a merged feature type allows any TIFF file to enter the workspace. If it is not set then only the original file could ever be read.

3. Add RasterPropertyExtractor

To test the quality of the incoming data we need to first extract some information about it. So, add a RasterPropertyExtractor transformer. You can do this by clicking on a clear part of the canvas and typing *RasterPropertyExtractor*. Drag a connection from the Reader to the transformer.

This will extract information about the incoming data and store it as attributes.

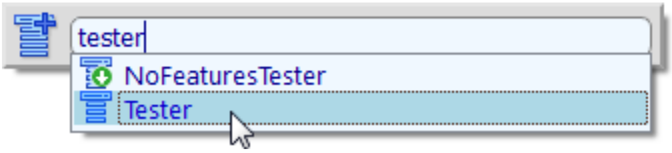


If you wish, you may connect a Logger transformer and run the workspace, just to see what the attributes look like.

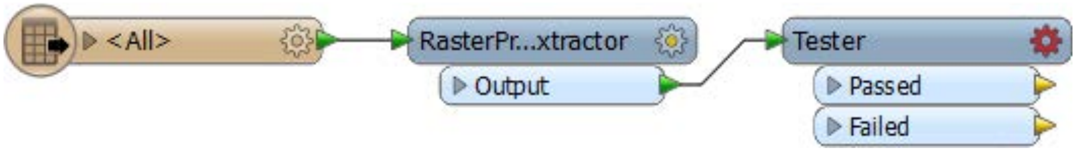
4. Add Tester

Now we will add a Tester transformer to test whether the properties of the data are a match for what we require.

Do this by clicking on the RasterPropertyExtractor transformer and typing *Tester*. Then select Tester from the drop-down list of transformers.



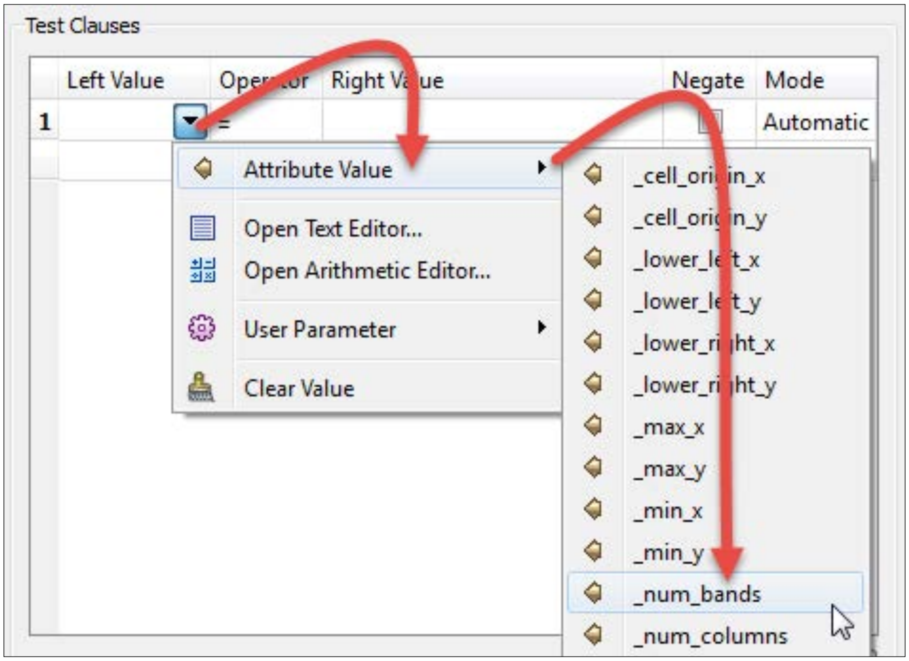
The workspace will now look like this:



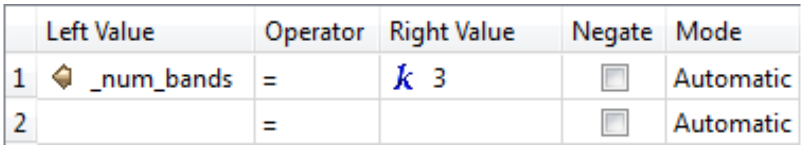
5. Set Tester Parameters

Click on the red cogwheel icon on the Tester to open its parameters dialog.

In the dialog double-click in the Left Value field. From the drop-down list select Attribute Value > _num_bands. This attribute tells us how many bands are in the source data:

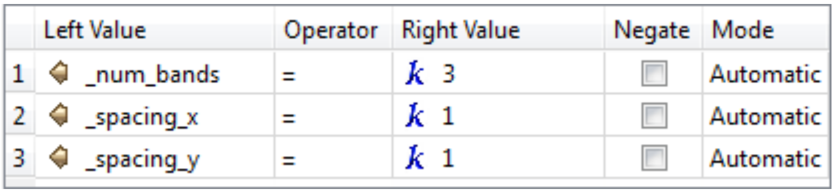


Set the Operator field to equals (=) and type the number 3 into the Right Value field:



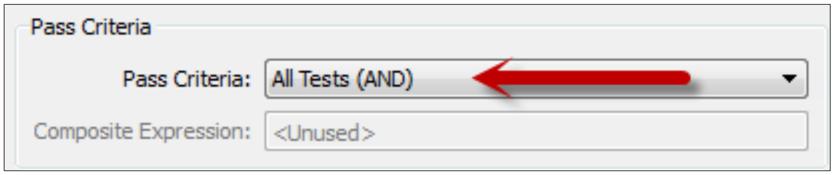
	Left Value	Operator	Right Value	Negate	Mode
1	_num_bands	=	3	□	Automatic
2		=		□	Automatic

Now repeat that process to test for `_spacing_x = 1` and `_spacing_y = 1`



	Left Value	Operator	Right Value	Negate	Mode
1	_num_bands	=	3	□	Automatic
2	_spacing_x	=	1	□	Automatic
3	_spacing_y	=	1	□	Automatic

Finally, set the Pass Criteria parameter to All Tests (AND). If you don't then the incoming data will pass even if only one aspect of it is correct.



Pass Criteria

Pass Criteria: All Tests (AND)

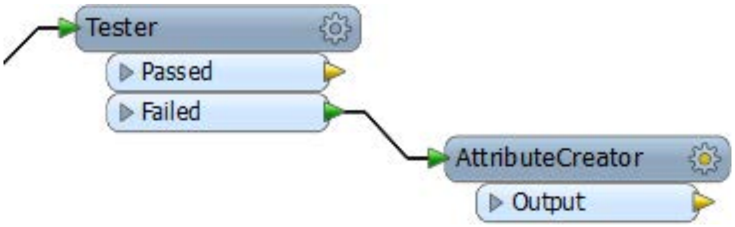
Composite Expression: <Unused>

Click OK to close the dialog. We now have a test to ensure the incoming data has the correct number of bands and the correct resolution.

6. Add AttributeCreator Transformers

We now have to decide how we are going to communicate the results of the translation to the end-user. The most effective way is perhaps to respond with a web page describing the results.

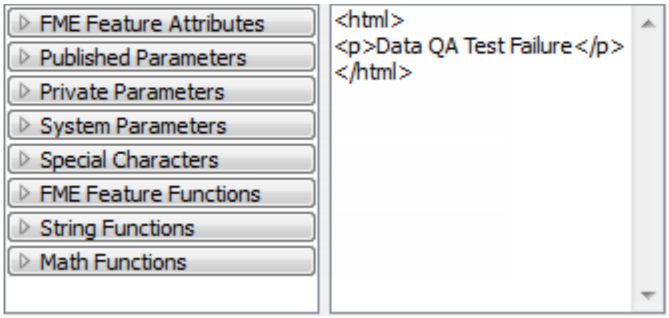
To generate some very simple HTML, we'll use an AttributeCreator transformer. So add an AttributeCreator transformer and connect it to the Tester:Failed port:



Now open the parameters for the AttributeCreator. Enter an attribute name of text_line_data (the name is important as we'll see shortly) and for the value click the […] button to open a text editor.

In the text editor, just enter some simple HTML such as:

```
<html>
<p>Data QA Test Failure</p>
</html>
```



```
<html>
<p>Data QA Test Failure</p>
</html>
```

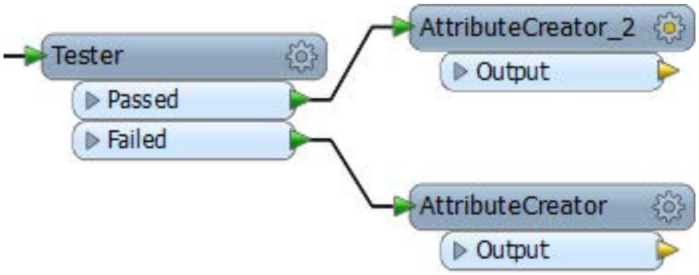
Click OK and then OK again to close the dialogs.

7. Duplicate AttributeCreator

Having created HTML for a failure we now need to do the same for a successful test. The simplest way to do this is select the existing AttributeCreator and press Ctrl+D to duplicate it (alternatively you can use Right Click > Duplicate).

Connect the new transformer to the Tester:Passed port.

Now open the parameters dialog and change the message inside from Failure to Success.



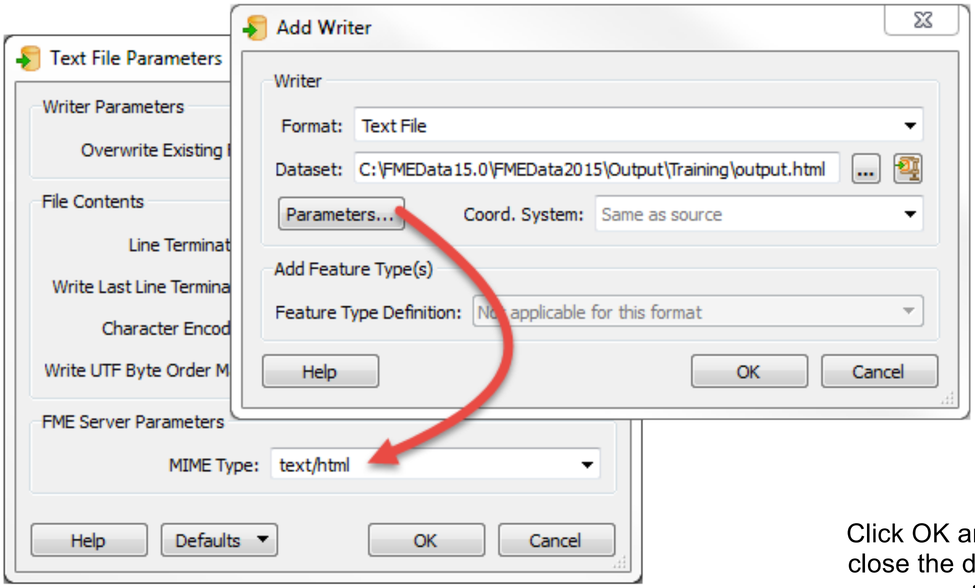
8. Add Writer

To output HTML in a way that will open automatically, we need to stream it through a Text File writer. So, select Writers > Add Writer from the menubar.

When prompted create a Writer as follows:

Reader Format	Text File
Reader Dataset	C:\FMEData2015\Output\Training\output.html
Parameters	
MIME Type	text/html

Really the output dataset is irrelevant here as it will never be used. The MIME type is set by clicking the Parameters button and using the setting inside that dialog.



Format: Text File

Dataset: C:\FMEData15.0\FMEData2015\Output\Training\output.html

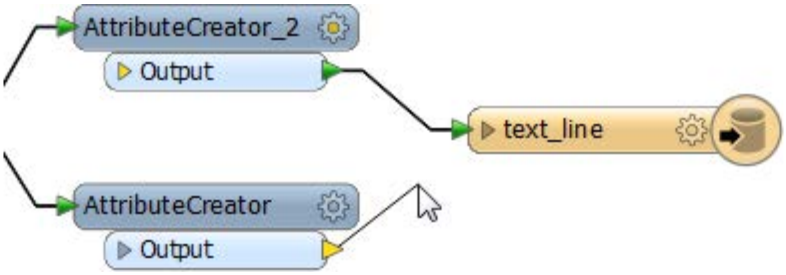
Coord. System: Same as source

Feature Type Definition: Not applicable for this format

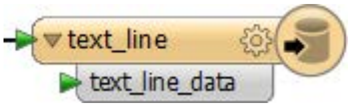
MIME Type: text/html

Click OK and OK again to close the dialogs and add the new Writer.

9. Connect Writer
Connect the Writer into the workspace by dragging connections from the output ports of both AttributeCreator transformers:

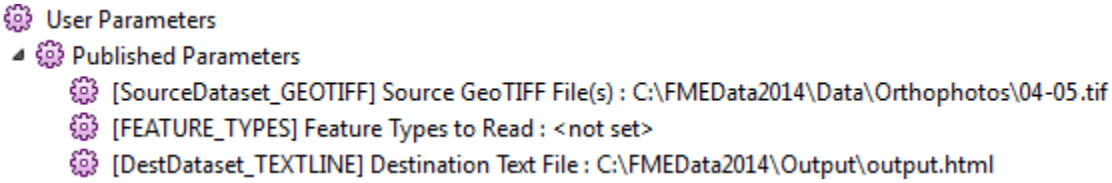


If you expand the text writer's attribute list you'll see it is expecting data to arrive in an attribute called text_line_data, which is why it was so important that the AttributeCreators create their attribute with that exact name.



10. Delete Existing Published Parameters
The workspace is now complete except for one thing, published parameters. Remember these are how the user is prompted to make a selection to control the translation.

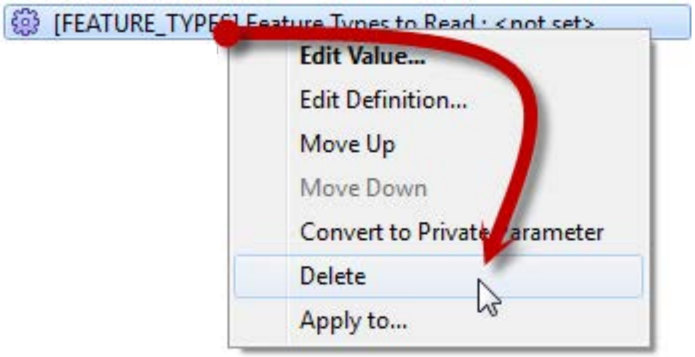
In this case we wish for the user to be given the chance to upload data, but not anything else. However, if you look in the Navigator window of Workbench you'll see that some parameters in a workspace are published by default.



User Parameters
Published Parameters
[SourceDataset_GEOTIFF] Source GeoTIFF File(s) : C:\FMEData2014\Data\Orthophotos\04-05.tif
[FEATURE_TYPES] Feature Types to Read : <not set>
[DestDataset_TEXTLINE] Destination Text File : C:\FMEData2014\Output\output.html

The source parameter is great, as it will let the user upload a source dataset; but the other parameters are not required. They will confuse the user and have no effect on the translation.

So delete these two parameters. You can either click them and press the delete key, or right-click them and select delete.


[FEATURE_TYPES] Feature Types to Read : <not set>
Edit Value...
Edit Definition...
Move Up
Move Down
Convert to Private Parameter
Delete
Apply to...

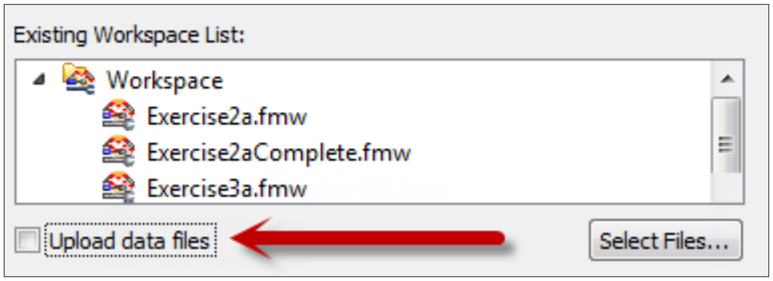
11. Test Run Workspace

Test run the workspace in FME Workbench to ensure it runs to completion. It will create a HTML file as output, though it won't open it automatically because only FME Server can do that.

12. Publish to FME Server

Save the workspace as Exercise3a.fmw. Then select File > Publish to FME Server from the menubar to start the publishing process.

As before (in Chapter 2) enter your connection parameters and then select the Training repository as the location to store the workspace. However, in that dialog be sure to uncheck the box labelled "Upload data files". The end-user will be uploading the files; we don't need to.


Existing Workspace List:

- Workspace
  - Exercise2a.fmw
  - Exercise2aComplete.fmw
  - Exercise3a.fmw

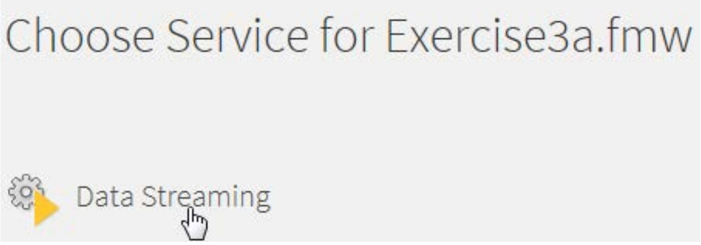
□ Upload data files

For the Service to register against, uncheck Job Submitter (and any other services) and check Data Streaming. We'll cover Data Streaming in more detail later in this chapter; for now just click **Publish** to finish publishing the workspace.

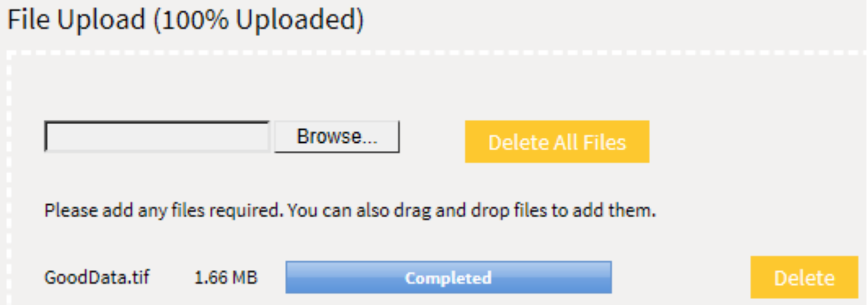
13. Run Workspace

Now – in the role of a data supplier – open the FME Server Web Interface in a web browser.

Browse to the newly uploaded workspace (Workspaces > Training > Exercise3a.fmw) and click Data Streaming:


Choose Service for Exercise3a.fmw

In the configuration page, under File Upload select Browse and add the file C:\FMEData2015\Resources\Self-Serve\GoodData.tif. It will then be uploaded to FME Server and become available for use.

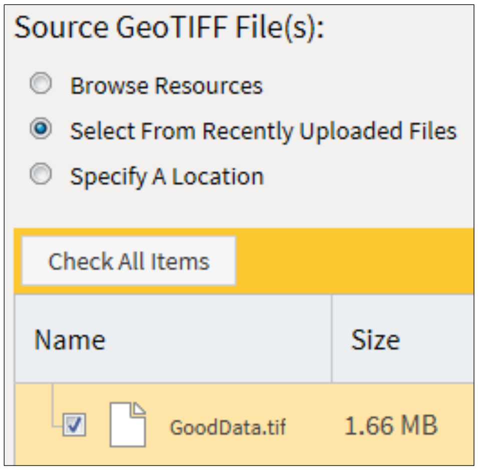

File Upload (100% Uploaded)

Please add any files required. You can also drag and drop files to add them.

GoodData.tif	1.66 MB	Completed	Delete

This is the key part of the process: the ability for an end-user to upload data to process.

Under Source GeoTIFF File(s), choose Select From Recently Uploaded Files for the source GeoTIFF data and select the newly uploaded *GoodData.tif*



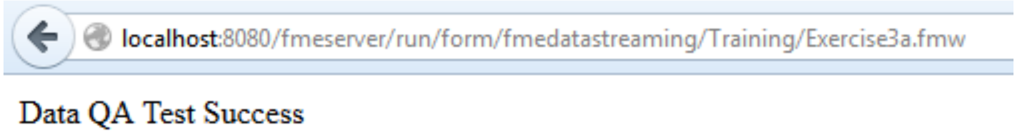
Source GeoTIFF File(s):

○ Browse Resources
● Select From Recently Uploaded Files
○ Specify A Location

Name	Size
☑ GoodData.tif	1.66 MB

Notice there are no other published parameters as there is nothing else we wish the end-user to have control over.

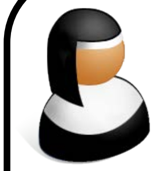
Now click the Run Workspace button. The translation should run to completion and a brief message given to alert us to the status of the dataset.



localhost:8080/fmeserver/run/form/fmedatastreaming/Training/Exercise3a.fmw

Data QA Test Success

You can now run the same process on the file *BadData.tif* to prove that it fails the tests.



Sister Intuitive says...

'Yes, the output from this exercise is sinfully basic, but we're not here to teach you HTML! Now you know how to create a Data Upload service from the FME Server side you can apply your web skills to producing a more aesthetic result – and even a custom form for uploading the data.

Additionally, you can add extra tests and functionality inside the FME workspace in order to carry out more QA. The GeometryValidator transformer is a great solution for dealing with vector data.'

Data Download Service

Data Downloads

What is the Data Download Service?
Creating a Data Download Service
Using a Data Download Service
Exercise 3b: Data Download

FME Server Basic Training 2015

The Data Download service is the key tool for Self-Serve setups

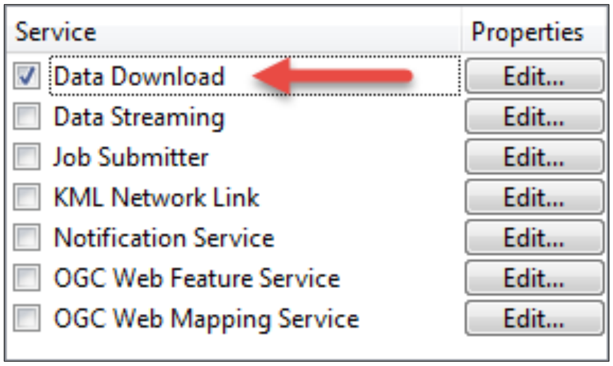
What is the Data Download Service?

When you run a workspace with the Job Submitter service (as in Chapter 2) the data is written to the location specified by the workspace; for example a file, directory, or database.

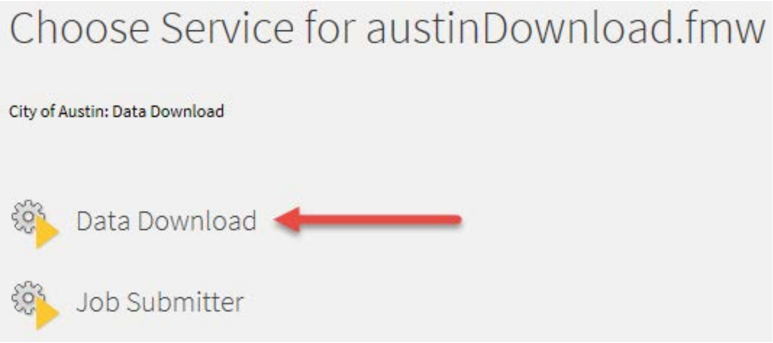
The Data Download service instead writes the output to a zip file and presents the user with a link to that file. This makes it ideal for self-serve, because the data is delivered directly to the user.

Creating a Data Download Service

A workspace becomes available for data download use when the author registers it against the Data Download service when publishing it to FME Server.



Service	Properties
☑ Data Download	Edit...
□ Data Streaming	Edit...
□ Job Submitter	Edit...
□ KML Network Link	Edit...
□ Notification Service	Edit...
□ OGC Web Feature Service	Edit...
□ OGC Web Mapping Service	Edit...



Choose Service for austinDownload.fmw

City of Austin: Data Download

Data Download

Job Submitter

Using a Data Download Service

Once registered this way, Data Download options become available in the FME Server Web interface for that particular workspace.

The workspace can be run through the web user interface, or using the developer URL information.

Monsieur D. Server says…

'Behind the scenes, the Data Download service overrides the Writer's destination parameter and writes a zip file to an internal folder, from where it can serve up the required data to the end-user in the form of a URL'

Exercise 3b: Data Download	
Scenario	Airphoto Data Vendor
Data	GeoTiff Orthophotos
Overall Goal	Create a data download service
Demonstrates	Authoring for and Using the Data Download Service
Starting Workspace	C:\FMEData2015\Workspaces\ServerAuthoring\Exercise3b-Begin-DataDownload.fmw
Finished Workspace	C:\FMEData2015\Workspaces\ServerAuthoring\Exercise3b-Complete-DataDownload.fmw

This exercise involves making orthophoto data available through a Data Download service. We will also mosaic the various files together and resample them to a lower resolution.

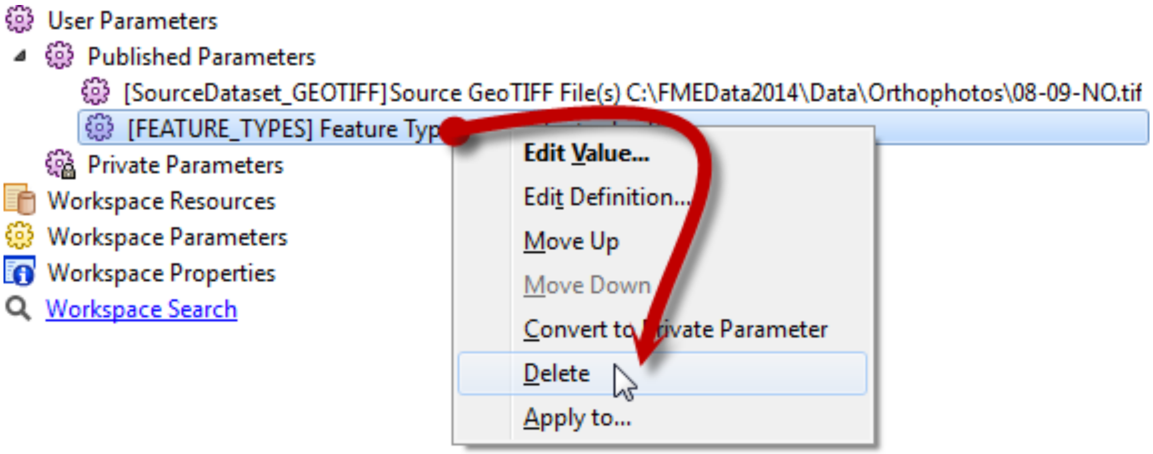
1. Start Workbench

Start Workbench (if necessary) and open the starting workspace for this Exercise. You will be familiar with it as it is the same workspace we created in Chapter 2.

2. Delete Existing Published Parameters

Some parameters in a workspace are published by default. This lets the user select the output dataset (for example) when the workspace is run. However, we don't need to let the user set any parameters and so best practice suggests we remove them.

Browse the Navigator window and expand the section labelled User Parameters - Published Parameters. Right-click and delete the the Feature Types to Read parameter.



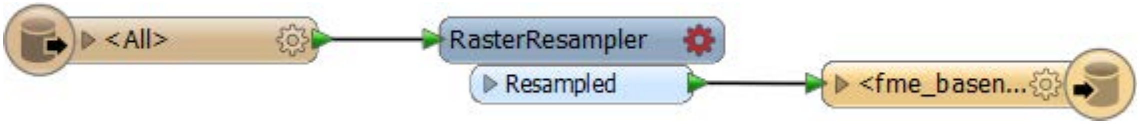
Only keep the published parameter for the source dataset.

3. Add a RasterResampler Transformer

Now we can add transformers to resample and mosaic the source data.

Click on the connection between the Reader and Writer feature types. Start typing “resampler”. A list of matching transformers will appear. When you see the RasterResampler transformer then select it.

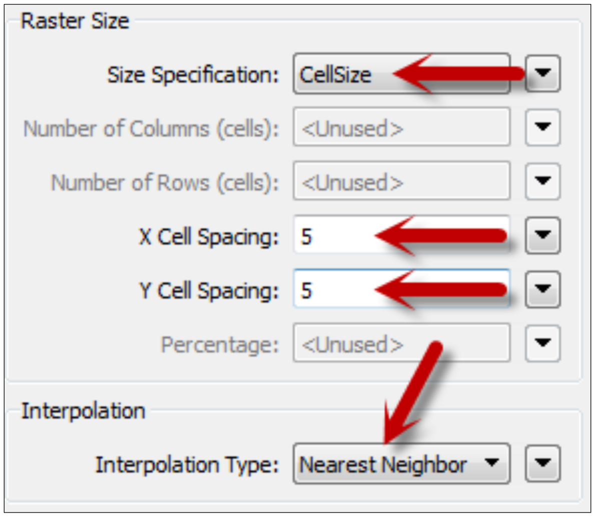
The transformer will now be added into the workspace.



Click the red cog wheel icon on the RasterResampler to open the parameters dialog.

Set the following values to resample the images from 1 metre resolution to 5 metres.

Size Specification:	Cell Size
X Cell Spacing:	5
Y Cell Spacing:	5
Interpolation Type:	Nearest Neighbor



Raster Size

Size Specification: CellSize

Number of Columns (cells): <Unused>

Number of Rows (cells): <Unused>

X Cell Spacing: 5

Y Cell Spacing: 5

Percentage: <Unused>

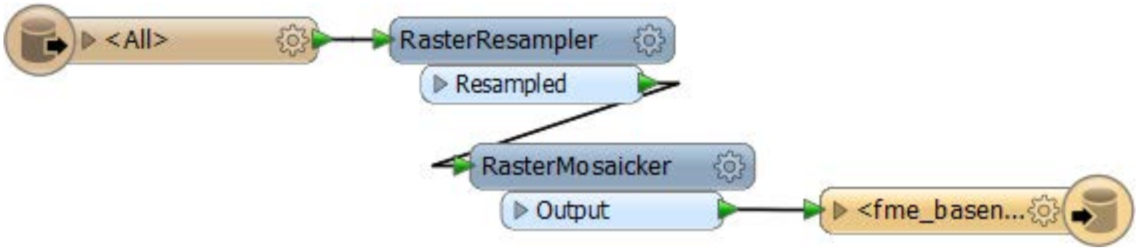
Interpolation

Interpolation Type: Nearest Neighbor

4. Add a RasterMosaicker Transformer

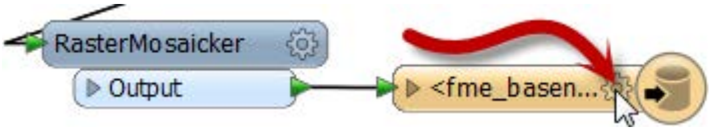
Now repeat step 3, but this time adding a RasterMosaicker transformer between the RasterResampler and the Writer feature type.

Open the parameters dialog as before, but now just click OK to accept the default values.

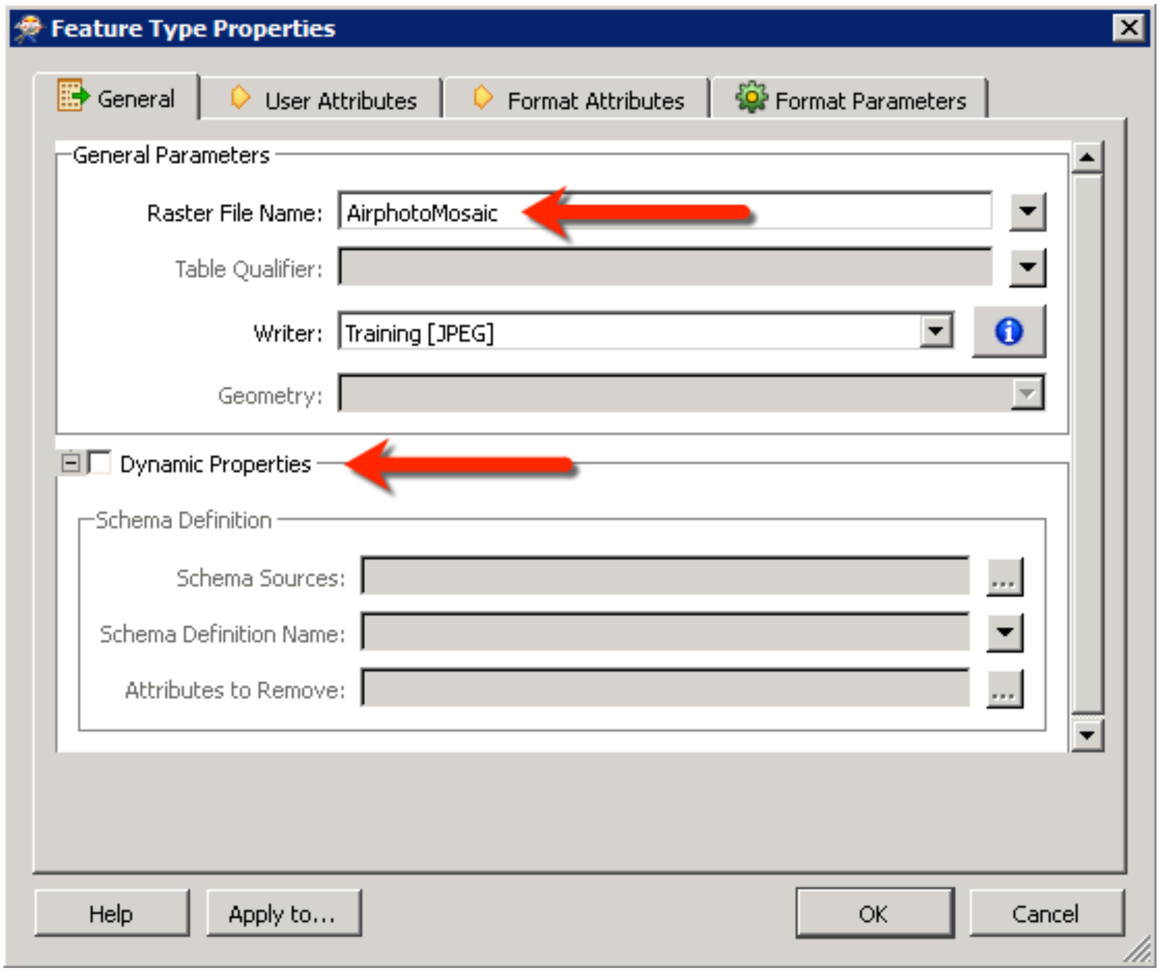


5. Modify the Destination Feature Type
In this scenario, we'll be writing a single (mosaicked) output file. This is at odds with the current dynamic settings that are part of the workspace.

So, click the cog wheel icon on the Writer feature type to open the properties dialog.



In this dialog uncheck the Dynamic Properties parameter and then type AirphotoMosaic for the Raster File Name.



Raster File Name: AirphotoMosaic

Table Qualifier:

Writer: Training [JPEG]

Geometry:

☐ Dynamic Properties

6. Publish the Workspace – Step 1

Save your workspace as Exercise3b.fmw

Start the "Publish to FME Server" process by using the toolbar button or File > Publish to FME Server from the menubar.

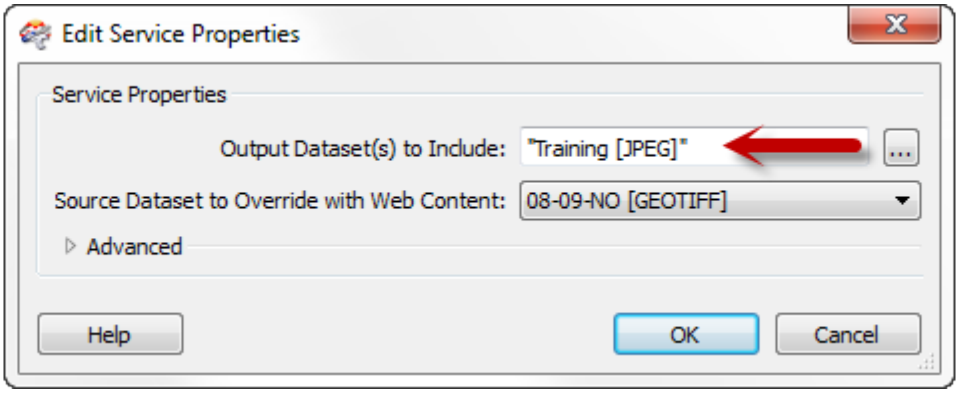
Enter the connection and credential settings and click **Next.** If you cannot remember them, or didn't save them as your defaults, refer to Exercise 2a for the required values.

For the repository select the Training repository you created in Exercise 2a. Click **Next**

7. Publish the Workspace – Step 2

In the last pane of the publish Wizard check the Data Download Service so that the workspace can be run with this service. You can also leave the Job Submitter Service checked so the workspace can be run via either service.

The default Data Download settings can be used but click the Edit button to review the possible settings anyway.



Edit Service Properties

Service Properties

Output Dataset(s) to Include: "Training [JPEG]"

Source Dataset to Override with Web Content: 08-09-NO [GEOTIFF]

The Output Dataset parameter is very important if there is more than one Writer in the workspace, as it specifies which is to be output for download. In this case it should be set to the JPEG writer.

Click **OK** to close this dialog and then **Publish** to finish publishing the workspace.

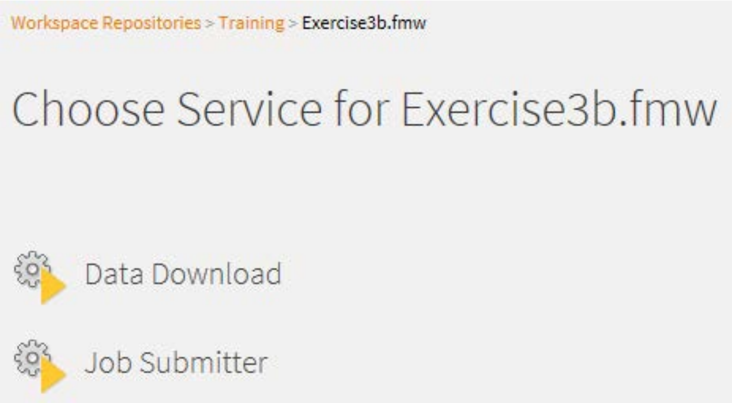
8. Access the Data Download Service

Now your role switches to that of an end-user who wants to download some data.

Log in to the FME Server Web User Interface. Use the menus and dialogs to navigate to Repositories > Training > Exercise3b.fmw

Notice that you can now run the workspace through either the Job Submitter or the Data Download Service.

Click Data Download.



Workspace Repositories > Training > Exercise3b.fmw

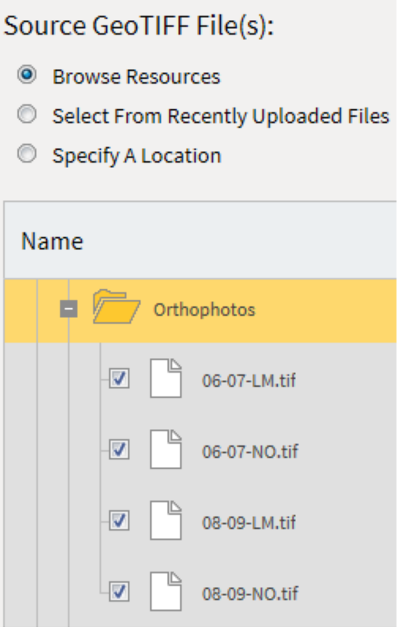
Choose Service for Exercise3b.fmw

Data Download

Job Submitter

9. Run the Workspace

Once again you are presented with a web form, but this time the form is for running the workspace via the Data Download Service.



Source GeoTIFF File(s):

● Browse Resources

○ Select From Recently Uploaded Files

○ Specify A Location

Name

Orthophotos

- ☑ 06-07-LM.tif
- ☑ 06-07-NO.tif
- ☑ 08-09-LM.tif
- ☑ 08-09-NO.tif

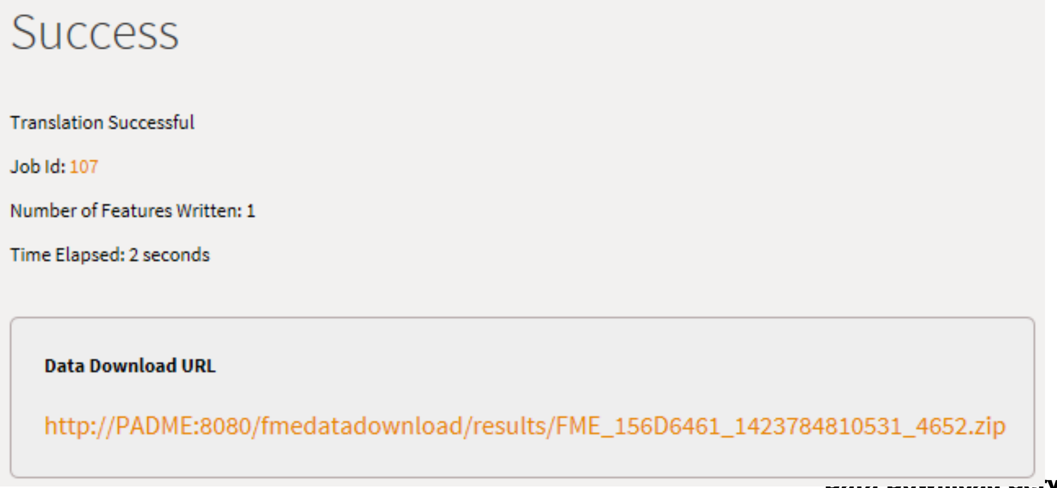
As you did in Exercise 2c, scroll down to the Source GeoTIFF parameter and click Browse Resources to browse to the image files you uploaded previously.

Select all of the images (or just a few that neighbor each other).

Click Run Workspace on the bottom right to submit the job via the Data Download Service.

10. Download the Output

After a short time a Success page will appear which now includes a Data Download URL link.



Success

Translation Successful

Job Id: 107

Number of Features Written: 1

Time Elapsed: 2 seconds

Data Download URL

http://PADME:8080/fmedatadownload/results/FME_156D6461_1423784810531_4652.zip

Click the link and download the zip file to your computer. Extract the contents of the zip file.

Notice that you have downloaded a single mosaicked JPEG dataset which has been resampled from the original source GeoTiff files.

Congratulations! You have now created a self-serve data download service for airphotos, which can be used by your customers.

Miscellaneous Published Parameters

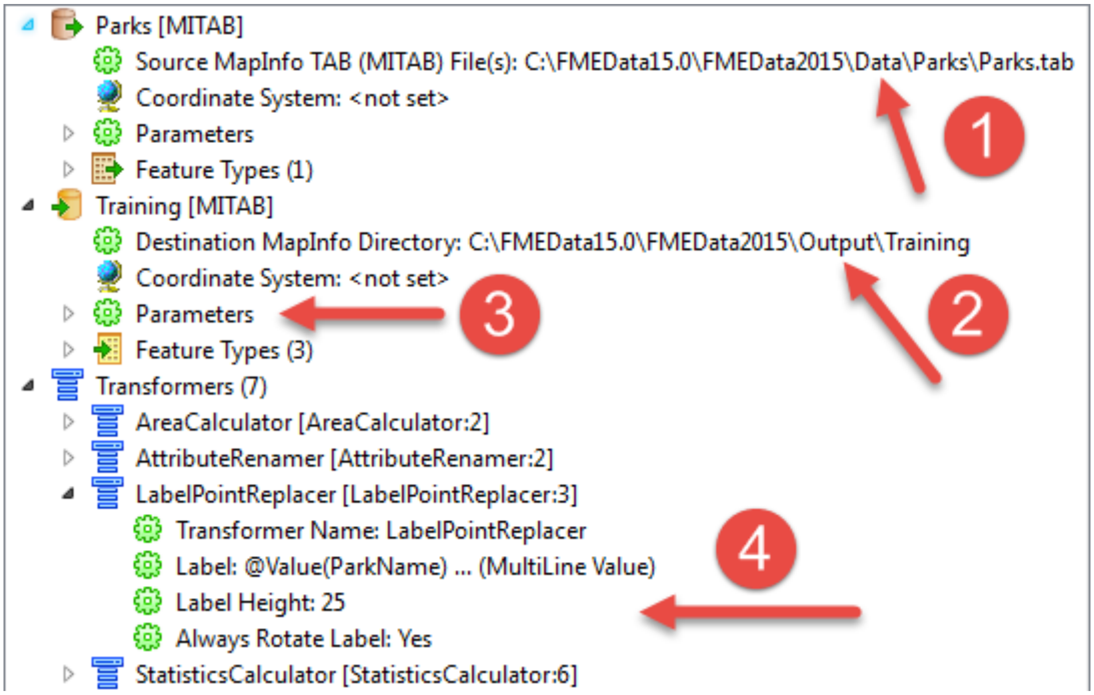
Self-Serve parameters can include any regular parameters, not just ones that are specific only to a Data Download setup

Any parameter in FME can be published and presented to the user as a choice to be made when running a workspace. They don't have to be specifically related to the Data Download service.

Finding a Parameter

Parameters are located in the Navigator window (usually on the left-hand side of FME Workbench).

As you can see, there are parameters for the source data location (1) and the destination data dataset (2), which is the one that the Data Download service overrides. There are also general Reader and Writer parameters (3) and parameters for transformers (4)

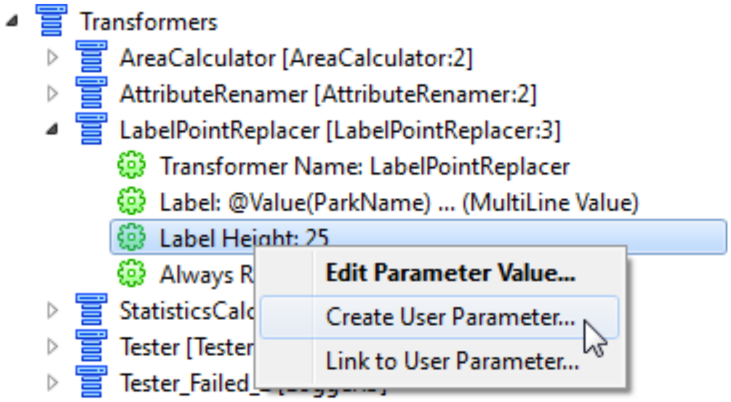


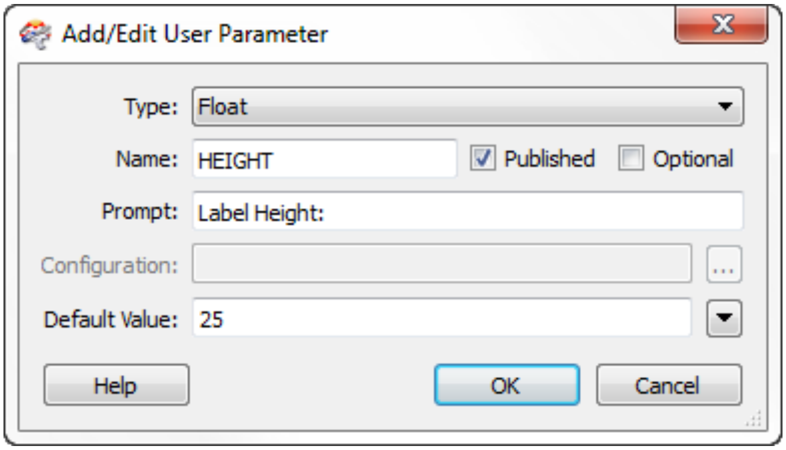
- Parks [MITAB]
  - Source MapInfo TAB (MITAB) File(s): C:\FMEData15.0\FMEData2015\Data\Parks\Parks.tab
  - Coordinate System: <not set>
  - Parameters
  - Feature Types (1)
- Training [MITAB]
  - Destination MapInfo Directory: C:\FMEData15.0\FMEData2015\Output\Training
  - Coordinate System: <not set>
  - Parameters
  - Feature Types (3)
- Transformers (7)
  - AreaCalculator [AreaCalculator:2]
  - AttributeRenamer [AttributeRenamer:2]
  - LabelPointReplacer [LabelPointReplacer:3]
    - Transformer Name: LabelPointReplacer
    - Label: @Value(ParkName) ... (MultiLine Value)
    - Label Height: 25
    - Always Rotate Label: Yes
  - StatisticsCalculator [StatisticsCalculator:6]

Publishing a Parameter
Parameters are published by right-clicking on them and choosing “Create User Parameter”.

For example, here a workspace author is about to publish a parameter on the LabelPointReplacer transformer.

When the workspace is run the end-user will now be able to decide what size they want labels to appear as.

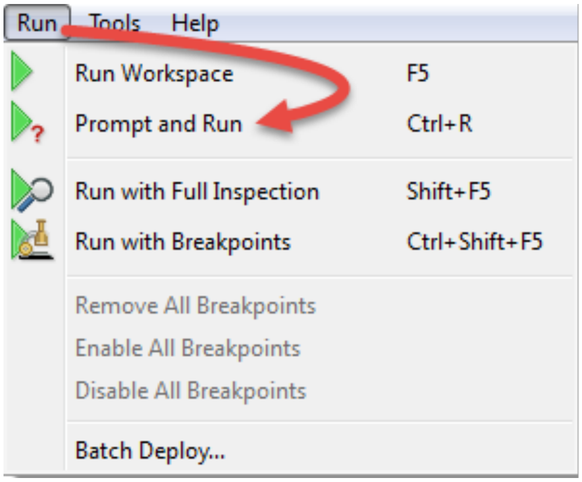

Transformers
AreaCalculator [AreaCalculator:2]
AttributeRenamer [AttributeRenamer:2]
LabelPointReplacer [LabelPointReplacer:3]
Transformer Name: LabelPointReplacer
Label: @Value(ParkName) ... (MultiLine Value)
Label Height: 25
Always R
StatisticsCalc
Tester [Tester
Tester_Failed_
Edit Parameter Value...
Create User Parameter...
Link to User Parameter...



Type: Float
Name: HEIGHT ☑ Published □ Optional
Prompt: Label Height:
Configuration:
Default Value: 25

The author gets to choose the type of parameter (in this case a floating point number), the user prompt, and the default value.

Using a Parameter
A User Parameter that is published becomes available to the end-user at run-time.



In FME Workbench, you would select Run > Prompt and Run from the menubar, to see these parameters. There is also a toolbar button and a keyboard shortcut (Ctrl+R).

In FME Server, these parameters become available on the Configuration page of a service, as seen in the Chapter 2 exercises.

Key Parameter Types

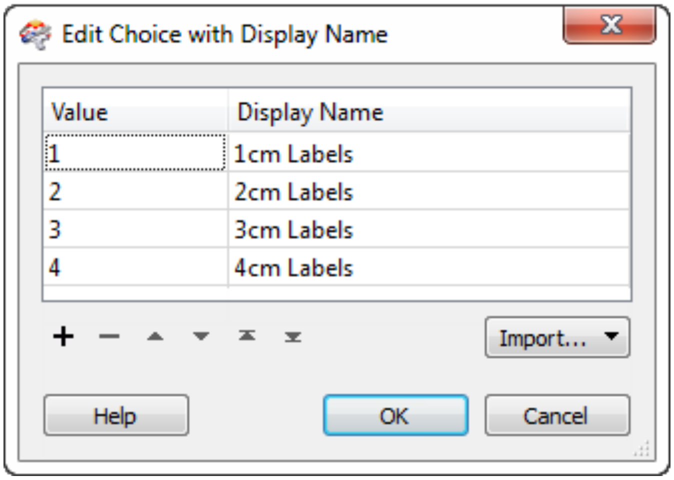
Two key parameter types for a self-serve setup are “Choice with Alias” and “Scripted”.

Choice with Alias parameters are where the user is presented with a choice, each of which returns a different value to FME.

For example, here the author wishes the user to select the height of labels to be created in the output dataset.

The user is presented with the list of (more intuitive) options on the right, but the value returned to FME will be the (more useful) number on the left.

This parameter type is also very useful for some Data Download specific tasks, as we’ll see shortly.



Value	Display Name
1	1cm Labels
2	2cm Labels
3	3cm Labels
4	4cm Labels

Scripted parameters are a way to incorporate a Python or Tcl script into the generation of a parameter value.

This script can include references to other user parameters, meaning one parameter value can be calculated based on the value of another. Parameters referred to in the scripts are generally published parameters that accept input from the user.

So, user input can be accepted from one parameter, and this input processed using a script, in order to produce the actual information required by the workspace.

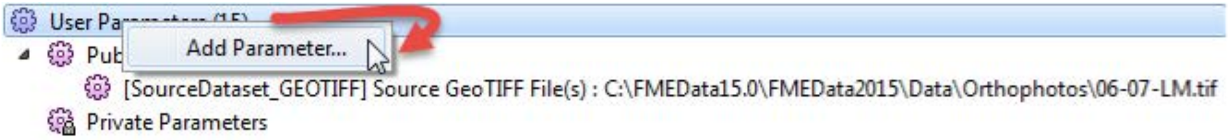
Exercise 3c: Data Download Parameters	
Scenario	Airphoto Data Vendor
Data	GeoTiff Orthophotos
Overall Goal	Improve the data download service
Demonstrates	Providing User Parameters in a Data Download service
Starting Workspace	C:\FMEData2015\Workspaces\ServerAuthoring\Exercise3c-Begin-DataDownload.fmw
Finished Workspace	C:\FMEData2015\Workspaces\ServerAuthoring\Exercise3c-Complete-DataDownload.fmw

The Data Download system from Exercise 3b works, but could be improved upon. In particular, it would be helpful for users to be able to choose the resolution and compression of the imagery they are downloading.

1. Create a Published Parameter for Resolution

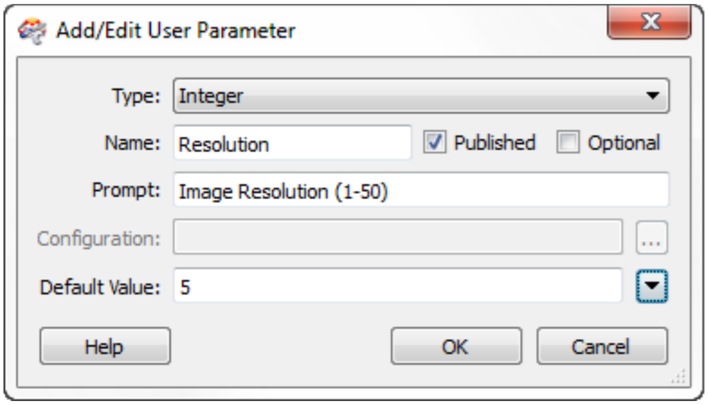
Start FME Workbench (if necessary). Open the workspace from Exercise 3b (or the start workspace for this exercise).

Locate the User Parameters section of the Navigator window. Right-click on the header for this section and choose Add Parameter.


User Pa...
Add Parameter...
Pub
[SourceDataset_GEOTIFF] Source GeoTIFF File(s) : C:\FMEData15.0\FMEData2015\Data\Orthophotos\06-07-LM.tif
Private Parameters

This opens the Add/Edit User Parameter dialog. In this dialog enter the following values:

Type	Integer
Name	Resolution
Published	Yes (Checked)
Optional	No (Unchecked)
Prompt	Image Resolution (1-50)
Default Value	5


Add/Edit User Parameter

Type: Integer

Name: Resolution ☑ Published □ Optional

Prompt: Image Resolution (1-50)

Configuration:

Default Value: 5

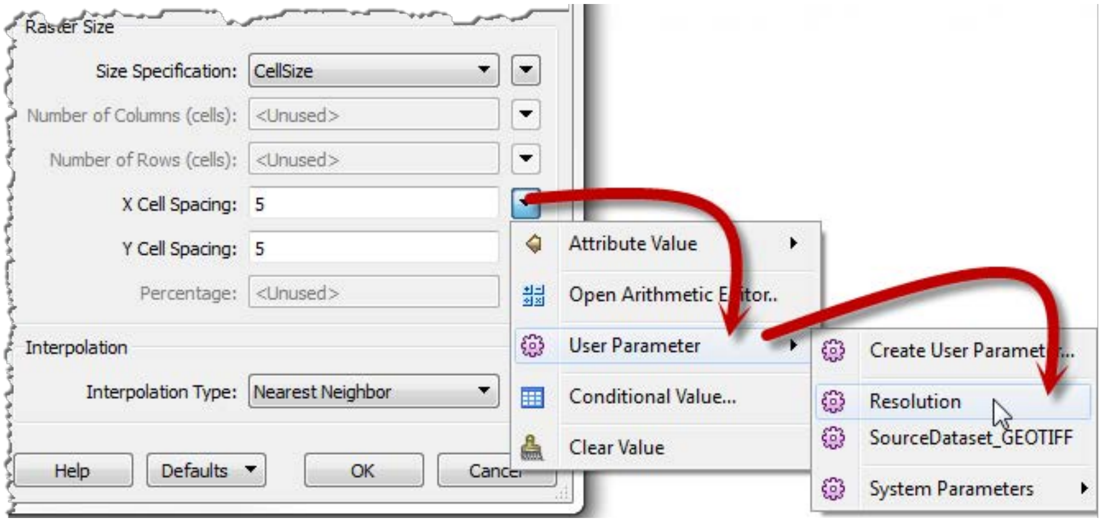
2. Apply Parameter to RasterResampler
Having created the parameter we can get input from the end-user, but, as-yet, the workspace isn't doing anything with that input. We must now apply the input to the transformers.

Click the cog wheel icon to open the parameters dialog for the RasterResampler transformer.

Notice that the X and Y Cell Spacing is currently hard-coded as 5.

Click the drop down arrow to the right of the X Cell Spacing parameter. Choose User Parameter > Resolution. This will apply the Resolution parameter you just created to this transformer.

Repeat the action for the Y Cell Spacing parameter.



Raster Size

Size Specification: CellSize

Number of Columns (cells): <Unused>

Number of Rows (cells): <Unused>

X Cell Spacing: 5

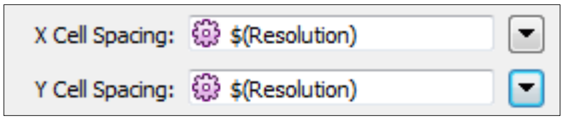
Y Cell Spacing: 5

Percentage: <Unused>

Interpolation

Interpolation Type: Nearest Neighbor

The transformer X/Y Cell parameters will now look like this:

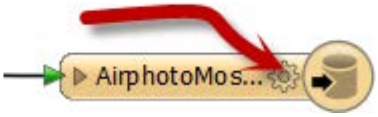


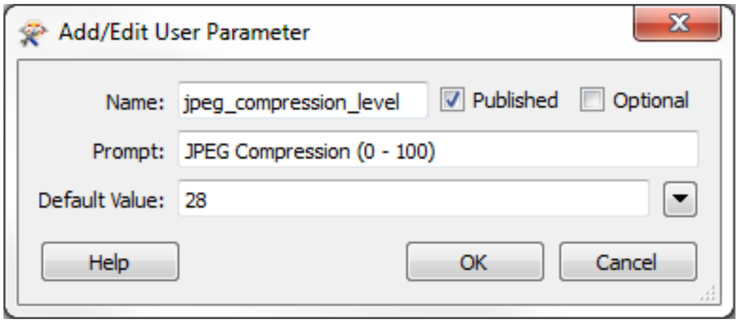
X Cell Spacing: $(Resolution)

Y Cell Spacing: $(Resolution)

3. Create a Published Parameter for JPEG Compression
Now we will create a parameter for the user to control the JPEG compression amount.

Open the properties dialog of the AirphotoMosiac feature type. Click on the Format Parameters tab.





Add/Edit User Parameter

Name: jpeg_compression_level ☑ Published □ Optional

Prompt: JPEG Compression (0 - 100)

Default Value: 28

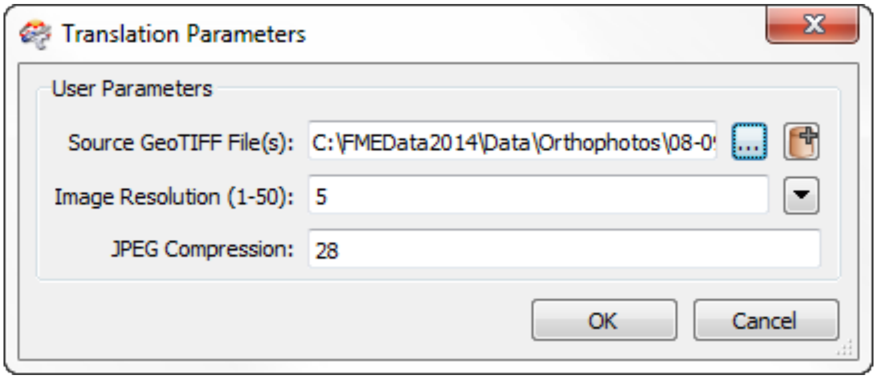
Click the drop down arrow beside Compression Level and choose User Parameter > Create User Parameter.

Change the prompt to JPEG Compression (0-100) and leave the default at 28.

4. Test your Workspace in FME Workbench

Save the workspace as Exercise3c.fmw.

As workspaces get more complex it is always a good idea to test them first in FME Desktop before publishing to FME Server. Use the menu Run - Prompt and Run (or Cntrl+R) to see the parameters which are now available. You should see your new parameters.



Translation Parameters

User Parameters

Source GeoTIFF File(s): C:\FMEData2014\Data\Orthophotos\08-0

Image Resolution (1-50): 5

JPEG Compression: 28

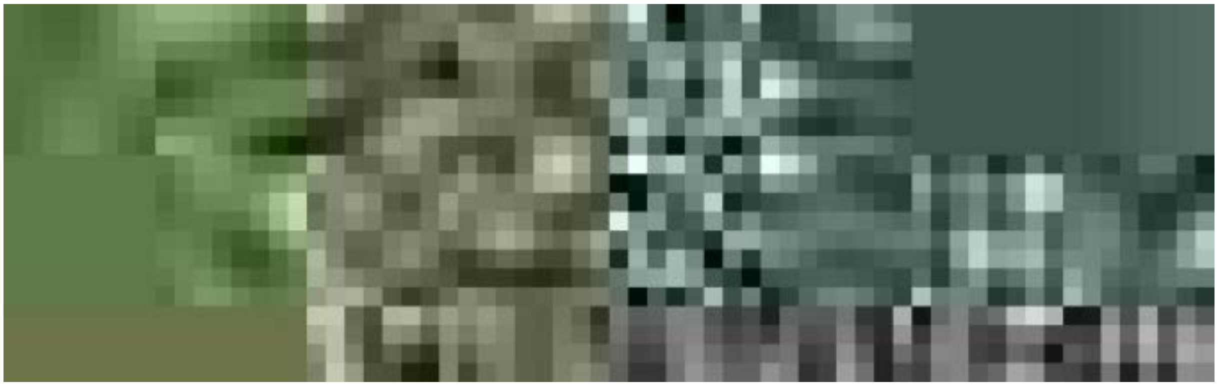
Click the file browse button and select some neighboring image files, for example 06-07-LM.tif and 06-07-NO.tif.

Enter some extreme values for your new parameters so it will be obvious that the output has been changed, for example:

Image Resolution	50
Compression Level	100

Click **OK** to run the workspace.

Check the output; you should have a single very low resolution, poor quality image.



If you ever need to change the parameter prompts or defaults then browse to the parameters in the Navigator window and right-click the one you want to change. Select Edit Definition and this will open a dialog in which to update the definition.

5. Publish the Workspace

Use the publish wizard to publish your workspace to FME Server. You don't need to upload any data (assuming you uploaded it in the earlier exercises), but be sure you check Data Download Service in the last window.

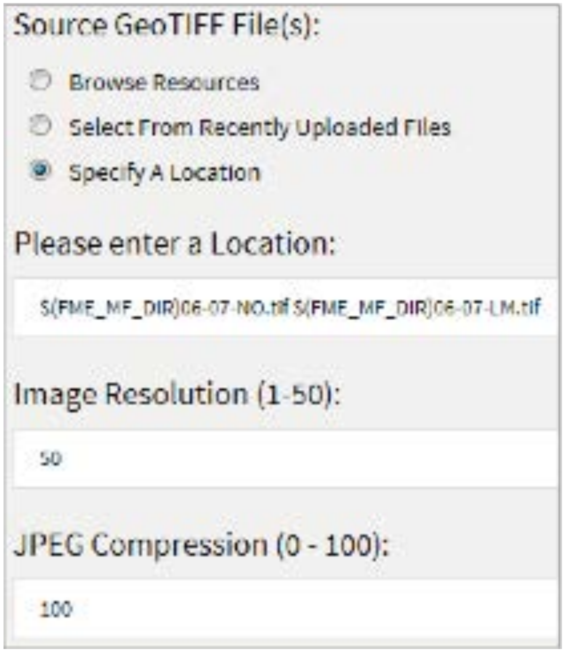
6. Run Data Download Using New Parameters

Acting as a user now, visit the Web User Interface and navigate to your workspace:
Workspaces > Training > Exercise3c.fmw > Data Download

Notice your new parameters are now shown automatically in the web form.

As before, select some or all of the source image files under Browse Resources.

Scroll down and enter reasonable values for the Image Resolution and the JPEG Compression. Click **Run Workspace** on the bottom of the form.



Source GeoTIFF File(s):

○ Browse Resources

○ Select From Recently Uploaded Files

● Specify A Location

Please enter a Location:

$(FME_MF_DIR)06-07-NO.tif $(FME_MF_DIR)06-07-LM.tif

Image Resolution (1-50):

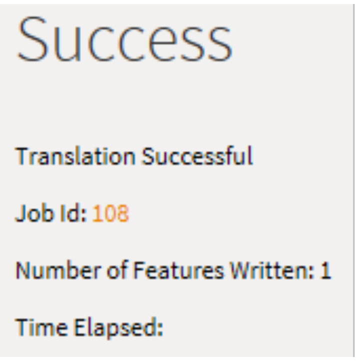
50

JPEG Compression (0 - 100):

100

7. Download the Output

The Success page will appear with a link to download the data. Download and extract the zip file contents to ensure it includes an image based on the specifications you entered in the web form.



Success

Translation Successful

Job Id: 108

Number of Features Written: 1

Time Elapsed:

Congratulations! You now have a Data Download Service where your customers can choose the resolution and compression for imagery they want to access.

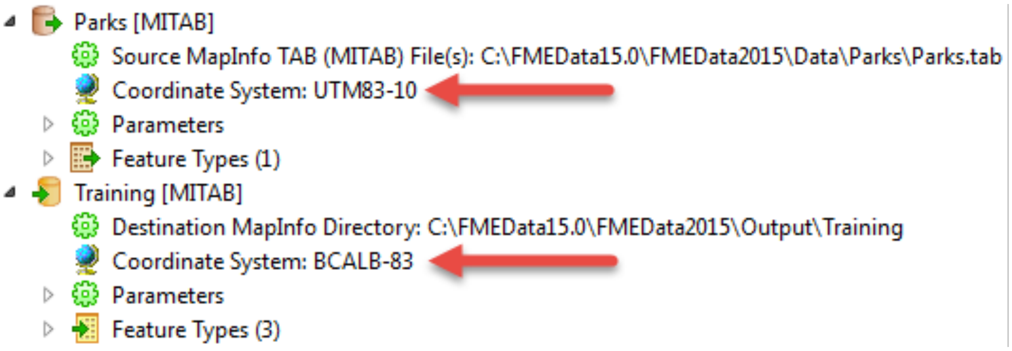
Coordinate System Selection

One of the most obvious requirements in Self-Serve is to be able to set the output coordinate system

One of the most useful parameters in a self-serve system controls the output coordinate system. Publishing this parameter allows the end-user to receive data in a coordinate system of their choice.

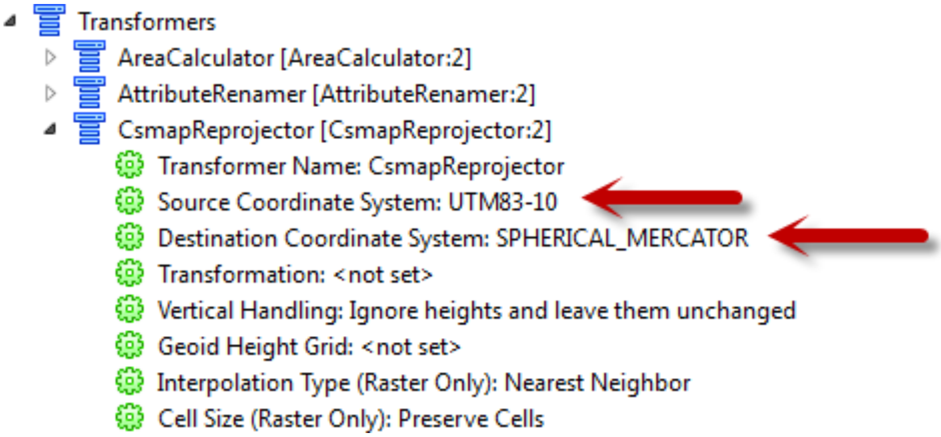
Coordinate System Parameters

Coordinate system reprojection can be carried out by setting the parameters under the Reader/Writer in the Navigator window:



Parks [MITAB]
Source MapInfo TAB (MITAB) File(s): C:\FMEData15.0\FMEData2015\Data\Parks\Parks.tab
Coordinate System: UTM83-10
Parameters
Feature Types (1)
Training [MITAB]
Destination MapInfo Directory: C:\FMEData15.0\FMEData2015\Output\Training
Coordinate System: BCALB-83
Parameters
Feature Types (3)

Alternatively, a transformer – such as the CSMapReprojector – can be used, in which case the relevant parameters can be found under the Transformers section of the Navigator window:



Transformers
AreaCalculator [AreaCalculator:2]
AttributeRenamer [AttributeRenamer:2]
CsmapReprojector [CsmapReprojector:2]
Transformer Name: CsmapReprojector
Source Coordinate System: UTM83-10
Destination Coordinate System: SPHERICAL_MERCATOR
Transformation: <not set>
Vertical Handling: Ignore heights and leave them unchanged
Geoid Height Grid: <not set>
Interpolation Type (Raster Only): Nearest Neighbor
Cell Size (Raster Only): Preserve Cells

The obvious advantage to using a transformer is that you have control over other reprojection factors, such as the geographic transformation and grid height.

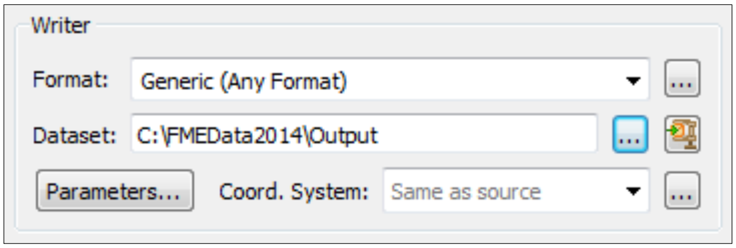
Format Selection

Another obvious requirements in Self-Serve is to be able to select the format of data to be downloaded

One of the most useful parameters in a self-serve system controls the output data format. With a simple Generic format, FME allows the end-user to select this format on the fly, without any format-specific setup from the workspace author.

The Generic Writer

The Generic writer is a way to write data in FME where the output format is chosen at run-time. This way a workspace may be created that can write to any format.



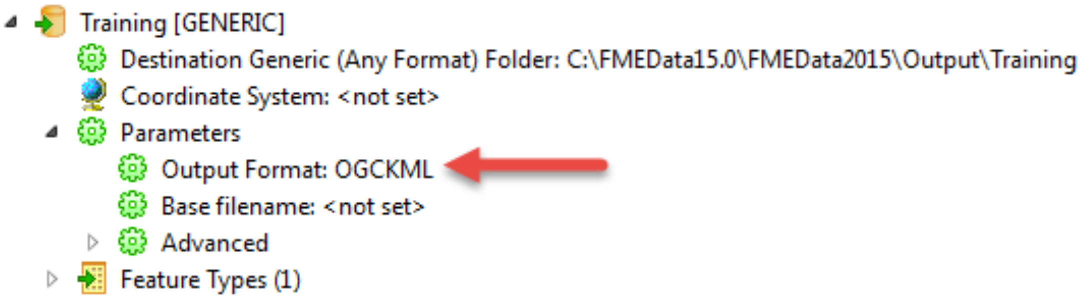
Writer

Format: Generic (Any Format)

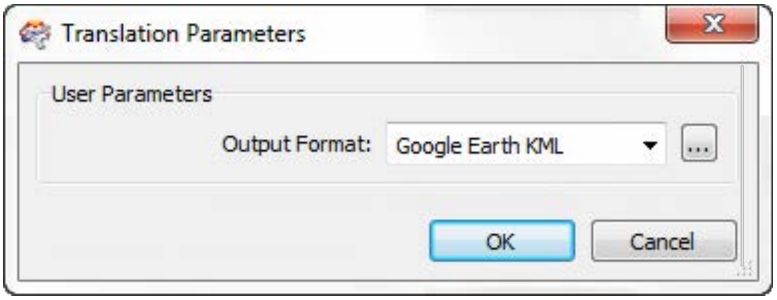
Dataset: C:\FMEData2014\Output

Coord. System: Same as source

The output format is defined by a parameter so, by publishing the parameter, the user gets to choose the output format at run-time.



```
Training [GENERIC]
    Destination Generic (Any Format) Folder: C:\FMEData15.0\FMEData2015\Output\Training
    Coordinate System: <not set>
    Parameters
        Output Format: OGCKML
        Base filename: <not set>
        Advanced
    Feature Types (1)
```



Translation Parameters

User Parameters

Output Format: Google Earth KML

There are a couple of points to keep in mind when using the writer:

- The parameter for output format is automatically published, but in FME Server's interface it appears only as a text field that accepts FME's short form name for any format. Best practice decrees this should usually be replaced by a new Choice with Alias parameter; to reduce the list to a reasonable set of choices, with descriptive format names.

- Each writer format has its own specific parameters, and these may still need to be set when a generic writer is used. For example, a specific template file might need to be used when AutoCAD is chosen as the output format. This can be achieved by adding a writer of the same format and setting the parameters in that writer. The Generic writer will inherit the parameters of this dummy writer, even if no features are connected to it.

Exercise 3d: Data Download	
Scenario	Airphoto Data Vendor
Data	GeoTiff Orthophotos
Overall Goal	Improve the data download service
Demonstrates	Providing Coordinate System and Format Parameters in a Data Download service
Starting Workspace	C:\FMEData2015\Workspaces\ServerAuthoring\Exercise3d-Begin - DataDownload.fmw
Finished Workspace	C:\FMEData2015\Workspaces\ServerAuthoring\Exercise3d-Complete - DataDownload.fmw

The Data Download system from Exercise 3c is still lacking some important user options. Users have asked if they can set the destination coordinate system and data format in the Data Download service.

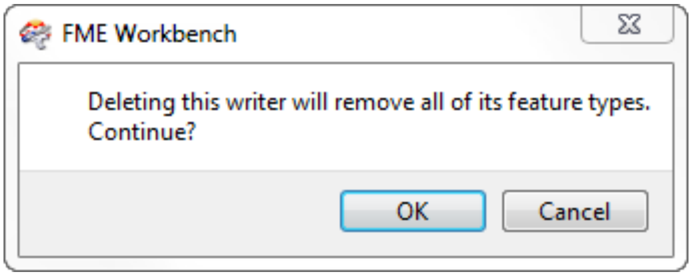
1. Start Workbench

Start FME Workbench (if necessary). Open the workspace from Exercise 3c (or the start workspace for this exercise).

2. Delete JPEG Writer

To allow selection of data format we'll need to remove the existing JPEG format Writer and replace with a Generic Writer.

Locate and click on the JPEG Writer in the Navigator window. Press the delete key to remove it. You will receive a prompt confirming your actions. Click OK to accept the deletion.



FME Workbench

Deleting this writer will remove all of its feature types. Continue?

3. Add Generic Writer

Now the JPEG Writer is removed, we can add a Generic Writer.
Select Writers > Add Writer on the menubar. When prompted enter the following settings:

Format	Generic (Any Format)
Dataset	C:\FMEData2015\Output\Training

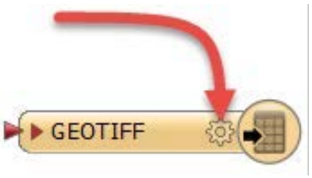
Click the **Parameters** button to ensure these are set. If no output format is already set then select a format (it doesn't matter what you choose) and click OK to close the dialog.

Under Add Feature Type(s), ensure Copy from Reader feature type definition is selected.
Click OK again to close the Add Writer dialog.

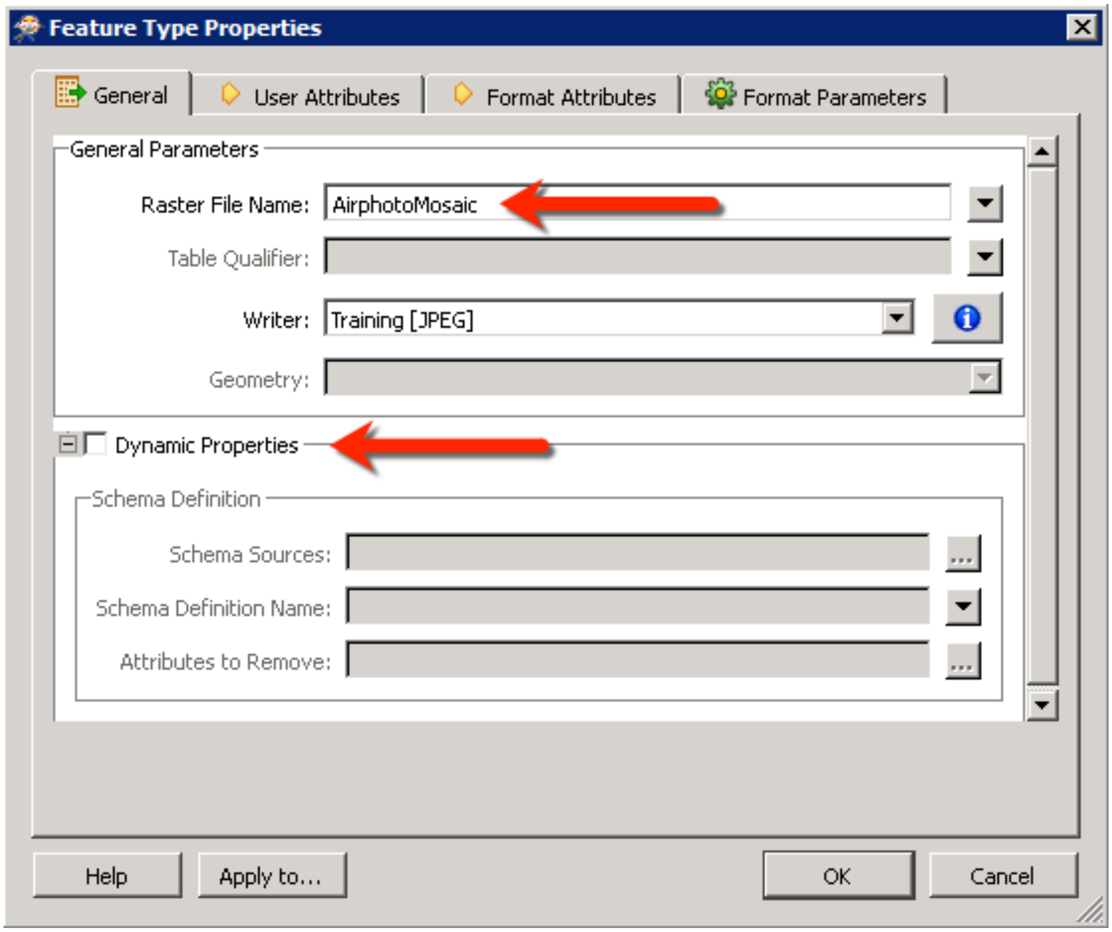
4. Edit Feature Type

When a new Writer is added with Copy from Reader selected, the writer inherits the schema of the specified reader feature type.

Click the cog wheel icon on the Writer feature type to open the properties dialog.



In this dialog uncheck the Dynamic Properties parameter and then type AirphotoMosaic for the Raster File Name.


Feature Type Properties

General Parameters

Raster File Name: AirphotoMosaic

Table Qualifier:

Writer: Training [JPEG]

Geometry:

☐ Dynamic Properties

Schema Definition

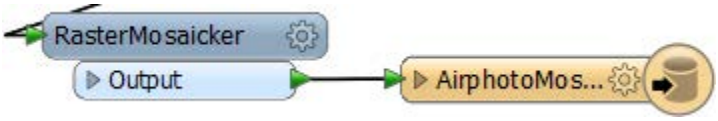
Schema Sources:

Schema Definition Name:

Attributes to Remove:

5. Connect Feature Type

In the workspace, connect the new feature type to the RasterMosaicker Output port.

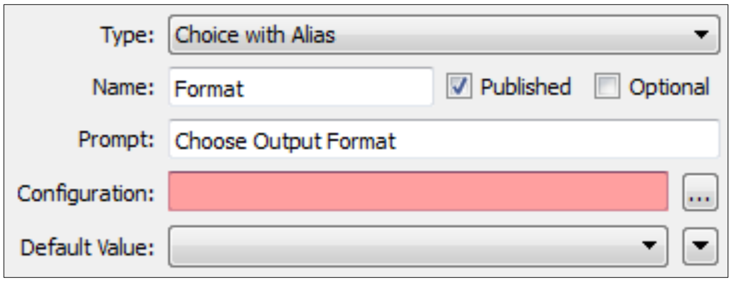


6. Create a Format Parameter

Now we've added a Generic format writer we should publish a parameter so that users can select the format to use. As in the previous example, right-click User Parameters in the Navigator window and select Add Parameter.

Use the following settings for your new parameter:

Type	Choice with Alias
Name	Format
Published	Yes
Optional	No
Prompt	Choose Format



Type: Choice with Alias
Name: Format ☑ Published □ Optional
Prompt: Choose Output Format
Configuration:
Default Value:

7. Configure Parameter
To configure the parameter click the ellipsis […] button next to the Configuration setting.

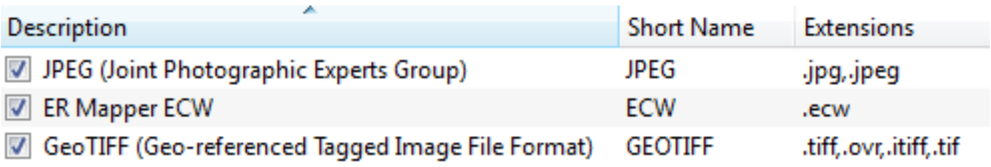
This will open a dialog in which to define the parameter configuration. Because this is a “Choice with Alias” parameter we need to define both values and a display name. However, since we are using this to select a format there is a nice shortcut that we can use to do so.

Click the Import drop-down and select Writer Format(s) from the menu. This provides a dialog with a list of Writer formats that we can add to our parameter.

Search for and add the following formats:

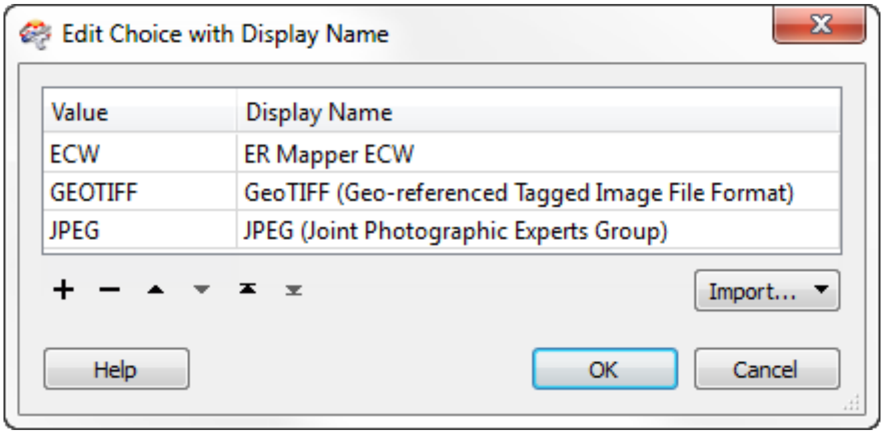
JPEG

ECW
GeoTiff



Description	Short Name	Extensions
☑ JPEG (Joint Photographic Experts Group)	JPEG	.jpg,.jpeg
☑ ER Mapper ECW	ECW	.ecw
☑ GeoTIFF (Geo-referenced Tagged Image File Format)	GEOTIFF	.tiff,.ovr,.itiff,.tif

Then click **OK** to close this dialog and return to the configuration dialog. It should look like this:



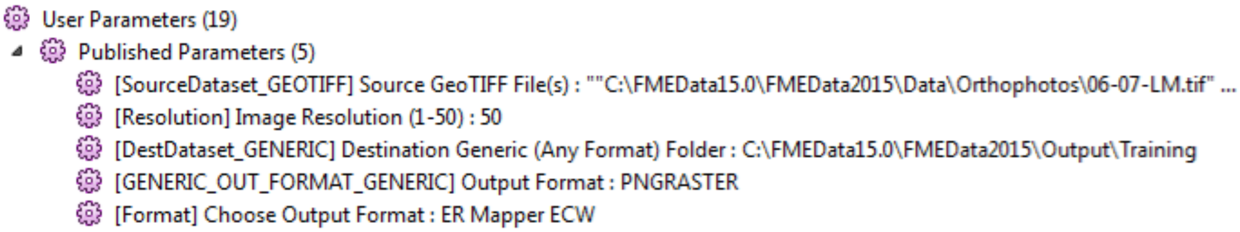
Edit Choice with Display Name

Value	Display Name
ECW	ER Mapper ECW
GEOTIFF	GeoTIFF (Geo-referenced Tagged Image File Format)
JPEG	JPEG (Joint Photographic Experts Group)

Notice the Display Name column allows you to provide your users a description or alias for each entry. Edit these in any way you think makes them easiest to understand and click **OK**.

Choose any of formats as the default and click **OK** to create the parameter.

The following should now be the list of current published parameters:

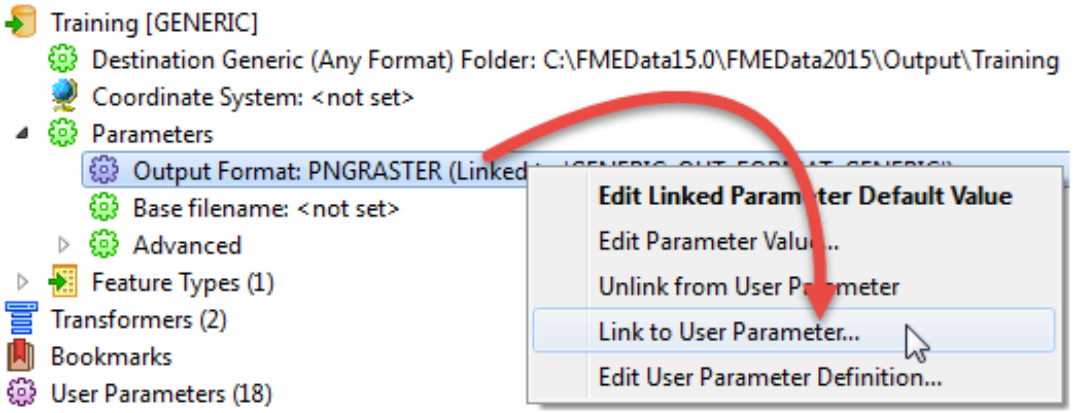


User Parameters (19)
- Published Parameters (5)
  - [SourceDataset_GEOTIFF] Source GeoTIFF File(s) : ""C:\FMEData15.0\FMEData2015\Data\Orthophotos\06-07-LM.tif" ...
  - [Resolution] Image Resolution (1-50) : 50
  - [DestDataset_GENERIC] Destination Generic (Any Format) Folder : C:\FMEData15.0\FMEData2015\Output\Training
  - [GENERIC_OUT_FORMAT_GENERIC] Output Format : PNGRASTER
  - [Format] Choose Output Format : ER Mapper ECW

8. Link the Custom Format Parameter

Now the format user parameter must be linked to the appropriate FME parameter.

In the Navigator window, locate the Generic Writer and expand its list of parameters. Locate the parameter Output Format and right-click upon it. Choose Link to User Parameter:

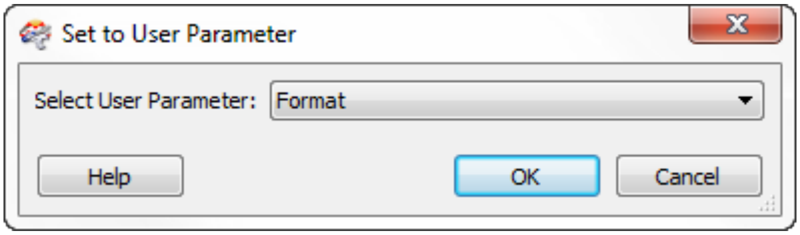


Training [GENERIC]
- Destination Generic (Any Format) Folder: C:\FMEData15.0\FMEData2015\Output\Training
- Coordinate System: <not set>
- Parameters
  - Output Format: PNGRASTER (Linked
  - Base filename: <not set>
  - Advanced
- Feature Types (1)

Transformers (2)
Bookmarks
User Parameters (18)

Edit Linked Parameter Default Value
Edit Parameter Value...
Unlink from User Parameter
Link to User Parameter...
Edit User Parameter Definition...

This will pop-up a list of user parameters. Select the newly created Format from the list.



Set to User Parameter

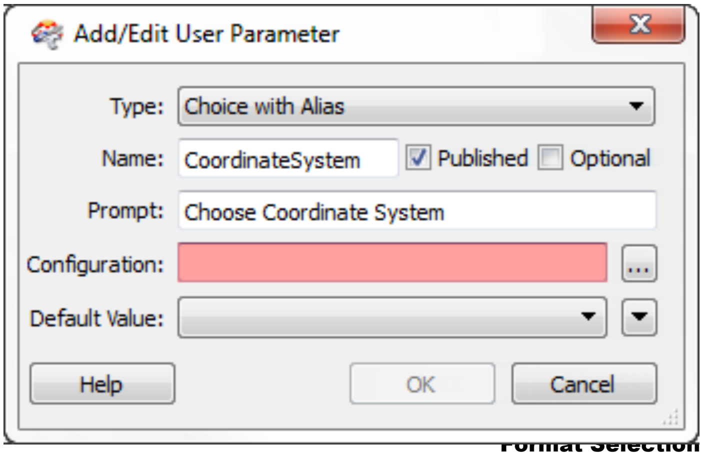
Select User Parameter: Format

9. Create a Coordinate System Parameter

Now we can create a parameter to control coordinate system.
Once again in the Navigator window locate User Parameters and right-click to Add Parameter.

Use the following settings for your new parameter:

Type	Choice with Alias
Name	CoordinateSystem
Prompt	Choose Coordinate System
Published	Yes
Optional	No



Add/Edit User Parameter

Type: Choice with Alias

Name: CoordinateSystem ☑ Published ☐ Optional

Prompt: Choose Coordinate System

Configuration:

Default Value:

10. Configure Parameter

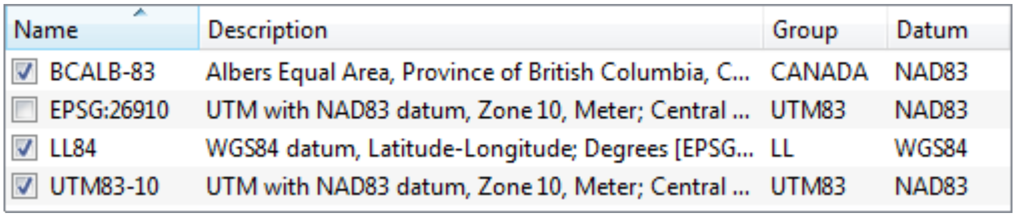
To configure the parameter click the ellipsis […] button next to the Configuration setting.

This will open a dialog in which to define the parameter configuration. Because this is a “Choice with Alias” parameter we need to define both values and a display name. However, since we are using this to select a coordinate system there is again a nice shortcut that we can use to do so.

Click the Import drop-down and select Coordinate System(s) from the menu. This provides a dialog with a list of coordinate systems that we can add to our parameter.

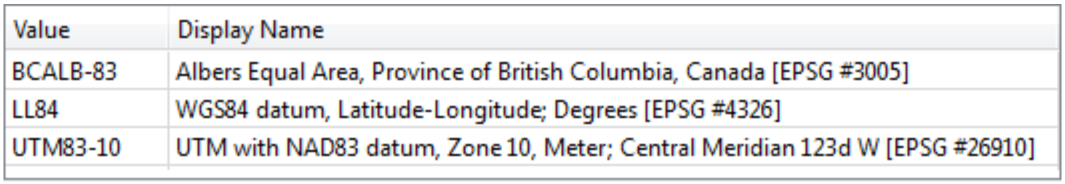
Search for and add the following coordinate systems:

LL84
BCALB-83
UTM83-10



Name	Description	Group	Datum
☑ BCALB-83	Albers Equal Area, Province of British Columbia, C...	CANADA	NAD83
□ EPSG:26910	UTM with NAD83 datum, Zone 10, Meter; Central ...	UTM83	NAD83
☑ LL84	WGS84 datum, Latitude-Longitude; Degrees [EPSG...	LL	WGS84
☑ UTM83-10	UTM with NAD83 datum, Zone 10, Meter; Central ...	UTM83	NAD83

Then click **OK** to close this dialog and return to the configuration dialog. It should look like this:



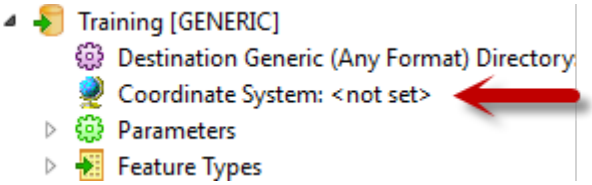
Value	Display Name
BCALB-83	Albers Equal Area, Province of British Columbia, Canada [EPSG #3005]
LL84	WGS84 datum, Latitude-Longitude; Degrees [EPSG #4326]
UTM83-10	UTM with NAD83 datum, Zone 10, Meter; Central Meridian 123d W [EPSG #26910]

Again, notice the Display Name column allows you to edit the display name to make it easier to recognize. Edit these in any way you think makes them easiest to understand, the click **OK**.

Choose any of coordinate systems as the default and click **OK** to create the parameter.

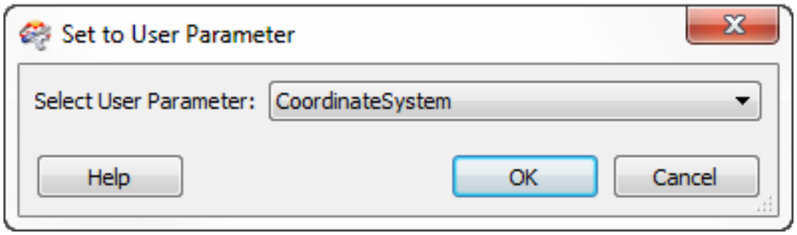
11. Link Custom Parameter

Now the coordinate system user parameter must be linked to the appropriate FME parameter.


Training [GENERIC]
Destination Generic (Any Format) Directory
Coordinate System: <not set>
Parameters
Feature Types

In the Navigator window, locate the Generic Writer and expand its list of parameters. Locate the Coordinate System parameter and right-click upon it. Choose Link to User Parameter:

Choose CordinateSystem from the list and click OK.



Select User Parameter: CoordinateSystem

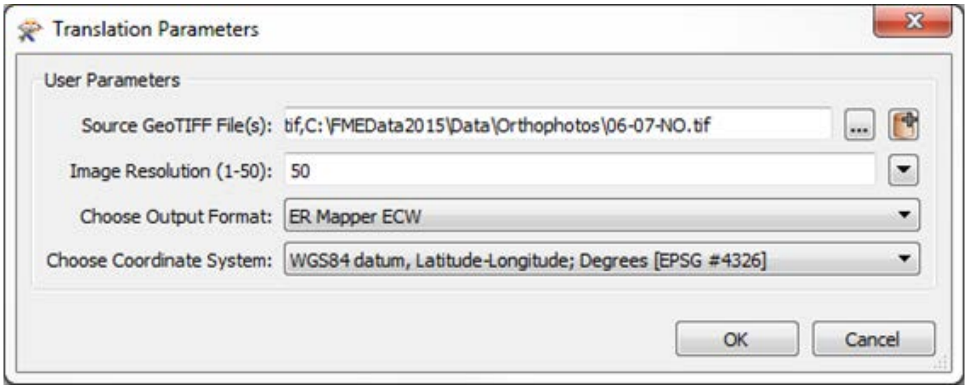
We now have user parameters set up for both coordinate system and format.

12. Run Workspace on FME Desktop

Save the workspace as Exercise3d.fmw.

Run the workspace using Run > Prompt and Run from the menubar.

The translation parameters dialog will open. Select a format and a coordinate system (and any existing parameters) and click **OK** to run the workspace.



Translation Parameters

User Parameters

Source GeoTIFF File(s): tif,C:\FMEData2015\Data\Orthophotos\06-07-NO.tif

Image Resolution (1-50): 50

Choose Output Format: ER Mapper ECW

Choose Coordinate System: WGS84 datum, Latitude-Longitude; Degrees [EPSG #4326]

Check the output folder (C:\FMEData2015\Output\Training) and inspect the datasets to ensure the translation is correct.

> Judge GIS says…
>
> *"You can use the Data Inspector to check the output however the JPEG file will show the coordinate system as Unknown.*

13. Publish to FME Server

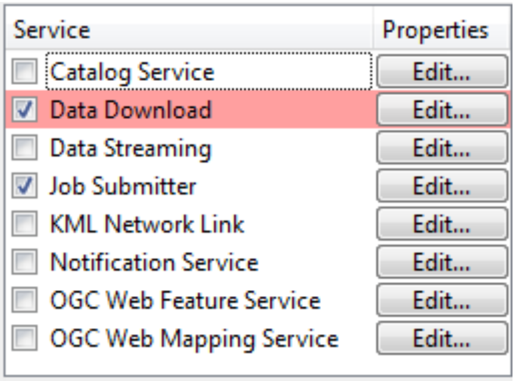
Use File > Publish to FME Server to publish the workspace.
Publish it to the Training repository and register it with the Data Download service.

Notice that in the Register Services window the Data Download Service is highlighted red.

This occurs because we deleted the original JPEG writer to which data was being downloaded from.

We need to specify a Writer for the Service so click Edit and select "Training[Generic]" as the dataset to include for the Data Download Service.

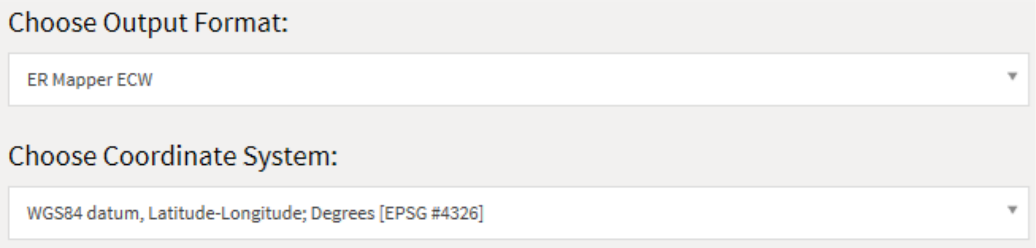
Click Publish to finish publishing the workspace.



Service	Properties
□ Catalog Service	Edit...
☑ Data Download	Edit...
□ Data Streaming	Edit...
☑ Job Submitter	Edit...
□ KML Network Link	Edit...
□ Notification Service	Edit...
□ OGC Web Feature Service	Edit...
□ OGC Web Mapping Service	Edit...

14. Run Workspace on FME Server
Changing once again to a user role, go to the Web User Interface and navigate to your workspace: Workspaces > Training > Exercise3d.fmw > Data Download

Notice your new parameters for format and coordinate system. Again select some image files from Browse Resources and this time choose one of the new formats and a coordinate system.



Choose Output Format:

ER Mapper ECW

Choose Coordinate System:

WGS84 datum, Latitude-Longitude; Degrees [EPSG #4326]

Run the workspace and then download your data to confirm the results.

Congratulations! Your self-serve Data Download service can now accommodate customers who work with different formats and coordinate systems.

Sister Intuitive says…

"You might have noticed that we could just publish the Writer dataset and coordinate system parameters directly, instead of creating them manually and then linking the two.

This will work, but by using a Choice with Alias parameter we can restrict the list of formats/coordinate systems, use an alias, and also get them as a drop-down list in FME Server.

Otherwise they would be text-only parameters that would be unlimited in what formats and coordinate systems were available, as long as you could guess what value was required."

Layer Selection and Handling

Layer (Feature Type) flexibility can be achieved through the Feature Types to Read parameter.

There is no single method by which users might wish to select data or features. Every organization implementing FME Server lets users select data in different ways.

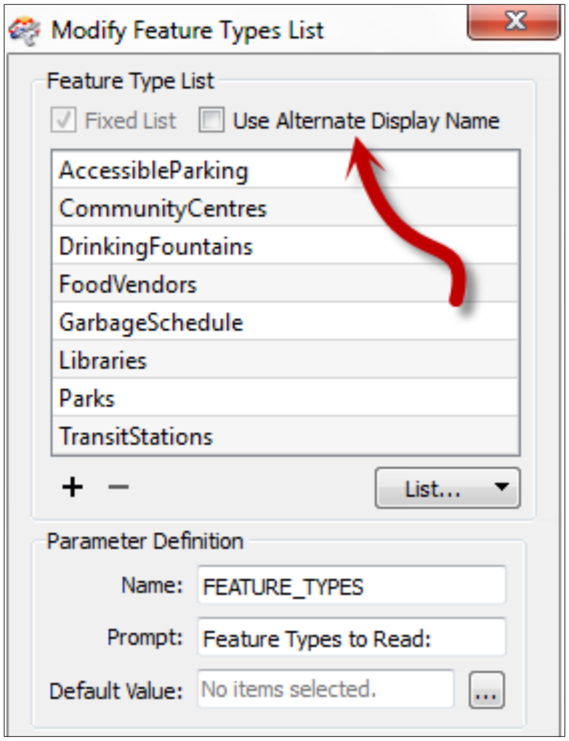
However, spatial data is often organized in layers (groups, classes, categories, feature types) and a common way that users will wish to choose data is on a layer-basis.

Selecting Layers

Allowing users to select data by layer is the simplest solution available in an FME Workspace, although in many situations users will wish to select data not just by layer, but by a mix of criteria.

Allowing choice by layer alone can be handled easily if the layers correspond to Feature Types in the source data.

Each reader in FME has a parameter called Feature Types to Read. When this parameter is published, the workspace will accept a list of layers to read from the user.



Modify Feature Types List

Feature Type List

☑ Fixed List ☐ Use Alternate Display Name

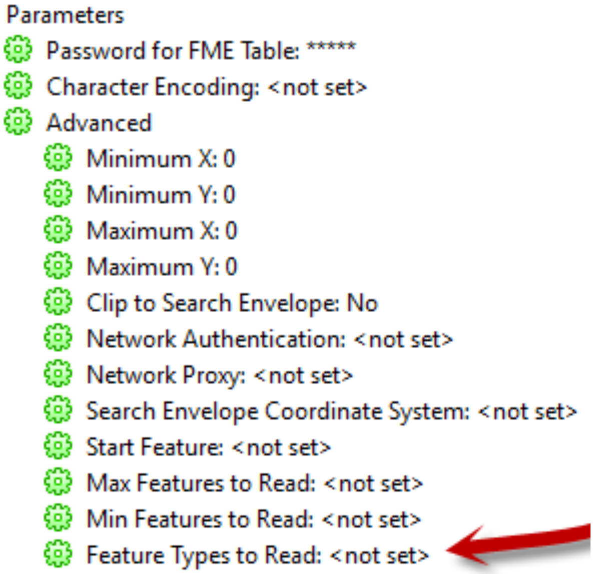
AccessibleParking
CommunityCentres
DrinkingFountains
FoodVendors
GarbageSchedule
Libraries
Parks
TransitStations

Parameter Definition

Name: FEATURE_TYPES

Prompt: Feature Types to Read:

Default Value: No items selected.



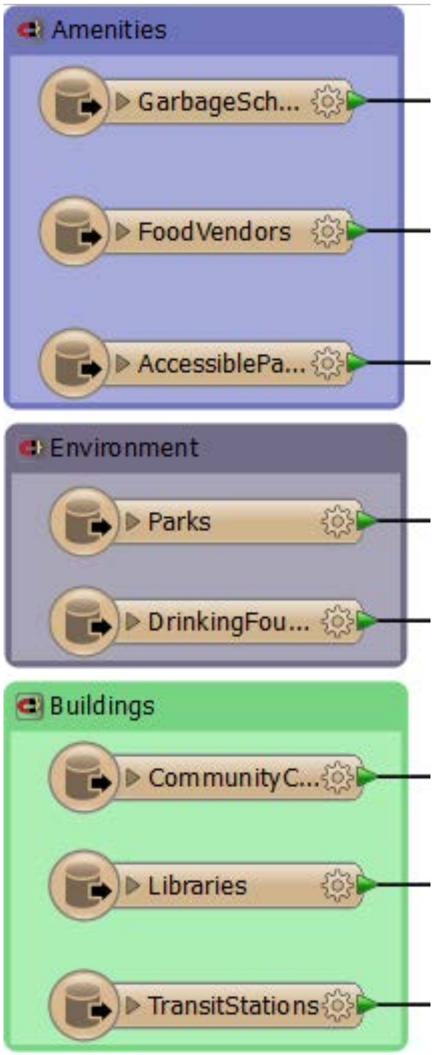
Parameters
- Password for FME Table: *****
- Character Encoding: <not set>
- Advanced
  - Minimum X: 0
  - Minimum Y: 0
  - Maximum X: 0
  - Maximum Y: 0
  - Clip to Search Envelope: No
  - Network Authentication: <not set>
  - Network Proxy: <not set>
  - Search Envelope Coordinate System: <not set>
  - Start Feature: <not set>
  - Max Features to Read: <not set>
  - Min Features to Read: <not set>
  - Feature Types to Read: <not set>

The publishing dialog for Feature Types to Read is different because FME can automatically scan the workspace and provide a list of available feature types.

Additionally, the Use Alternate Display Name allows the author to publish alternate names (for example Road Centrelines instead of RdCtrLns) in a similar way to the “Choice with Alias” parameter.

Grouped Layers

On some occasions the user does not need to be presented with a full list of layers. Instead they might need to see “groups” of layers.

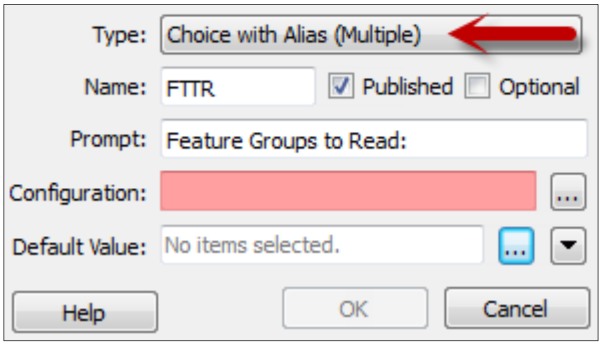


For example, what is presented to the end-user as “Water” features may actually be made up of multiple source feature types such as Lakes, Rivers, and Streams.

This type of scenario can be achieved using a parameter of type “Choice with Alias (Multiple)”.

Here the author has eight feature types, but wishes to present them to the end-user as three groups of layers.

The first step is to create a parameter of type “Choice with Alias (Multiple)”:



Type: Choice with Alias (Multiple)

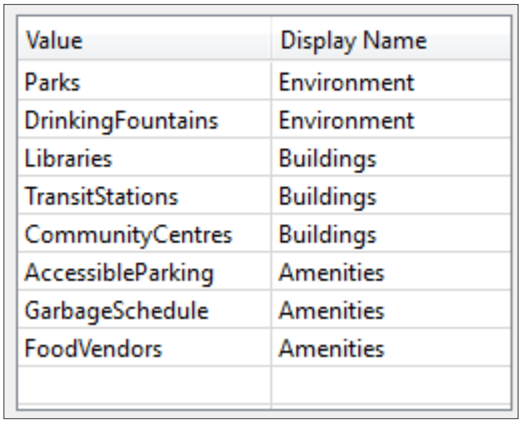
Name: FTTR ☑ Published ☐ Optional

Prompt: Feature Groups to Read:

Configuration:

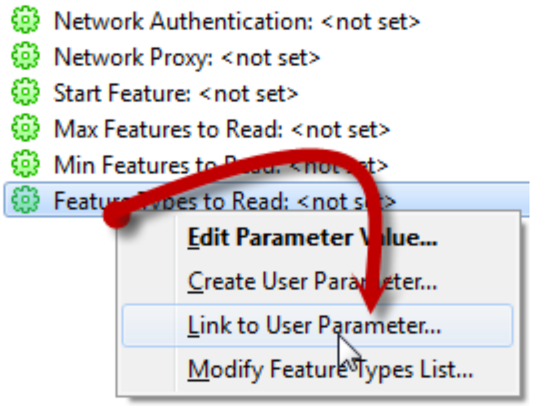
Default Value: No items selected.

In the parameter configuration, all feature types from the same group are given the same display name:



Value	Display Name
Parks	Environment
DrinkingFountains	Environment
Libraries	Buildings
TransitStations	Buildings
CommunityCentres	Buildings
AccessibleParking	Amenities
GarbageSchedule	Amenities
FoodVendors	Amenities

Finally, the Feature Types to Read parameter is linked to the newly created Choice with Alias parameter:



Network Authentication: <not set>
Network Proxy: <not set>
Start Feature: <not set>
Max Features to Read: <not set>
Min Features to Read: <not set>
Feature Types to Read: <not set>

Now when the workspace is run, the end-user is presented with a set of three options (Environment, Buildings, and Amenities) that reflect a series of different layers/feature types.

Exercise 3e: Data Download	
Scenario	Airphoto Data Vendor
Data	GeoTiff Orthophotos
Overall Goal	Improve the data download service
Demonstrates	The Feature Types to Read parameter for layer selection
Starting Workspace	C:\FMEData2015\Workspaces\ServerAuthoring\Exercise3e-Begin-DataDownload.fmw
Finished Workspace	C:\FMEData2015\Workspaces\ServerAuthoring\Exercise3e-Complete-DataDownload.fmw

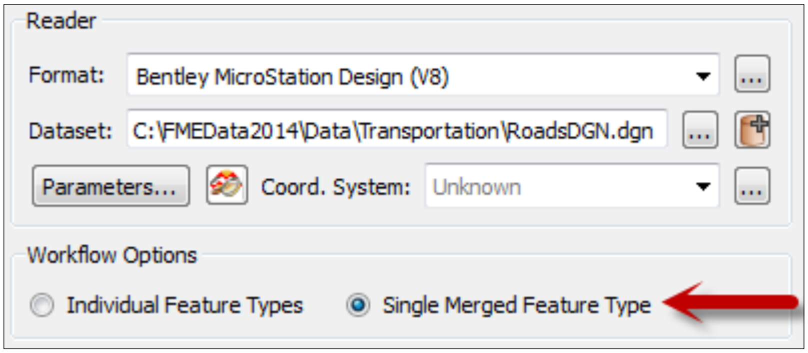
The task here is to update the current data download service to deliver a hybrid image of aerial photo and map data. We'll do this by adding a vector data Reader to the workspace and burning (stamping) it into the raster.

1. Start Workbench

Start FME Workbench (if necessary). Open the workspace from Exercise 3d (or the start workspace for this exercise).

Select Readers > Add Reader from the menubar and add a new Reader for some roads data with the following parameters:

Reader Format	Bentley MicroStation Design (V8)
Reader Dataset	C:\FMEData2015\Data\Transportation\RoadsDGN.dgn
Workflow Options	Single Merged Feature Type



Reader

Format: Bentley MicroStation Design (V8)

Dataset: C:\FMEData2014\Data\Transportation\RoadsDGN.dgn

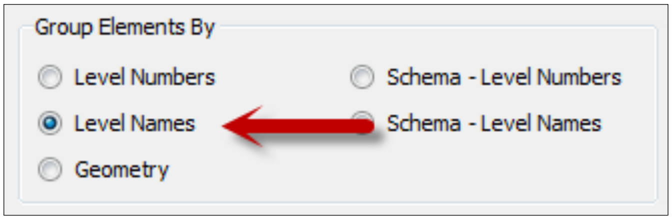
Parameters... Coord. System: Unknown

Workflow Options

○ Individual Feature Types ● Single Merged Feature Type

Click **Parameters** to open the parameters dialog.

In the dialog check Group Elements by Level Names and click **OK**. This will ensure data is added with the correct level names (not numbers).

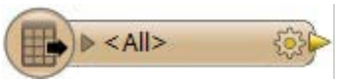


Group Elements By

○ Level Numbers ○ Schema - Level Numbers

● Level Names ○ Schema - Level Names

○ Geometry

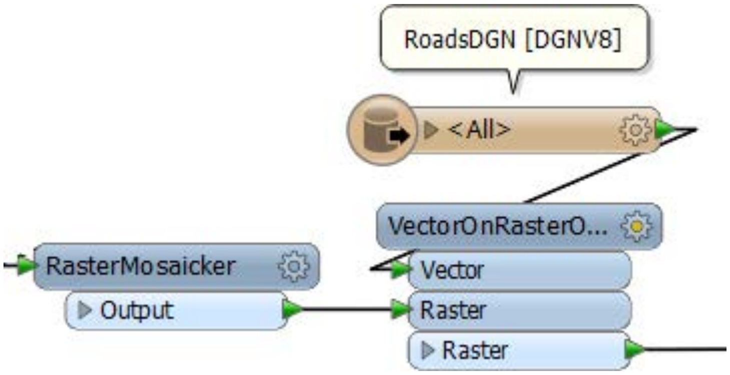


Click **OK** again to add the Reader. You will see a single source feature type which will allow any layer to pass through.

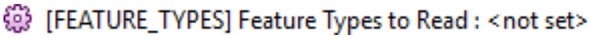
2. Merge Roads with Raster
Vector data can be added into a raster image by using the VectorOnRasterOverlayer transformer.

Add a VectorOnRasterOverlayer to the canvas.

Connect the RasterMosaicker Output port to the VectorOnRasterOverlayer Raster input port and connect the new MicroStation Reader feature type to the Vector input port. Connect the Raster output port to the AirphotoMosaic.

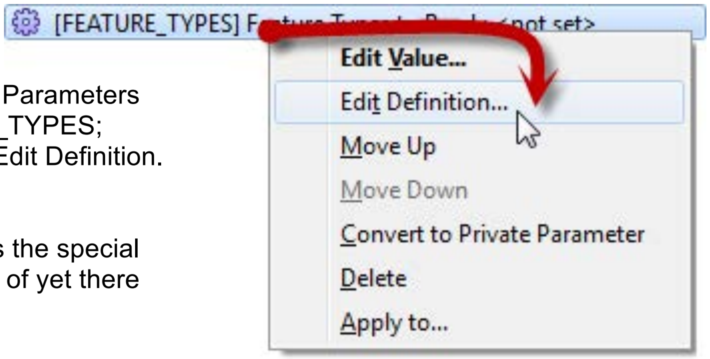


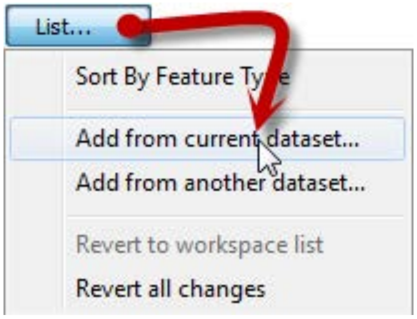
3. Modify the Feature Types to Read Parameter
When you added the MicroStation Reader, a user parameter was automatically created to allow users to choose which level to read from the file.


[FEATURE_TYPES] Feature Types to Read : <not set>

In the Navigator pane browse to User Parameters > Published Parameters > FEATURE_TYPES; right-click the parameter and choose Edit Definition.

This opens a dialog that demonstrates the special nature of this parameter. However, as of yet there is no content filled in.


[FEATURE_TYPES] Feature Types to Read : <not set>
Edit Value...
Edit Definition...
Move Up
Move Down
Convert to Private Parameter
Delete
Apply to...



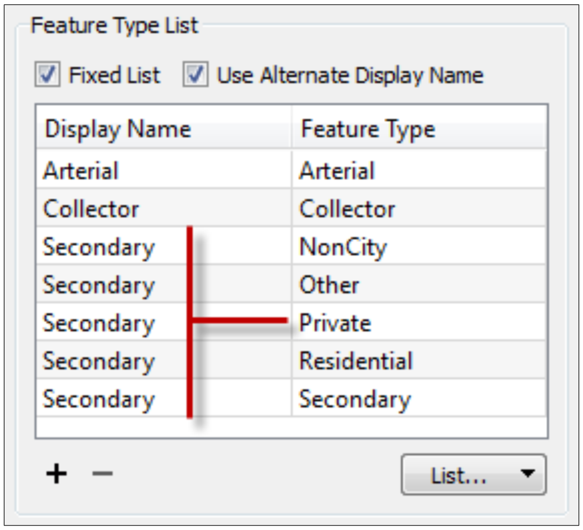
Check the parameter marked Fixed List. Then click the dropdown menu labelled List and choose Add from Current Dataset. Select all of the available feature types in the dataset and click **OK**.

This will basically create a list of layers for the end-user to select from.

4. Set up Layer Groups
As a final step in modifying this parameter, check the parameter labelled Use Alternate Display Name. This allows you to enter groups and aliases.

Edit the Display Names and give all the minor roads the name *Secondary* so that they become a group for the user.

Select Arterial as the default and click **OK** to close the dialog.

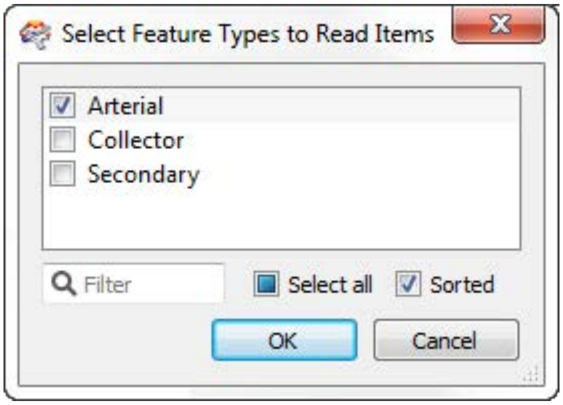


Feature Type List

☑ Fixed List ☑ Use Alternate Display Name

Display Name	Feature Type
Arterial	Arterial
Collector	Collector
Secondary	NonCity
Secondary	Other
Secondary	Private
Secondary	Residential
Secondary	Secondary

5. Run Workspace in FME Workbench
As always, confirm the parameters work as expected – and that the vector on raster overlay is working – by running the workspace using Run > Prompt and Run in FME Workbench.



Select Feature Types to Read Items

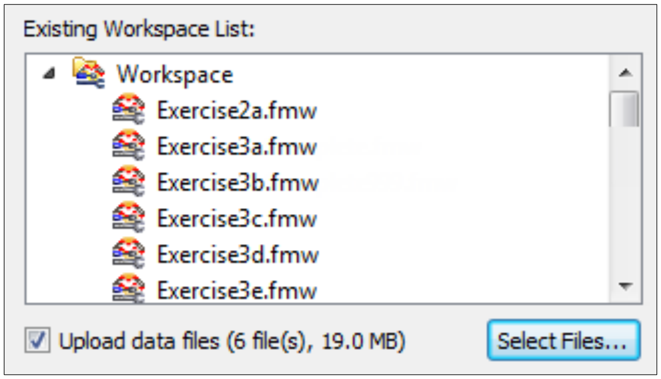
☑ Arterial
☐ Collector
☐ Secondary

When you check the Feature Types To Read parameter, you'll notice that the options are Arterial (the default), Collector, and Secondary – the groups that you just created.

6. Delete the MicroStation Source Dataset Parameter
Your users do not need to know where the MicroStation file is coming from, nor do they need to see a parameter for it. By deleting the parameter, we'll remove it from the user's view and ensure it gets uploaded to FME Server.

In the Navigator pane, browse to User Parameters > Published Parameters > SourceDataset_DGNV8, right-click the parameter and choose the option to delete.

7. Publish to FME Server
Save the workspace as Exercise3e.fmw.
Select File > Publish to FME Server from the menubar to start the publishing process.



Existing Workspace List:

- Workspace
  - Exercise2a.fmw
  - Exercise3a.fmw
  - Exercise3b.fmw
  - Exercise3c.fmw
  - Exercise3d.fmw
  - Exercise3e.fmw

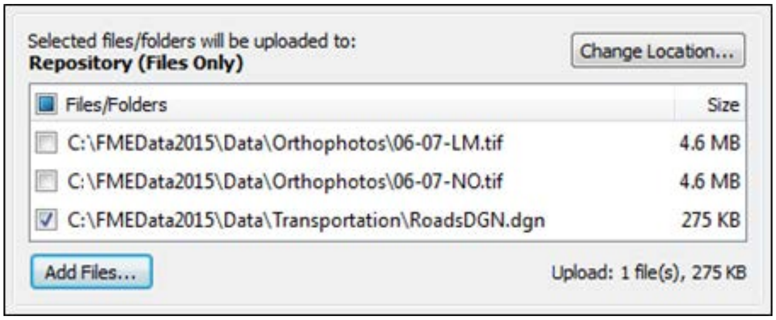
☑ Upload data files (6 file(s), 19.0 MB)

Fill in the connection details and credentials as usual and then click **Next** to continue.

In the next dialog select Training as the repository to publish to. However, do not yet click Next. Instead, click on the button labelled **Select Files**.

The Resources dialog shows what resources – such as source data – are being uploaded to FME Server along with the workspace.

Remember all GeoTIFF orthophotos are already uploaded, so uncheck those files from the list.

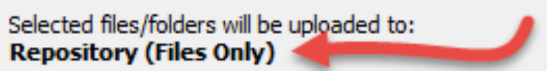


Selected files/folders will be uploaded to:
Repository (Files Only)

	Files/Folders	Size
□	C:\FMEData2015\Data\Orthophotos\06-07-LM.tif	4.6 MB
□	C:\FMEData2015\Data\Orthophotos\06-07-NO.tif	4.6 MB
☑	C:\FMEData2015\Data\Transportation\RoadsDGN.dgn	275 KB

Upload: 1 file(s), 275 KB

However, we do want to upload the MicroStation DGN file. In this case we'll upload it to a repository (available only to this workspace) rather than to a Resources folder where it would be open to everyone.



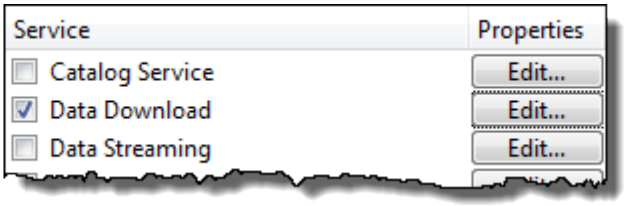
Selected files/folders will be uploaded to:
Repository (Files Only)

Check to see where the file will be uploaded. If the dialog already says repository, then all is OK. Else click the button to change the location and set it to repository.

When you click **OK**, a warning message will alert you to the fact that the GeoTIFF files are being referenced but not uploaded. That's OK – we know the files already exist on the server – so simply click **OK** to dismiss the dialog.

Then click **Next** to pass on to the final dialog of the publishing wizard.

8. Register with Data Download Service
In the Register Services dialog check Data Download if it is not already.



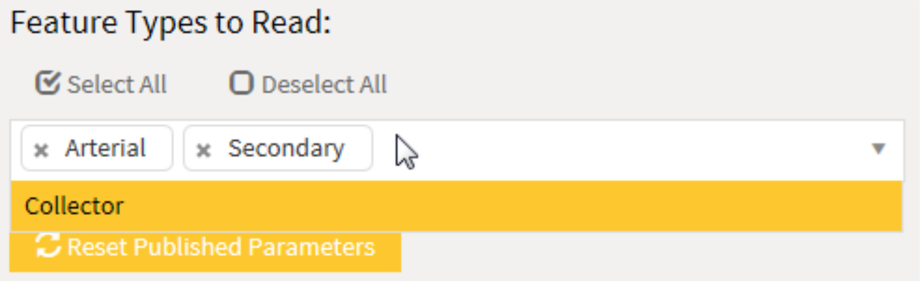
Service	Properties
□ Catalog Service	Edit...
☑ Data Download	Edit...
□ Data Streaming	Edit...

Click the Edit button and confirm that the Output Dataset for the service is set to your Generic Writer.

Click **OK** to close the dialog and **Publish** to complete the wizard and publish the workspace.

9. Run Workspace on FME Server
In the Web User Interface run your workspace using the Data Download Service as if you are a user, as we have been doing in the previous exercises.

Notice how the Feature Types to Read parameter is a selection parameter, like this:



Feature Types to Read:

☑ Select All   □ Deselect All

× Arterial   × Secondary

Collector

…letting the user select which roads layers (if any) they would like overlaid on the imagery.

Congratulations: Your Data Download service now lets the end-user retrieve data with a selection of required layers.

Geographic Selection

In most Self-Serve applications the user will need to be given a choice of geographic area of interest

Although most workspaces are designed around FME feature types, this may not be how the end-users see the data, or how they need to select it. An installation that provided selection only by layer (or even a group of layers) would be very limiting indeed.

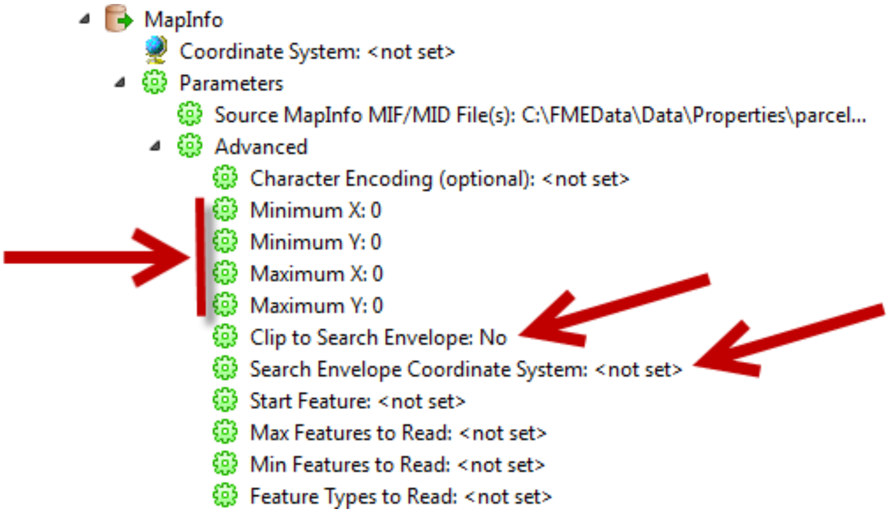
Besides layers, other methods of data selection revolve around spatially defining an area of interest. There are three obvious ways of doing so:

- A simple, rectangular bounding box
- An existing area such as a regional or municipal boundary
- An ad-hoc area (one drawn by the user on the fly)

Bounding Box

A bounding box can be defined in several ways, but the easiest is to use a set of parameters.

All FME readers have parameters to define the bounding box (envelope) of data that is being read. This is the most efficient method of selecting an area of interest because it means FME does not have to read the entire dataset first and then clip it to size; it can read only what data is necessary. It's particularly efficient if the source format has a spatial index, and it applies to both vector and raster datasets.



MapInfo
Coordinate System: <not set>
Parameters
Source MapInfo MIF/MID File(s): C:\FMEData\Data\Properties\parcel...
Advanced
Character Encoding (optional): <not set>
Minimum X: 0
Minimum Y: 0
Maximum X: 0
Maximum Y: 0
Clip to Search Envelope: No
Search Envelope Coordinate System: <not set>
Start Feature: <not set>
Max Features to Read: <not set>
Min Features to Read: <not set>
Feature Types to Read: <not set>

If these search envelope parameters are published, then the end-user is able to enter values that define the extent of the data he/she is interested in.

Besides the actual coordinates, there are two related parameters: “Clip to Search Envelope” and “Search Envelope Coordinate System”.

Clip to Search Envelope causes features that overlap the bounding box to be clipped and the outer part discarded. This will be the most common use for FME Server.

The Search Envelope Coordinate System means that the bounding box can have its own coordinate system, separate from the reader coordinate system. So the bounding box coordinates can be defined in a general coordinate system like Lat./Long, even if the data is UTM.

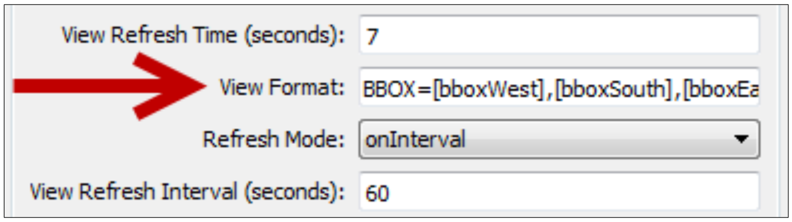
Web Mapping and Bounding Boxes

Bounding box parameters are particularly useful when a web mapping application (for example Google Earth) is requesting data. The application will provide the extents of the current map view and the workspace restricts itself to reading features within those extents.

This, of course, is done using published parameters. The parameters in the workspace need to be given specific names when published, as follows:

- bboxWest
- bboxEast
- bboxNorth
- bboxSouth

If given these names, both OGC services and the KML Network Link service can make use of these parameters. In fact, if you have used an OGC web service, or a KML Network Link service (as below), you might have noticed how these are used to pass information back to the workspace:



View Refresh Time (seconds): 7

View Format: BBOX=[bboxWest],[bboxSouth],[bboxEa

Refresh Mode: onInterval

View Refresh Interval (seconds): 60

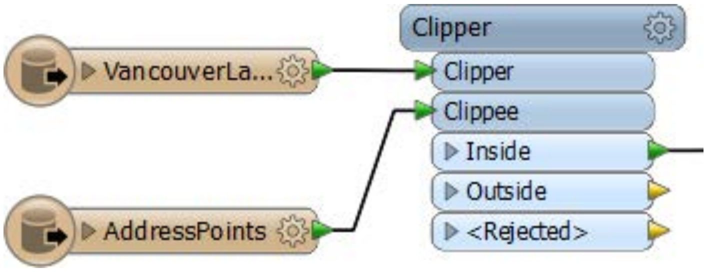
Ms. Analyst says…

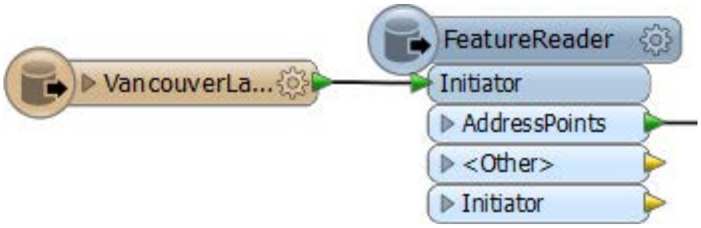
“Using reader parameters like this ensures the data is reduced in size at the earliest possible opportunity and so gives a better performance.”

Irregular Areas of Interest
An irregularly shaped area cannot be used as a clip boundary in a Reader's parameters. However, it can be used to clip data using either a Clipper transformer or a FeatureReader transformer.

The Clipper has an input port for the area of interest (Clipper) and one for the main dataset (Clippee).

For example, here the portal will deliver only addresses that fall inside the boundary of the city of Vancouver:





Here is the same result but using a FeatureReader transformer.

The FeatureReader reads the main dataset directly, executing out one of any number of spatial filters.

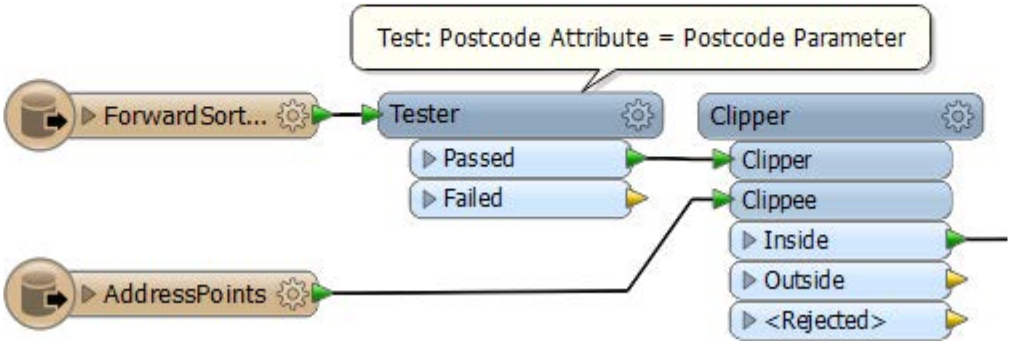
In general the FeatureReader is a better choice than a Clipper when using a database with a spatial index, or when the spatial predicate is other than a simple CONTAINS function.

Existing Area of Interest
The main issue with an irregular area of interest is how to define that area. In some cases, the feature is already defined as a known shape, such as an administrative or municipal boundary, and already exists in a spatial dataset.

In that case the area can simply be read from the dataset and routed into the appropriate input port of the Clipper/FeatureReader transformer.

If the source contains a number of features, then a published parameter can be used for the end-user to identify the area they want and for FME to filter other features out.

For example, this Self-Serve workspace uses a Clipper transformer to deliver only addresses that fall within a chosen postal code. The postal code is selected by the user through a published parameter. The Tester transformer filters out postal code boundaries that don't match the user's selection.

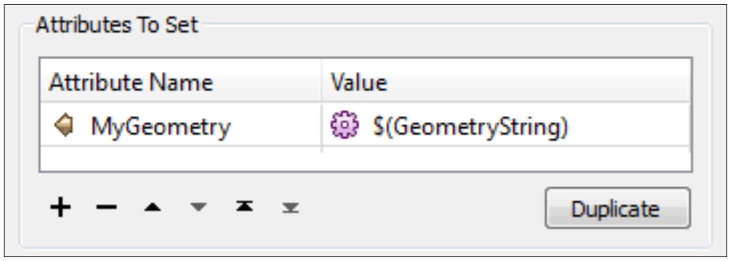


Ad Hoc Area of Interest

In the case that the area of interest is not a previously known shape – for example it is one defined entirely by the end user – then a different technique is needed.

This scenario needs a published parameter through which a geometry string can be provided. The geometry could be, for example, either WKT (Well-Known Text) or XML.

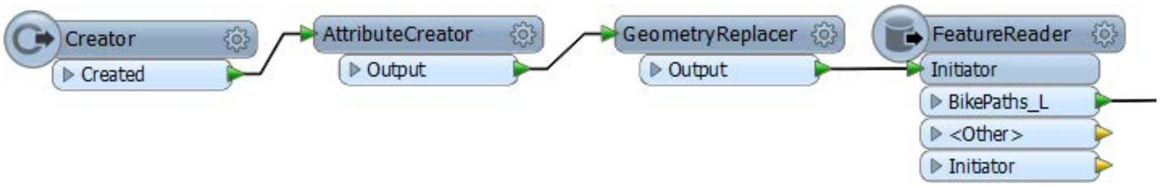
One simple method would be to use an AttributeCreator transformer and publish the Value part.



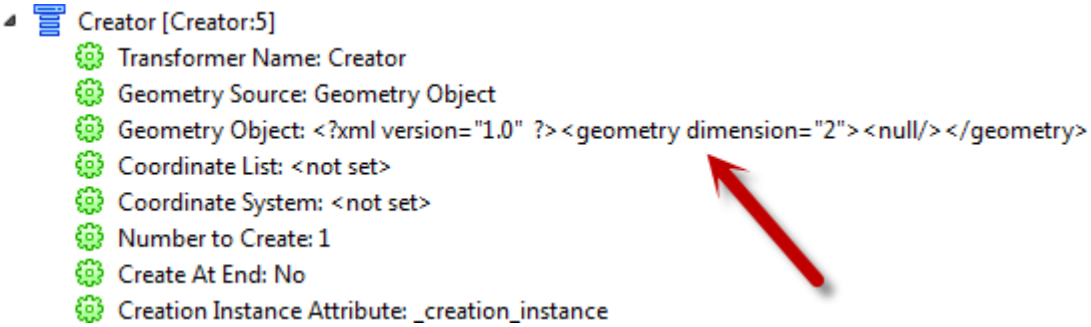
Attributes To Set

Attribute Name	Value
MyGeometry	$(GeometryString)

This could be used in a sequence like this, where the translation is triggered with a Creator, the geometry string retrieved with the AttributeCreator, the attribute contents converted into true geometry with a GeometryReplacer, and then the real data read with a FeatureReader:



To use XML you could also just publish the geometry parameter in a Creator transformer.



Creator [Creator:5]
Transformer Name: Creator
Geometry Source: Geometry Object
Geometry Object: <?xml version="1.0" ?><geometry dimension="2"><null/></geometry>
Coordinate List: <not set>
Coordinate System: <not set>
Number to Create: 1
Create At End: No
Creation Instance Attribute: _creation_instance

With these methods, a web mapping interface that allows the user to define their own custom area of interest can pass it to the FME workspace to be used as the area of interest in a Clipper or FeatureReader.

Exercise 3f: Data Download Area of Interest	
Scenario	Airphoto Data Vendor
Data	GeoTiff Orthophotos
Overall Goal	Improve the Data Download Service
Demonstrates	Selection of data via geographic area
Starting Workspace	C:\FMEData2015\Workspaces\ServerAuthoring\Exercise3f-Begin-DataDownload.fmw
Finished Workspace	C:\FMEData2015\Workspaces\ServerAuthoring\Exercise3f-Complete-DataDownload.fmw

The task here is to let the end user decide what area of data is to be delivered. For this exercise we'll let them choose which neighborhood of the city they wish the data for.

1. Start Workbench

Start FME Workbench (if necessary). Open the workspace from Exercise 3e (or the start workspace for this exercise).

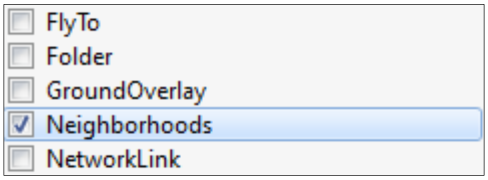
2. Add Reader

To filter by neighborhood we need to read a dataset of neighborhood boundaries.

Select Reader > Add Reader from the menubar and add a new Reader with the following parameters:

Reader Format	Google Earth KML
Reader Dataset	C:\FMEData2015\Data\Boundaries\VancouverNeighborhoods.kml
Workflow Options	Individual Feature Types

When prompted to select feature types to add to the workspace, simply deselect everything except the Neighborhoods layer



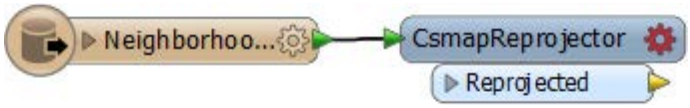
☐ FlyTo
☐ Folder
☐ GroundOverlay
☑ Neighborhoods
☐ NetworkLink

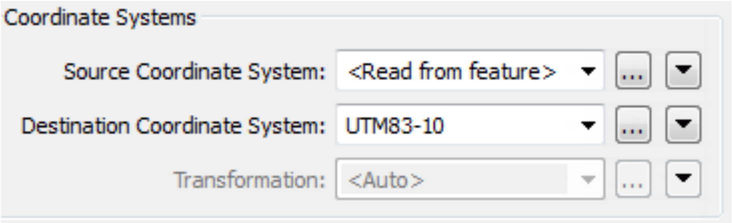
3. Add Reprojector

Being a KML dataset, this data is in a latitude-longitude coordinate system and must therefore be reprojected to match the UTM coordinate system of the raster data.

Add a CsmapReprojector transformer to the workspace.

Connect it to the Neighborhood Reader feature type:





Coordinate Systems

Source Coordinate System: <Read from feature>

Destination Coordinate System: UTM83-10

Transformation: <Auto>

Open the transformer's parameters dialog.

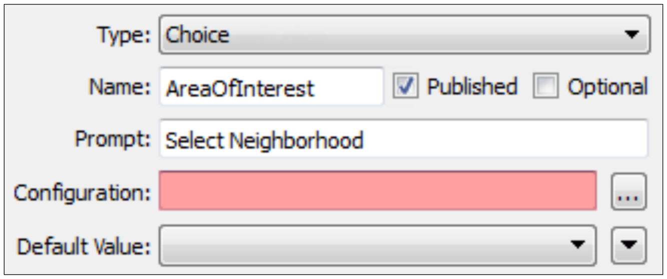
Leave the Source Coordinate System as <Read from feature> and set the Destination to UTM83-10.

4. Add Parameter

Now let's add a custom parameter for the user to select which neighborhood to download.

Right-click on Published Parameters in the Navigator window and choose Add Parameter.
Add a parameter with the following settings:

Type	Choice
Name	AreaOfInterest
Prompt	Select Neighborhood
Published	Yes
Optional	No



Type: Choice

Name: AreaOfInterest ☑ Published □ Optional

Prompt: Select Neighborhood

Configuration:

Default Value:

Start to configure the parameter by clicking the […] button for the Configuration parameter.

Instead of entering each neighborhood name separately, we'll import a list of them from the KML dataset. Start this process by clicking the Import button.

When prompted enter:

Reader Format	Google Earth KML
Reader Dataset	C:\FMEData2015\Data\Boundaries\VancouverNeighborhoods.kml

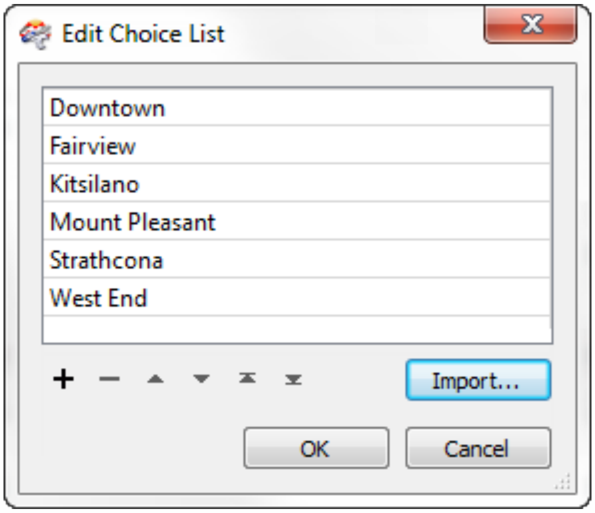
…and click **Next**. Select Neighborhoods as the Feature Type and click **Next**. Select NeighborhoodName as the attribute to import from and click **Next**.

The dataset will now be scanned for valid values. When it is complete click **Import**.

We now have a list of available neighborhoods to choose from.

Notice that the names are already complete so we don't need to use any sort of alias.

Click OK and OK again to close the parameter dialogs.

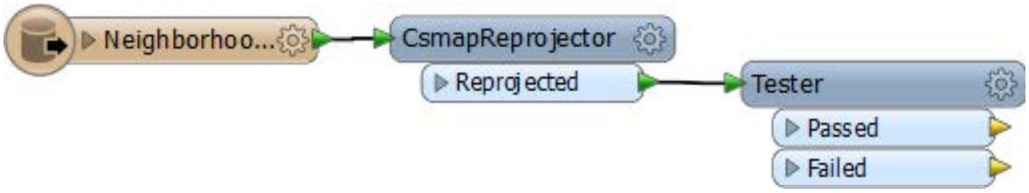


Edit Choice List

Downtown
Fairview
Kitsilano
Mount Pleasant
Strathcona
West End

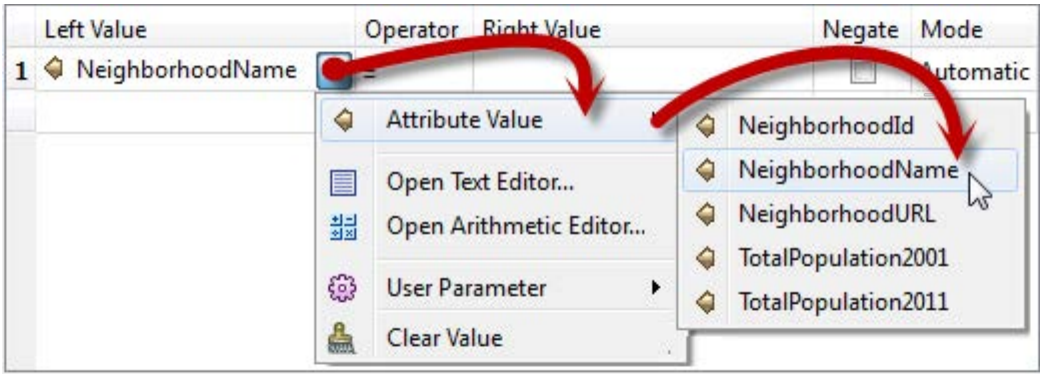
5. Add Tester
Now let's add a Tester transformer. This will filter out the neighborhoods that weren't selected.

Add a Tester and connect it to the CsmapReprojector transformer:



Open the Tester parameters dialog by clicking the red cog wheel button.

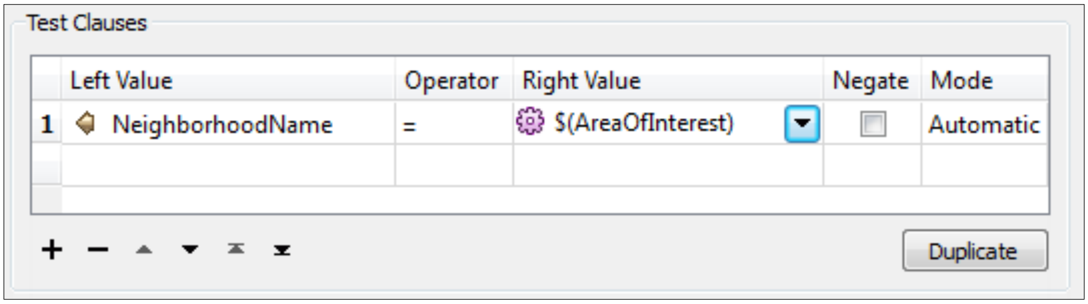
For the "Left Value" of the test, double-click in the field, click the drop-down arrow, and select Attribute Value > NeighborhoodName.



	Left Value	Operator	Right Value	Negate	Mode
1	NeighborhoodName	=		□	Automatic

Set the Tester operator to equals (=).

For the Right Value double-click in the field, click the drop-down arrow, then select User Parameter > AreaOfInterest.



Test Clauses

	Left Value	Operator	Right Value	Negate	Mode
1	NeighborhoodName	=	$(AreaOfInterest)	□	Automatic

Click **OK** to close the dialog.

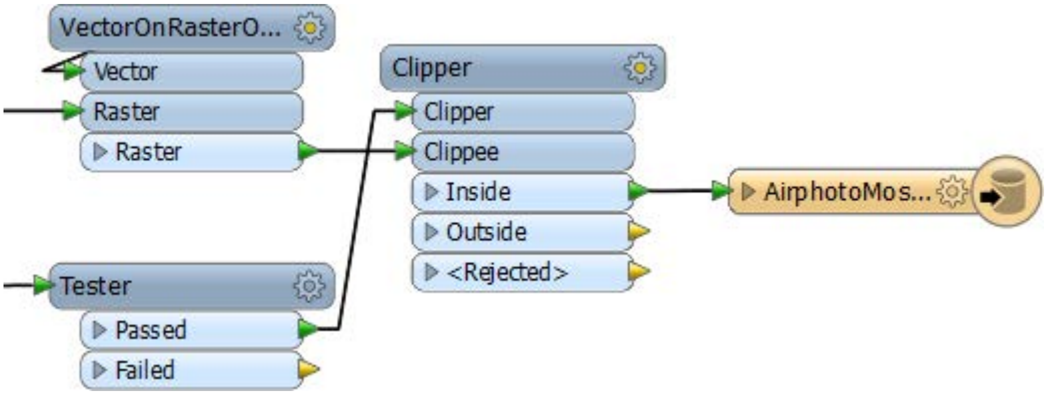
6. Add Clipper

We now have a test to check where the neighborhood is the one chosen by the user, but we haven't yet done anything with the result. We must use the chosen neighborhood as a boundary to clip the raster output.

Add a Clipper transformer.

Connect it up by connecting the Tester Passed port to the Clipper Clipper input port.
Connect the VectorOnRasterOverlayer Raster port to the Clipper Clippee input port.

The Clipper:Inside port can be connected to the output feature type and any existing connector deleted. The workspace will now look like this:



7. Delete Parameter

As with the Roads data, the end-user does not need to see a parameter for the neighborhoods source, so locate the source Google Earth KML parameter and delete it.

8. Test Run Workspace

Save the workspace as Exercise3f.fmw
Run the workspace using Run > Prompt and Run in FME Workbench.

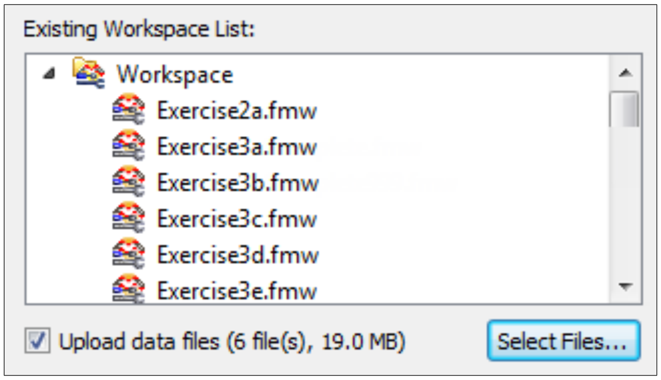
For the source GeoTIFF files you'll want to select all of the files to ensure the neighborhood is covered fully.

Then select a neighborhood to view the data; West End or Downtown are good to use.

9. Publish to FME Server
Save the workspace as Exercise3f.fmw.
Select File > Publish to FME Server from the menubar to start the publishing process.

Fill in the connection details and credentials as usual and then click **Next** to continue.

In the next dialog select Training as the repository to publish to and, as before, click on the button labelled **Select Files**.

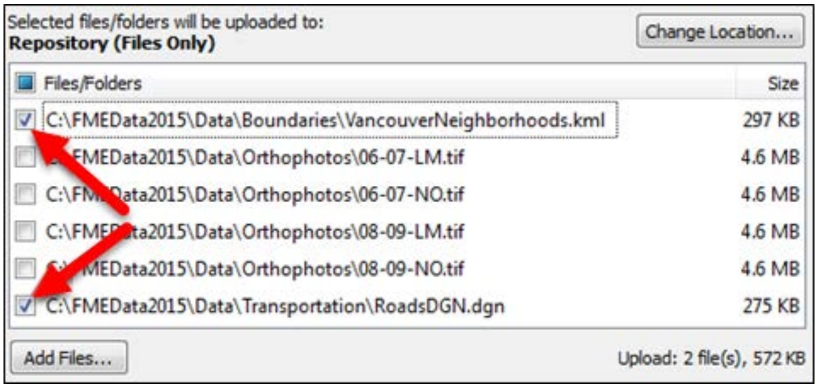


Existing Workspace List:

- Workspace
  - Exercise2a.fmw
  - Exercise3a.fmw
  - Exercise3b.fmw
  - Exercise3c.fmw
  - Exercise3d.fmw
  - Exercise3e.fmw

☑ Upload data files (6 file(s), 19.0 MB)

Ensure that both the MicroStation roads dataset, and the KML neighborhoods dataset, are being uploaded to the repository, but that the Orthophotos are not.



Selected files/folders will be uploaded to:
Repository (Files Only)

Files/Folders	Size
☑ C:\FMEData2015\Data\Boundaries\VancouverNeighborhoods.kml	297 KB
□ C:\FMEData2015\Data\Orthophotos\06-07-LM.tif	4.6 MB
□ C:\FMEData2015\Data\Orthophotos\06-07-NO.tif	4.6 MB
□ C:\FMEData2015\Data\Orthophotos\08-09-LM.tif	4.6 MB
□ C:\FMEData2015\Data\Orthophotos\08-09-NO.tif	4.6 MB
☑ C:\FMEData2015\Data\Transportation\RoadsDGN.dgn	275 KB

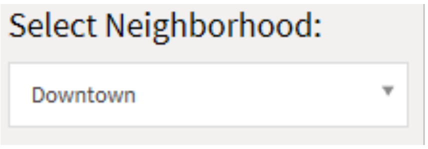
Upload: 2 file(s), 572 KB

Again, a warning message will alert you to the fact that the GeoTIFF files are being referenced but not uploaded. Again, it's fine to simply click **OK** to dismiss the dialog.

Register the workspace with the Data Download service again.

10. Run Workspace on FME Server
In the Web User Interface, run your workspace using the Data Download Service as if you are a user, as we have been doing in the previous exercises.

Notice the neighborhood is available as a dropdown list.



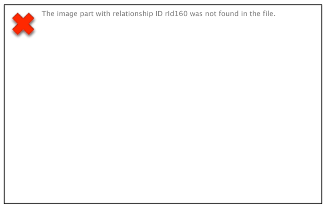
Select Neighborhood:

Downtown ▾

Pick a neighborhood – and enough source files to cover it – and run the workspace.

Congratulations: Your Data Download service now lets the end-user retrieve data within a set geographic area.

Other Self-Serve Services


The image part with relationship ID rId160 was not found in the file.

Data Download is not the only Self-Serve Service available.

Data Streaming

Whereas the Job Submitter service writes data, and the Data Download service returns a link to the data, a Data Streaming service returns a file of the data itself, streamed back to the client.

For example, the URL for the service can be posted into a web browser, and the data will be automatically downloaded and opened in the application associated with that file type.

Alternatively, the URL can be used directly as the source for a client application. When the client actively downloads the contents on a regular basis – as a GeoRSS reader would – then you have a feed, which is significantly different to a regular data download service.

Data Streaming is a slight misnomer in that a data streaming service does not supply a continuous stream of data; it merely provides a snapshot of the data at a particular point in time.

What Formats can be Streamed?

You can use any workspace with the data streaming service, provided it writes data in a format that is file-based or folder-based.

If an output dataset is comprised of more than one file, the service automatically creates a compressed (zip) folder out of the data. For example, AutoCAD DWG format could be streamed, whereas ESRI Shape would be returned in a zipped file.

The most popular formats to stream are those that have a suitable client to read the feed. Some of the main formats that are output using the data streaming services include:

- RSS
- GeoRSS
- GeoJSON
- KML
- HTML
- JSON

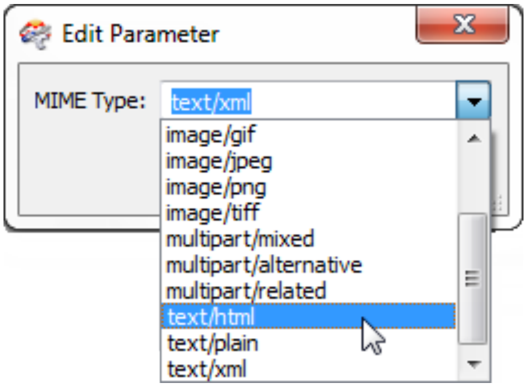
MIME Types in Workspaces

A MIME header is a component of a file or e-mail message that is capable of indicating the content type of the file; for example, Content-Type: text/plain indicates a simple text file.

The application that opens a streamed file will depend on the MIME type and file association on the client's system. MIME type is provided in FME by the first Writer added to the workspace.

The only Writers in FME with a MIME type setting are the TextLine (text file) and Data File Writers.

This parameter only appears when the Writer is added, and (for example) it allows the workspace author to set a MIME type of HTML and therefore stream data directly into a web browser.


Edit Parameter

MIME Type: text/xml

KML Network Link

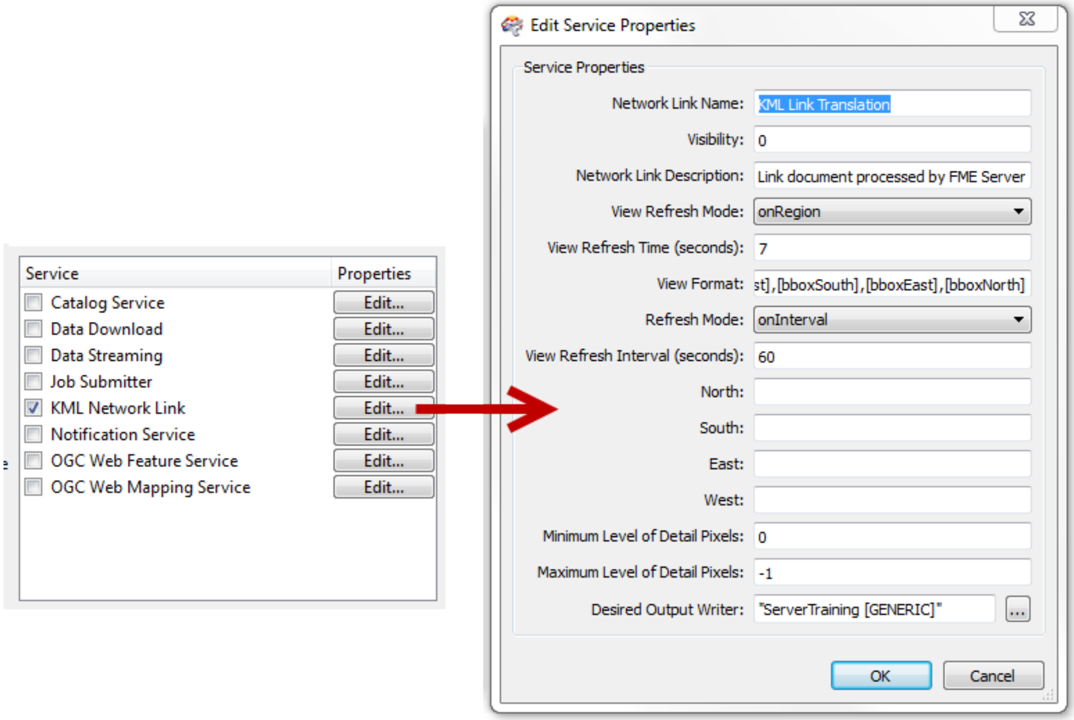
A KML Network Link service is when the output from an FME Server translation is a simple KMZ (zipped KML) file that contains solely a network link to reference the workspace on the server.

Whenever a KML browser (such as Google Earth) opens this dataset it follows the link, triggering FME Server to run the translation and return the true output data.

Because Google Earth permits a refresh rate for network links, the translation can be re-run at a user-defined interval. This way the results are always as up-to-date as the chosen interval.

Of course, in this scenario the output is never written to a permanent dataset; the resulting data is simply streamed to the browser, which writes it to a cache.

As with other services, a workspace becomes available for use as a KML Network Link service when the author checks the KML Network Link checkbox in the FME Server Publishing Wizard. The edit button provides a whole series of different parameters that control the refresh rate when the data is opened in Google Earth.



Service	Properties
□ Catalog Service	Edit...
□ Data Download	Edit...
□ Data Streaming	Edit...
□ Job Submitter	Edit...
☑ KML Network Link	Edit...
□ Notification Service	Edit...
□ OGC Web Feature Service	Edit...
□ OGC Web Mapping Service	Edit...

Edit Service Properties

Service Properties

Network Link Name: KML Link Translation

Visibility: 0

Network Link Description: Link document processed by FME Server

View Refresh Mode: onRegion

View Refresh Time (seconds): 7

View Format: st],[bboxSouth],[bboxEast],[bboxNorth]

Refresh Mode: onInterval

View Refresh Interval (seconds): 60

North:

South:

East:

West:

Minimum Level of Detail Pixels: 0

Maximum Level of Detail Pixels: -1

Desired Output Writer: "ServerTraining [GENERIC]"

Exercise 3g: Data Streaming	
Scenario	Airphoto Data Vendor
Data	GeoTiff Orthophotos
Overall Goal	Create a Data Streaming Service
Demonstrates	Data Streaming
Starting Workspace	C:\FMEData2015\Workspaces\ServerAuthoring\Exercise3g-Begin-DataStreaming.fmw
Finished Workspace	C:\FMEData2015\Workspaces\ServerAuthoring\Exercise3g-Complete-DataStreaming.fmw

Any exercises we've completed up until now could also be used with a simple Data Streaming service, just as easily as a Data Download service, and with very little editing.

This would allow customers to preview their downloads in a browser.

1. Open Starting Workspace

Start Workbench and open the starting workspace. Alternatively open the workspace from any of the previous exercises.

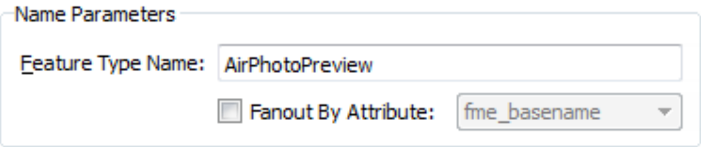
2. Add Writer

For data streaming it's important that we have a workspace with a JPEG writer, not Generic. Even if the Generic writer writes a JPEG file, it can't be streamed because the MIME type is unknown.

So, use Writers > Add Writer from the menubar. When prompted add a Writer as follows:

Writer Format	JPEG (Joint Photographic Experts Group)
Writer Dataset	C:\FMEData2015\Output\Training

Ensure Copy from Reader file definition is selected. Click OK to close the dialog. Open the Feature Type Properties dialog, change the Raster File Name to AirphotoPreview, and make sure that the fanout option is unchecked.

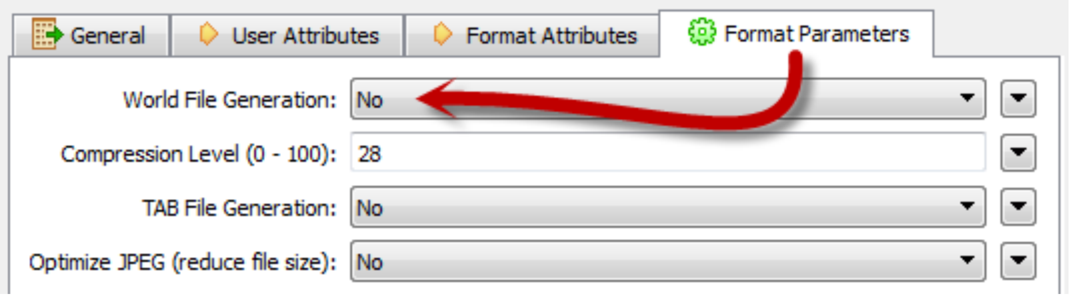


Name Parameters

Feature Type Name: AirPhotoPreview

□ Fanout By Attribute: fme_basename

Click the Format Parameters tab and check the parameter for World File Generation. It needs to be set to "No". If it is not "No" then multiple files will be output and streaming will fail.



World File Generation: No

Compression Level (0 - 100): 28

TAB File Generation: No

Optimize JPEG (reduce file size): No

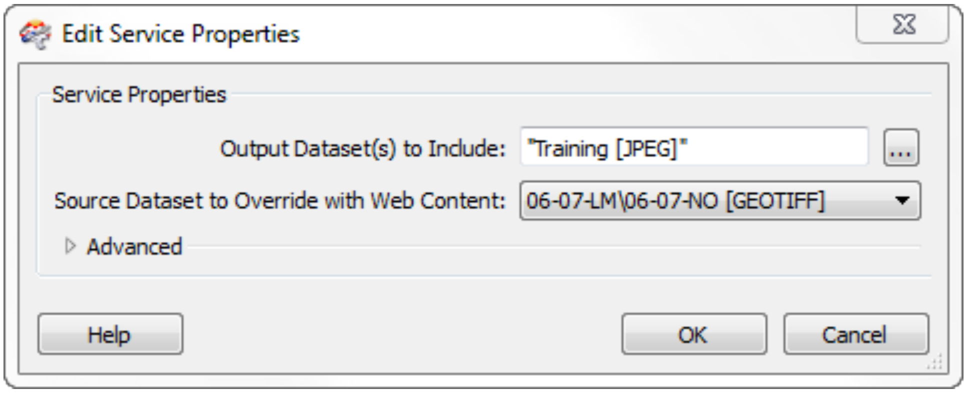
3. Publish the Workspace to the Data Streaming Service

Connect the newly created Feature Type to the RasterMosaicker Output port. Then select File > Publish to FME Server from the menubar to publish the workspace.

Save the workspace (as Exercise3g.fmw) and publish it to the Training repository again; but this time, register it with both the Data Download *and* Data Streaming services.

For the Data Download service, click the Edit button to make sure the Output Dataset associated with the service is the Generic Writer.

For the Data Streaming service, click the Edit button to make sure the Output Dataset associated with the service is the JPEG Writer, and *only* the JPEG writer.



Edit Service Properties

Service Properties

Output Dataset(s) to Include: "Training [JPEG]"

Source Dataset to Override with Web Content: 06-07-LM\06-07-NO [GEOTIFF]

4. Run the Workspace in the Data Streaming Service

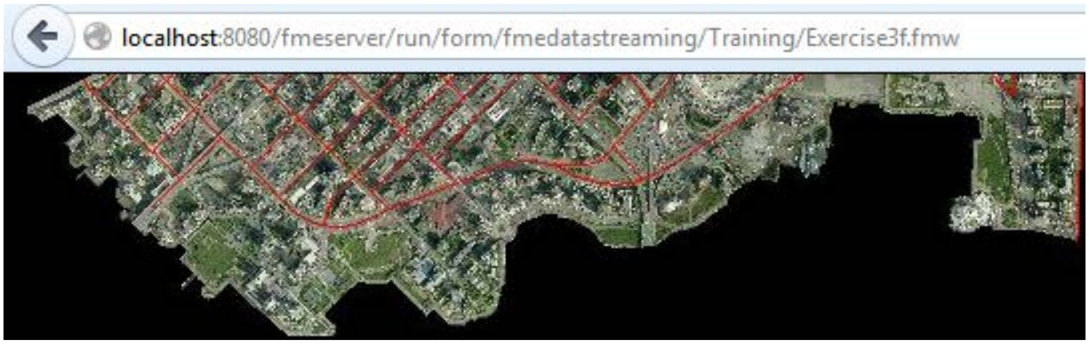
Acting as a user now, go to the Web User Interface and navigate to your workspace: Workspaces > Training > Exercise3g.fmw > Data Streaming

Enter some parameters. Data Download specific parameters like destination coordinate system and output format will not apply when streaming.

Click Run Workspace on the bottom of the form.

After a short time, the resulting image will appear on the webpage. Your browser automatically displays the image because FME Server provided the MIME type image/jpeg in the headers of the page it returned.

Hopefully you see how this service could be incorporated into your self-server data sharing solution, so users can preview the data they want to download at lower resolutions first.



localhost:8080/fmeserver/run/form/fmedatastreaming/Training/Exercise3f.fmw

Congratulations! You have now created a Data Streaming service for your customers.

Troubleshooting for Administrators

This section shows a few basic troubleshooting techniques in case of emergency

Data Download: No Output

If no zip file is output at the end of a Data Download process then the following suggestions may be of help.

- Run the workspace on FME Desktop to confirm that it does actually write some data. If there is no output data then no zip file will be delivered.

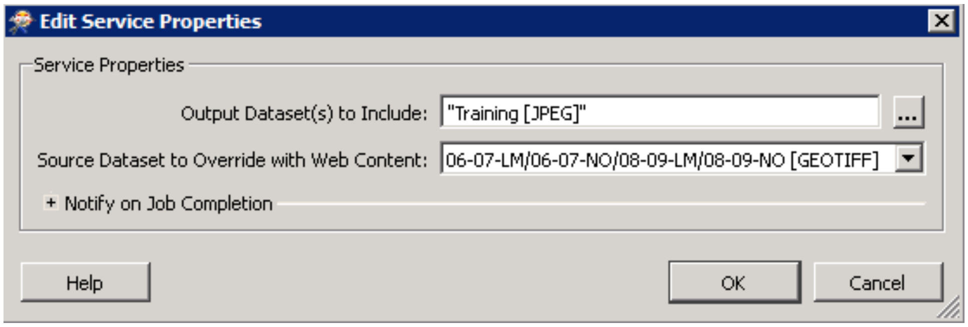
- Check that the FME Server has access to the source data. If the source data is not available to FME Server then there will be no output.

- On FME Server check the log file using Jobs > Completed to confirm that the process did actually try to write some data. If not an error message may help to indicate why.

Data Download: Empty Output

If an empty zip file is output at the end of a Data Download process then the following may be of help.

- Ensure that when you publish the workspace to FME Server you check the Data Download parameters and choose an output dataset to be associated with the Service. If you don't, then FME has no way to determine which Writer will provide the output.



Edit Service Properties

Service Properties

Output Dataset(s) to Include: "Training [JPEG]"

Source Dataset to Override with Web Content: 06-07-LM/06-07-NO/08-09-LM/08-09-NO [GEOTIFF]

+ Notify on Job Completion

Data Streaming: Zipped Output

If the Data Streaming service does not provide a dataset, but a zip file, then the following may be of help.

- If the output format is one composed of a number of files – for example Esri Shape – then the Data Streaming service will create a zip of the output.

  Often raster Writers will write an additional world file, and vector Writers a prj file, in order to preserve coordinate system information, and this can unexpectedly cause this issue. In most cases wld/prj file creation can be turned off with either Writer or Feature Type parameters.

Module Review

Module Review

What you should have learned from the module

This module introduced you to FME Server's Self-Serve Services and related functionality in FME Workbench

What You Should Have Learned from this Module

The following are key points to be learned from this module.

Theory

- Services handle communication between FME Server and its clients. Each workspace published to FME Server can be registered with any number of services.
- The Data Download service overrides the destination dataset parameter and presents data to the end-user as a link to a zip file
- Parameters control the different actions of a Data Download system, including the output coordinate system.
- The Generic Writer is a writer whose format is only defined at run time. This can be provided by the user through a published parameter.
- Selection of source data by layer is achieved using the Feature Types to Read parameter
- Selection of source data by area can be achieved using Bounding Box parameters, or a transformer such as the Clipper or FeatureReader
- Other FME Services allow data to be streamed live into a suitable application

FME Skills

- The ability to publish a workspace and register it to the FME Server Data Download service
- The ability to create and use published parameters
- The ability to give the end-user control over format, coordinate system, layers, and area of interest.
- The ability to use Data Streaming and KML Network Link services

Real-Time with FME Server

Real-Time with FME Server

Real-Time with FME Server

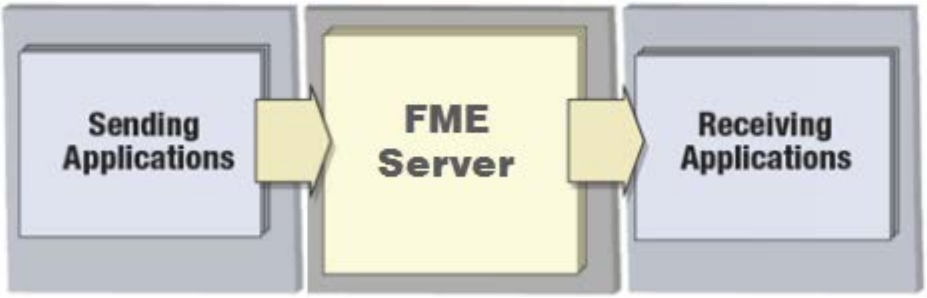
What is Real-Time?
Notifications
Message Streams

FME Server Basic Training 2015

Real-Time systems are those that act on events as they happen, and send information as it becomes available.

What is Real-Time?

In real-time FME Server is either set up to receive incoming information in the form of notifications, messages, and alerts, or to send information in a similar format.



Real-time is often implemented to monitor sensor networks, run event-driven processing, and/or to send alerts to mobile devices such as cell phones.

Therefore real-time is usually related to small amounts of information, rather than large datasets, which can arrive either individually or as a continuous communication stream.

Real-world examples include processing simple location data from a vehicle tracking system, or handling a continuous stream of data from a temperature sensor or lightning detector. A simpler example would be sending email to a system administrator in response to database table updates.

Real-time data in FME is handled in two ways: **Notifications** and **Message Streams.**

Notifications

Notifications are how FME Server handles individual messages and alerts.

FME Server has a specific Notification Service that can be set up to listen for an incoming message from outside of FME, and trigger a certain action in response; or that can be set up to send a single alert in response to an event that takes place on FME Server.

Protocols supported include email, websockets, and notifications for Apple and Android devices.

Message Streams

Message Streams are how FME Server handles a continuous flow of information; literally thousands of messages per second.

Rather than use the Notification service, message streams are handled by a number of transformers. A workspace is set up to run continuously and the transformers within it listen for, and dispatch, messages on a number of different protocols.

Protocols supported include TCP, JMS, and HTML5 WebSockets.

The Notification Service

The Notification Service

What is a Notification?
When to use Notifications
Elements of the Notification Service

FME Server Basic Training 2015

The Notification Service is a way for data to be pushed to and from FME Server in the form of short messages.

What is a Notification?

A notification is a simple message (sometimes called an “alert”) that informs someone or something that a particular event has happened.

Notifications on FME Server can occur in two different forms; incoming and outgoing.

Incoming notifications alert FME Server to an event that has taken place elsewhere.

Outgoing notifications alert users to an event that has taken place on FME Server.

In this way, FME Server can take action in response to an event notification, or a user can take action in response to a notification from FME Server.

The **Notification Service** is the part of the FME Server architecture that handles incoming and outgoing notifications.

When to Use Notifications

Notifications should be used when you want to trigger an FME Server response to an event that is reported from outside of FME. The event should not be a continuous series of messages; if there is more than one message per second you should consider using Message Streaming techniques instead.

Notifications are also for when you want to send a message about something that happened on FME Server, to an outside subscriber. Again, if there is likely to be more than one message per second you should consider using Message Streaming.

In either case a notification is for sending a brief message, usually in order to trigger an action from the recipient. It’s not intended to be used for transmitting large amounts of spatial data. At most you should use it for a single geographic feature such as a point location.

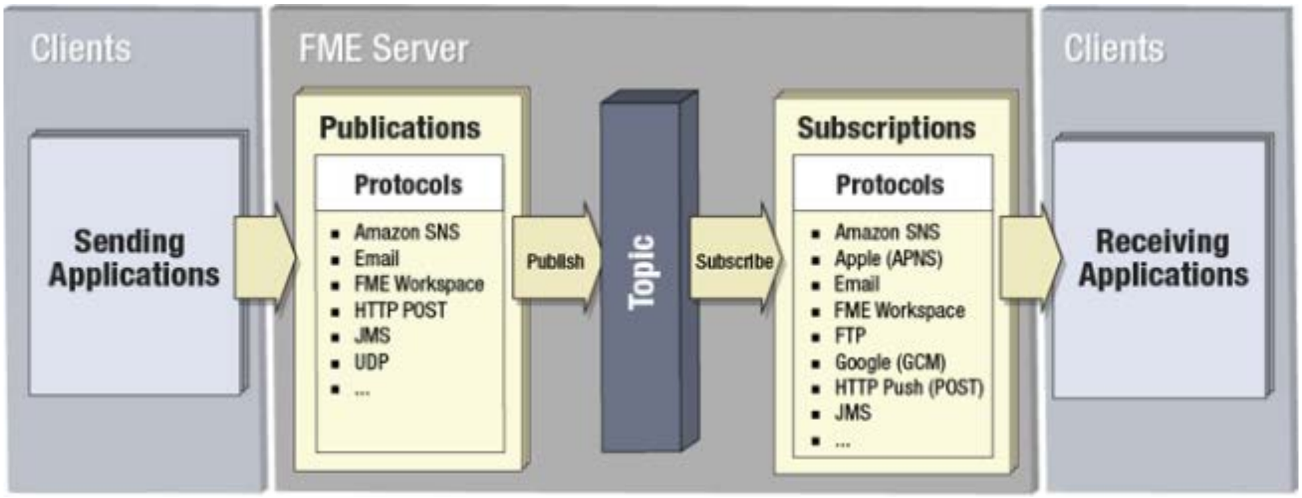
Elements of the Notification Service

The Notification Service includes a number of different components.

Clients	An external user or system that sends or receives a notification
Publications	An event handler that listens for incoming notifications
Subscriptions	An event handler that dispatches outgoing notifications
Topics	A subject that describes what the notification is about
Protocols	Methods by which FME Server can receive or send notifications

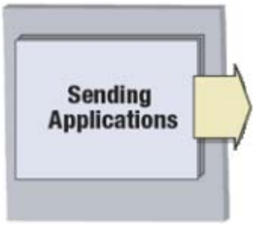
Although the diagram below shows a continuous process, from a sending client through FME Server to a receiving client, it is not necessary for all of these components to be used in a setup.

If the system is designed for FME Server to only receive notifications, then only the publications side of the diagram is relevant. Likewise, if the system is intended for FME Server to only send notifications, then only the subscriptions side of the diagram applies.



Clients

A client is a user or system that **sends** or **receives** a notification. The client may be a physical person, or may just be a component in a computer system.

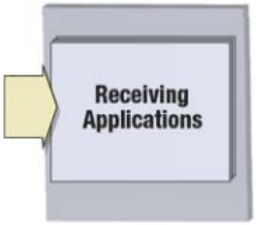


For example, a database update might cause a trigger to **send** a notification to FME Server, in which case the database system is the client. But a client could also be a person who, for example, triggers a notification by sending an email to FME Server.

Since they are sending information these clients are called “**Publishers**”.

Likewise, FME Server can send a notification for another client system to **receive.** Alternatively this client can also be a real person, who might receive a notification in the form of an email.

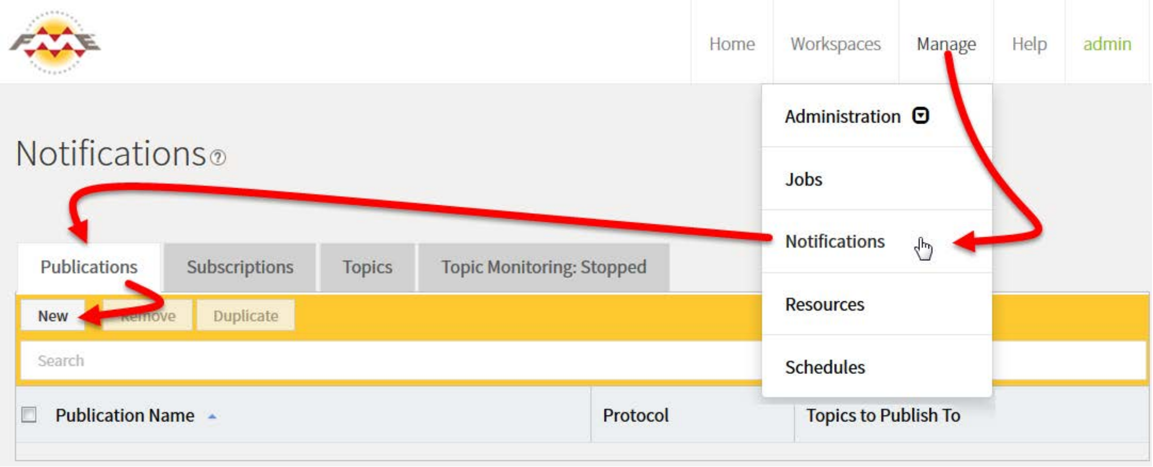
Since they are receiving information, these clients are called “**Subscribers**”.



Publication
Topic

Publications
A Publication is a component that receives **incoming** notifications *from* a client.

To receive notification in FME Server a user must create a new Publication.



A Publication is created in the FME Server Web User Interface under the Notifications section.

Topics
A Topic is a component that acts as a mediator for messages and defines the message content. Think of it as a subject line for notifications.

Like Publications, a Topic is created in the FME Server Web User Interface, under the Notifications section.

All Publications are linked to one (or more) Topics, so that when an incoming message is received it is categorized and the related Topic is triggered.

Because a Publication can be linked to multiple Topics, each incoming message can trigger multiple actions to occur. Additionally, multiple Publications can trigger the same topic.

For example, a lightning strike sensor might publish to the topics *WeatherEvent* and *AircraftAlerts*, whereas a flood sensor might publish to the topics *WeatherEvent* and *RoadConditions*.

Daily Interop Reporter, Chad Pugh-Litzer says…

"It's like when I write a news article. I publish the article to the Daily Interop web site, tagged with a number of topics to describe it.

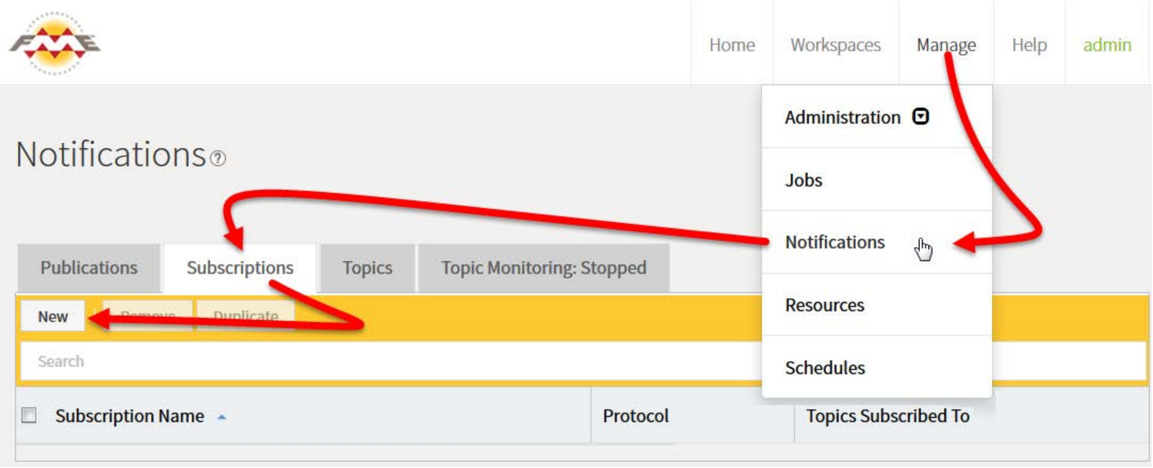
For example, a report about a soccer team's tax return would be filed under both 'Financial' and 'Sports' because it relates to both."

Subscriptions

A Subscription is a component that sends **outgoing** notifications *to* a client.

Topic
Subscription

To send a notification in FME Server a user must create a new Subscription.



A Subscription is created in the FME Server web interface under the Notifications section.

As with Publications, each Subscription is also tied to one (or more) Topics, and each Topic can be subscribed to by multiple Subscriptions.

For example, a police headquarters might subscribe to the *RoadConditions* topic, to receive notifications on that subject. The local TV weather channel also subscribes to the *RoadConditions* topics, but in addition subscribes to *WeatherEvent* to hear about those particular events.

InteropGeek68 says…

"I <u>subscribe</u> to articles published on the Daily Interop web site according to their <u>topic</u>.

For example, I subscribe to reports on technology and gaming, whereas my friend – InteropJock72 – subscribes to sports articles"

*Although the actions seem odd in regard to the name – Publications receive messages, Subscriptions send them – it is correct. The name refers to the **Client**, not the Server.*

So FME Server receives a Publication because <u>the client is publishing it</u>.
Likewise, FME Server sends a Subscription because <u>the client is subscribing to it</u>.

Protocols

A protocol is a system of data exchange between FME Server and a client. It is the **method** by which FME Server receives notifications from a client and sends notifications to a client.

E-mail is a messaging protocol, as are Apple and Android alerts.

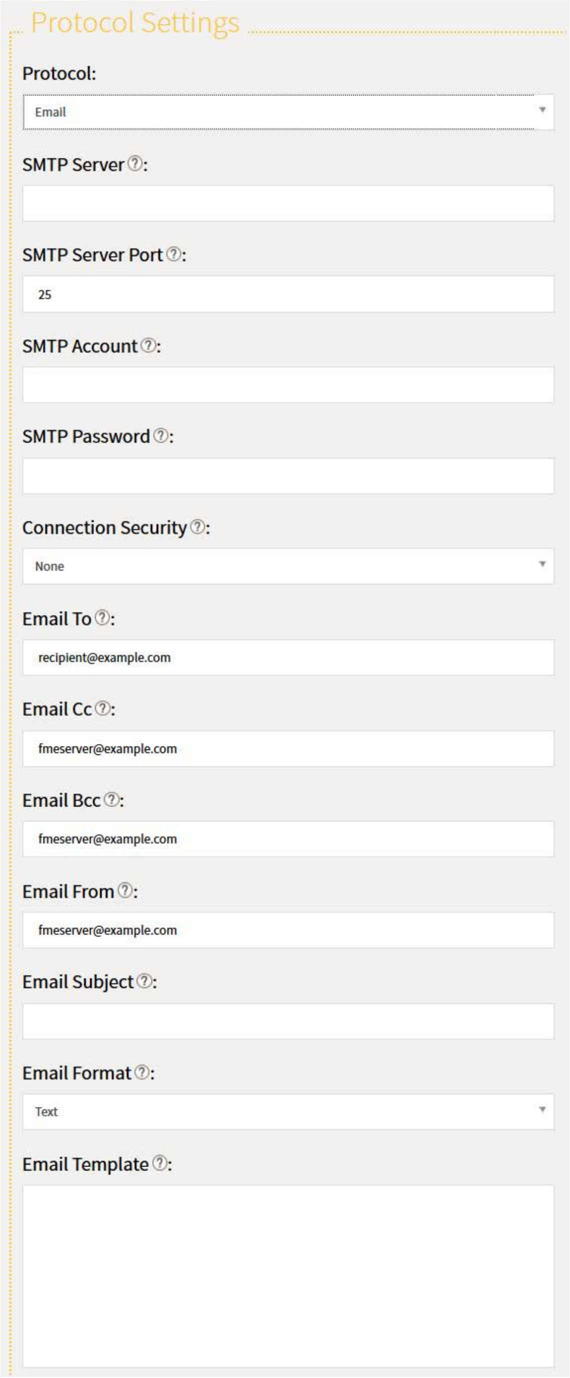
Each Publication and Subscription must be defined using a particular communication protocol. An e-mail Publication – for example – will accept notifications via an incoming email message. An Apple Subscription will send notifications to an Apple mobile device.

Protocols are pre-defined components in the FME Server architecture and do not need to be defined in the Web User Interface.

However, a number of fields are made available to configure them when a Publication or Subscription chooses to make use of that protocol.

For example, here are the parameters for an e-mail Subscription.

These parameters must be set when the Subscription is created as they are needed in order to be able to send out a notification using the e-mail protocol.



Protocol Settings

Protocol: Email

SMTP Server:

SMTP Server Port: 25

SMTP Account:

SMTP Password:

Connection Security: None

Email To: recipient@example.com

Email Cc: fmeserver@example.com

Email Bcc: fmeserver@example.com

Email From: fmeserver@example.com

Email Subject:

Email Format: Text

Email Template:

An Introduction to Email Notifications

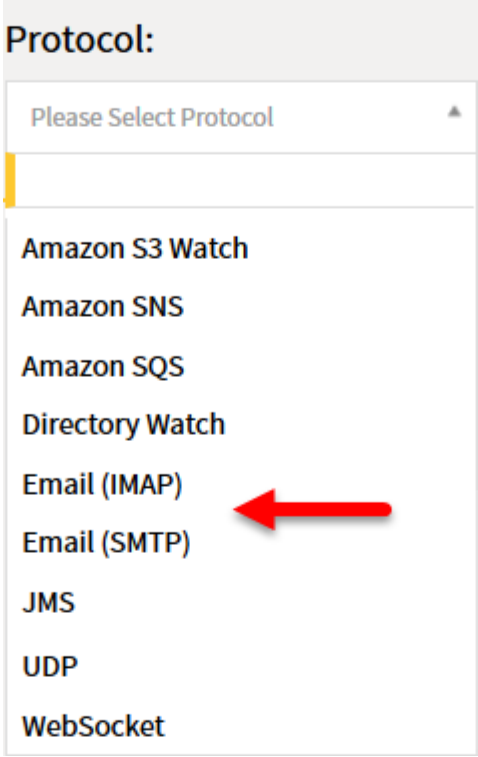
An Introduction to Email Notifications

Email Protocols
SMTP Publications
IMAP Publications
Exercise 4a: Basic Notifications and Email Subscription

Email is probably the most commonly used notification protocol

It’s important to cover email notifications in some detail because they are the most commonly used type of notification on FME Server.

“Email” is actually a protocol used by Publication and Subscription components. Email Publications receive incoming email and Email Subscriptions send outgoing email.



Protocol:

Please Select Protocol

- Amazon S3 Watch
- Amazon SNS
- Amazon SQS
- Directory Watch
- Email (IMAP)
- Email (SMTP)
- JMS
- UDP
- WebSocket

Email Protocols
FME Server can make use of email messages as both an incoming and outgoing notification protocol.

There are two email-related protocols available in FME Server: “SMTP” and “IMAP”.

The SMTP protocol is the ability to directly send or receive an email through an email server built into FME Server.

The IMAP protocol acts a little differently. It is an indirect process that connects to an email account on a server elsewhere.

When that account receives an email the IMAP protocol passes it on to FME Server.

SMTP can be used as both a Subscription and a Publication protocol; IMAP can only be used for a Publication (i.e. it can only be used to detect incoming emails).

SMTP Publications

SMTP Publications are when data is published to FME Server via a direct email. FME Server receives an email and triggers a Topic in response.

Setting up an Email Publication

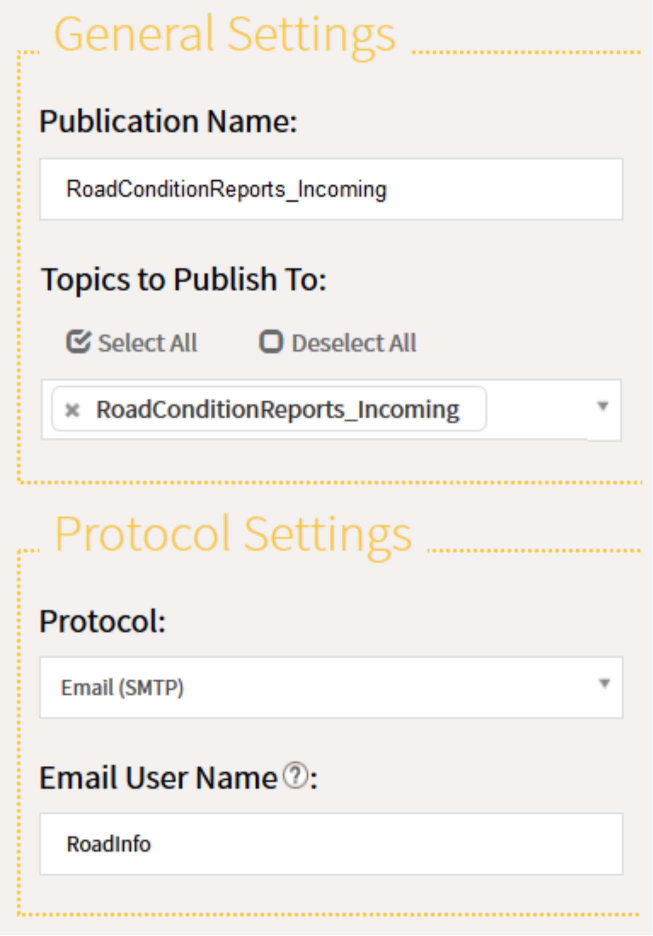
Creating an SMTP publication is done in the Notifications section of the Web User Interface, by choosing the Email (SMTP) protocol for a new Publication.

Once the protocol type is selected, the publication must also be given a name and an existing topic chosen to be triggered.

The final parameter is the Email User Name; in this example it is set up to be *RoadInfo*.

Notice the author is suffixing the publication and topic names with *_Incoming* so that there is no confusion over which topics are being used for incoming notification and which ones for outgoing.

Now, whenever an email is sent to RoadInfo@serverhost.com – for example someone is sending an email to report snow on the road – the *RoadConditions_Incoming* topic is triggered.



General Settings

Publication Name: RoadConditionReports_Incoming

Topics to Publish To:

☑ Select All □ Deselect All

× RoadConditionReports_Incoming

Protocol Settings

Protocol: Email (SMTP)

Email User Name: RoadInfo

Setting up the Email Server

SMTP email publications are possible because FME Server includes an in-built email server as one of its components. However, this does require that the server has a domain name registered with a DNS name server.

The steps to set up the email server for notifications are documented in the FME Server Reference Manual, and also in the following exercise.

Sister Intuitive says …

“FME Cloud instances are automatically configured for email notification, and have a public domain name too, so you don’t need to do any additional setup.”

IMAP Publications

IMAP (Internet Message Access Protocol) is a variation on email for incoming (Publication) notifications.

Instead of using the in-built email server, an IMAP Publication connects to another email server and monitors it for incoming email. When an email arrives in that account then the Topic to which it is tied becomes triggered.

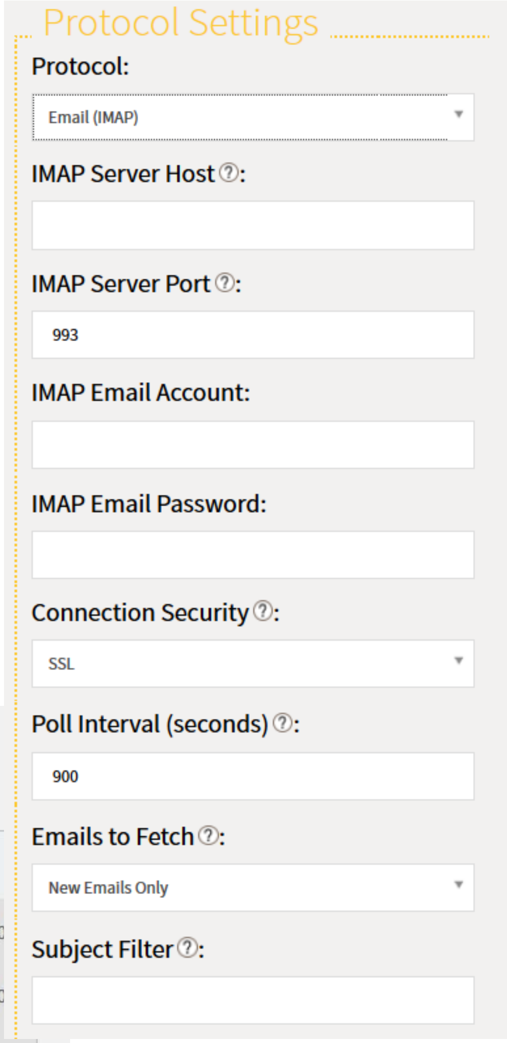
Setting up an IMAP Publication

Like the SMTP protocol, IMAP Publications are set up in the Notifications section of the Web User Interface, by creating a new Publication and choosing the Email (IMAP) protocol.

Here are the parameters for an IMAP Publication:

Notice that most parameters are for defining the IMAP (Email) server connection.

Two important parameters let you decide the interval to check for emails and decide whether to fetch all unread emails or new emails only.



Protocol Settings

Protocol: Email (IMAP) ▾

IMAP Server Host ⓘ:

IMAP Server Port ⓘ: 993

IMAP Email Account:

IMAP Email Password:

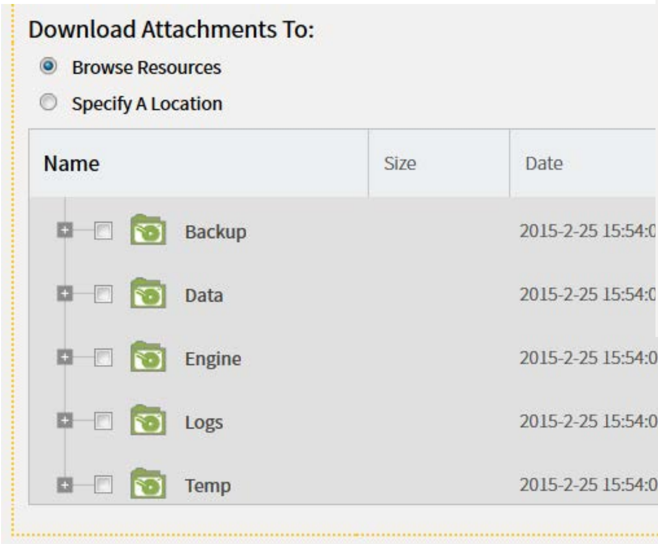
Connection Security ⓘ: SSL ▾

Poll Interval (seconds) ⓘ: 900

Emails to Fetch ⓘ: New Emails Only ▾

Subject Filter ⓘ:

There is also a parameter to select an FME Server resource location in which to store any email attachments:



Download Attachments To:

● Browse Resources
○ Specify A Location

Name	Size	Date
Backup		2015-2-25 15:54:01
Data		2015-2-25 15:54:01
Engine		2015-2-25 15:54:01
Logs		2015-2-25 15:54:01
Temp		2015-2-25 15:54:01

Monsieur D. Server says …

"Most email servers support IMAP functionality, as do the majority of cloud-based email providers such as Gmail, Outlook.com, Yahoo!, etc; so it's very easy to have FME Server scan a Gmail account for incoming mail, and then act on its contents."

Exercise 4a: Basic Notifications and Email Subscription	
Scenario	Airphoto Data Vendor
Data	GeoTiff Orthophotos
Overall Goal	Provide email-driven self-serve access to image files
Demonstrates	Email Publications and Topic Notification
Starting Workspace	n/a
Finished Workspace	n/a

This exercise is the first part in setting up a system that provides email-driven access to image files. In this part we'll concentrate on setting up a Topic and Publication.

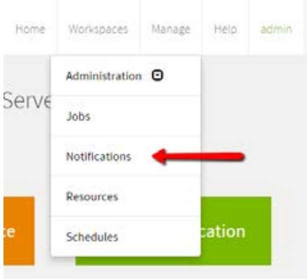
For this setup you may use either the SMTP or IMAP protocol for email.

For SMTP, this exercise requires your FME Server to have a DNS record and SMTP configured.

For IMAP, this exercise requires access to an email server that supports the IMAP protocol. Gmail, Outlook, and Yahoo! all are acceptable web-based solutions.

1. Create a Topic
The first step is to create a topic that will be triggered by the email.
Open the Web User Interface in a web browser by visiting http://<servername>/fmeserver

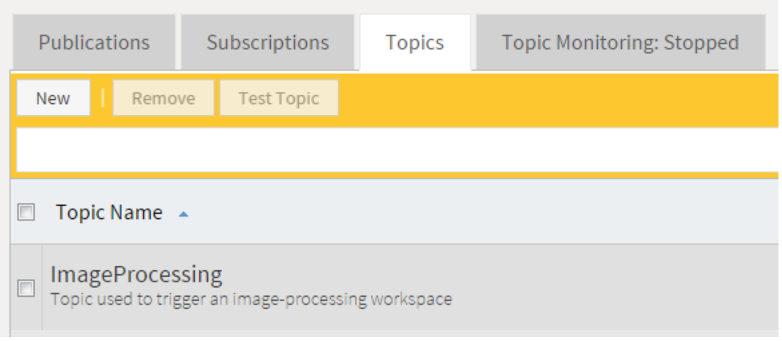
Click on the menu item labelled Manage > Notifications and then click the tab labelled Topics.



Click New and enter a topic name such as ImageProcessing.

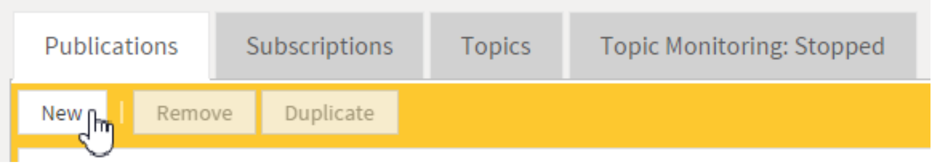
Enter a description such as "Topic used to trigger an image processing workspace".

Click OK to close the dialog and create the Topic.



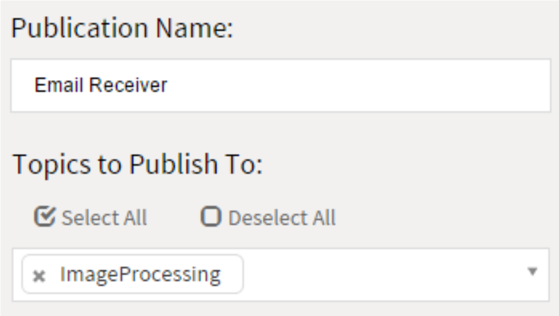
ImageProcessing
Topic used to trigger an image-processing workspace

2. Create a Publication
In the Notifications part of the web interface click on the Publications tab. Click the New button to create a new Publication.



The new publication can be created to use either the Email protocol or the IMAP protocol.

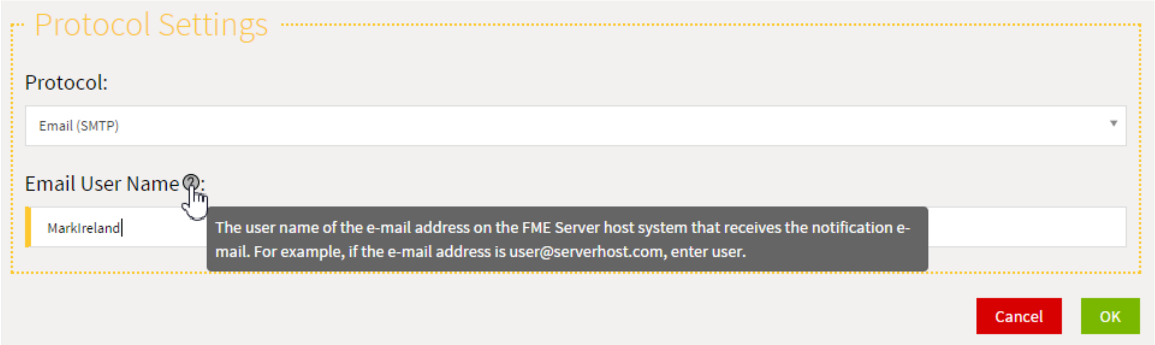
Email Protocol
Enter *Email Receiver* as the new publication's name. Then click in the text box labelled *Click to Select* under *Topics to Publish To*. Select the newly created ImageProcessing topic from the drop-down list.


Publication Name: Email Receiver

Topics to Publish To: ImageProcessing

Now select Email (SMTP) as the Publication Protocol. This will open the Email User Name parameter. Enter a name for receiving email, for example in the format *FirstnameLastname*

This will create an email address <name>@<hostname>
The hostname is assumed, so you don't need to enter it yourself.


Protocol Settings

Protocol: Email (SMTP)

Email User Name: MarkIreland

The user name of the e-mail address on the FME Server host system that receives the notification e-mail. For example, if the e-mail address is user@serverhost.com, enter user.

Now all emails sent to that address will trigger the ImageProcessing topic.

InteropGeek68 says …

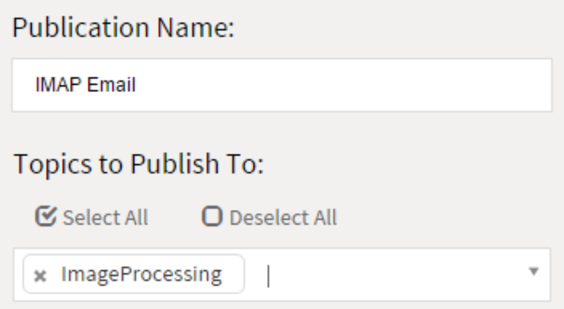
"If you don't have access to a web-based email account, use fmeserver2014@gmail.com with the password lizardisland."

IMAP Protocol

IMAP is the easier way to handle incoming email where computers might not have a proper DNS record, such as on an internal network.

Enter IMAP Email as the new publication's name.

Next, click in the text box labelled *Click to Select* under *Topics to Publish To*. Select the newly created ImageProcessing topic from the drop-down list.



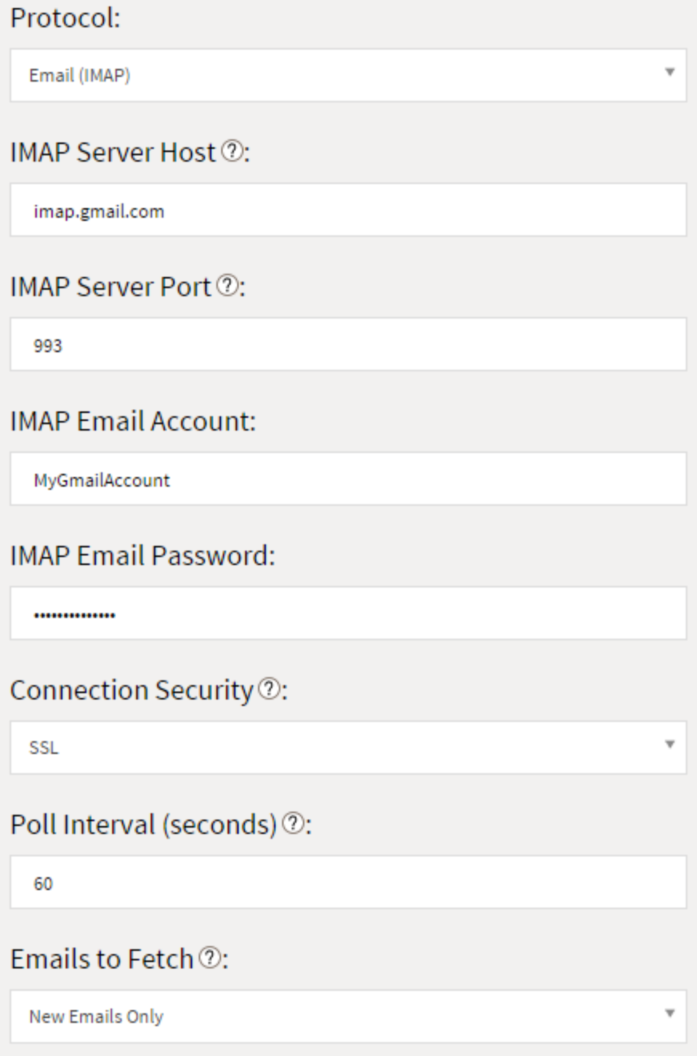
Publication Name:

IMAP Email

Topics to Publish To:

× ImageProcessing

Now select Email (IMAP) as the Publication Protocol. This will open a number of other parameters. Enter them according to your email account:



Protocol: Email (IMAP)

IMAP Server Host: imap.gmail.com

IMAP Server Port: 993

IMAP Email Account: MyGmailAccount

IMAP Email Password: ••••••••••••••

Connection Security: SSL

Poll Interval (seconds): 60

Emails to Fetch: New Emails Only

In case it is of use, the usual server information is as follows:

	Gmail	Outlook	Yahoo!
IMAP Server Host	imap.gmail.com	imap-mail.outlook.com	imap.mail.yahoo.com
IMAP Server Port	993	993	993
IMAP Email Account	<email address>	<email address>	<email address>
IMAP Email Password	<password>	<password>	<password>
Connection Security	SSL	SSL	SSL

Set the Poll Interval to 60 seconds and Emails to Fetch to *New Emails Only*.

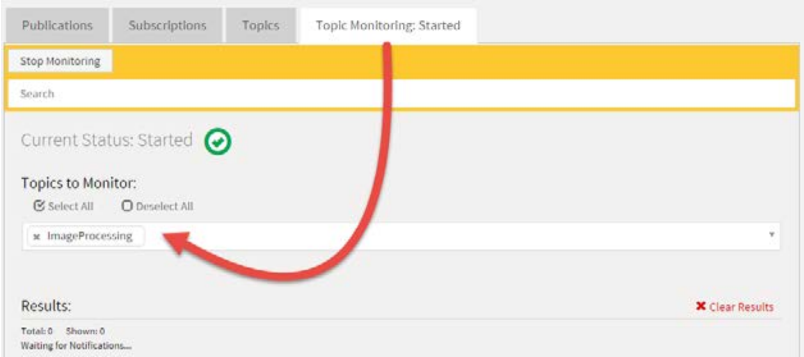
Select a Resource Folder for attachments to be saved to and click **OK** to close the dialog and create the new Publication.

3. Start Monitoring Topic

The monitoring tool lets you know when a notification is received.

NB: *To use monitoring requires the WebSocket port (usually 7078) to be open. This should automatically be the case on FME Cloud, but other systems may require some manual firewall configuration.*

On the Notifications page click the tab labelled *Topic Monitoring* and select ImageProcessing as the topic to monitor:



Current Status: Started

Topics to Monitor: ImageProcessing

Results:

Total: 0 Shown: 0

Waiting for Notifications...

4. Send an Email

Now let's test the setup so far by sending an email to FME Server. At the moment you can use any subject line or contents for the email, as FME is not yet using them to do anything.

If you chose the SMTP protocol then the email should be sent to <emailname>@<hostname>.

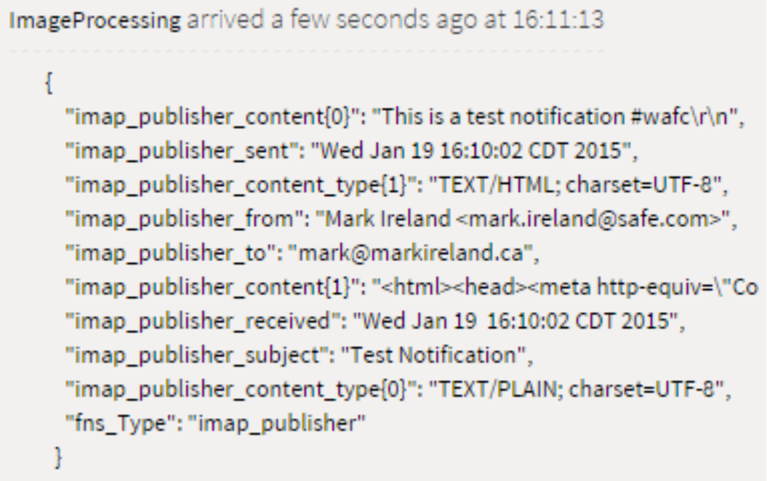
For example an FME Cloud address would be like this:
`MyName@server-tutorial.safe-software.fmecloud.com`

And an Amazon AWS machine would have an address like this:
`MyName@ec2-54-224-106-213.compute-1.amazonaws.com`

If you used the IMAP protocol then send an email to the account you chose in the Publication.

For IMAP there will probably be a slight delay because FME is only checking for email every minute, plus it will take a short while for the email to arrive. However, shortly the email will be recognized by FME Server, the notification topic triggered, and the activity reported by the monitor:

Congratulations!
You have now triggered an FME Server notification using email.



ImageProcessing arrived a few seconds ago at 16:11:13

```
{
  "imap_publisher_content{0}": "This is a test notification #wafc\r\n",
  "imap_publisher_sent": "Wed Jan 19 16:10:02 CDT 2015",
  "imap_publisher_content_type{1}": "TEXT/HTML; charset=UTF-8",
  "imap_publisher_from": "Mark Ireland <mark.ireland@safe.com>",
  "imap_publisher_to": "mark@markireland.ca",
  "imap_publisher_content{1}": "<html><head><meta http-equiv=\"Co
  "imap_publisher_received": "Wed Jan 19  16:10:02 CDT 2015",
  "imap_publisher_subject": "Test Notification",
  "imap_publisher_content_type{0}": "TEXT/PLAIN; charset=UTF-8",
  "fns_Type": "imap_publisher"
 }
```

Notifications and Workspaces

Notifications and Workspaces

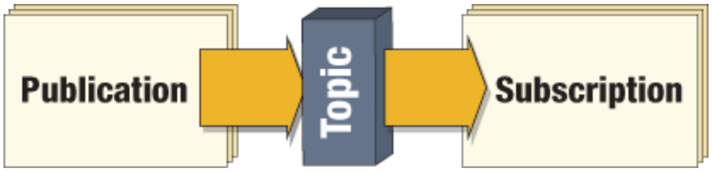
Message Transformation
Workspace as a Subscription
Exercise 4b: Basic Notifications and Email Subscription
Exercise 4c: Basic Notifications and Email Publication

FME Server Basic Training 2015

Workspaces are what make FME Server the ideal engine for Spatial Data notifications

Message Transformation

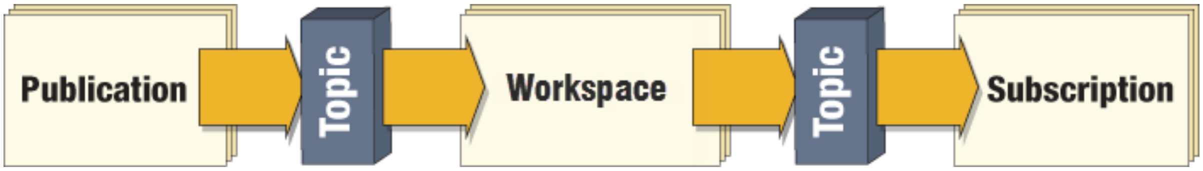
As shown in the previous exercise, the simplest setup for the FME Server Notification Service is an incoming message (Publication) tagged with a specific topic that gets passed as an outgoing message to any Subscription registered with that topic:



However, that scenario does not include any transformation – restructuring – of the message contents. If the message needs to be processed in some way then an FME workspace can be employed between Publication and Subscription.

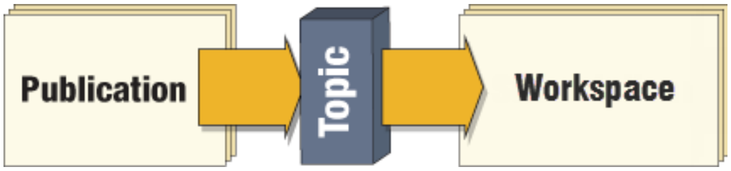
A workspace is the key functionality of FME and with one you can read spatial data from a message, in almost any format, carry out spatial transformations on that data, and then pass on the results to a set of subscribers.

This blend of live messaging with Spatial ETL is unique.



In this scenario a workspace is literally a Subscriber to the incoming Publication, and receives the message content from it. The workspace then acts as a Publisher, sending an outgoing message to any Subscribers.

Of course, the above diagram shows the full potential of a notification system. It would be just as appropriate to have a system that triggered a workspace in response to a Publication, but without the workspace sending an outgoing message:



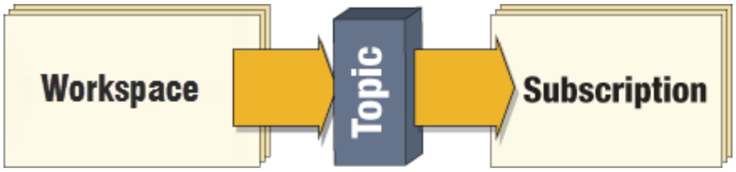
For example, FME might listen for a message from a database trigger, and run a workspace in response.

The workspace just writes output and does not need to send any outgoing message.

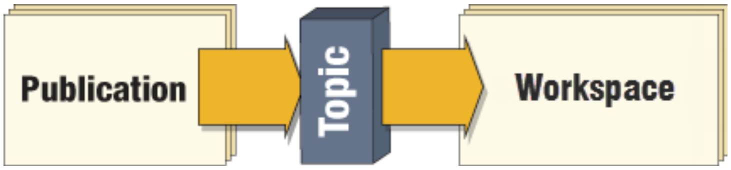
Similarly, it's just as appropriate to have a workspace that is run by some other means, and which sends an outgoing message to a Subscription when it is complete:

For example, a workspace might be started as a scheduled task.

When the task is complete FME sends a notification email to inform an administrator.



Workspaces as a Subscription
When a workspace needs to react to an incoming notification it is done by connecting the workspace to a particular topic. The workspace is run in response to that topic being triggered by a Publication.

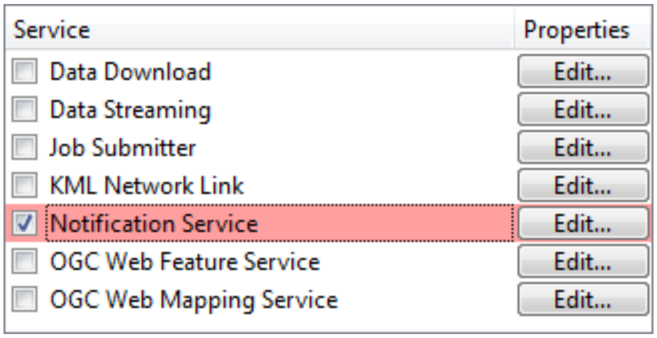


In effect, the workspace is subscribing to that topic.

In this case the message is received and passed into the workspace for processing.

Registering a Workspace
A workspace is set up to receive messages when it is published to FME Server and registered with the Notification service.

The Service is initially highlighted in red because there are several parameters that need to be set before it can be used.

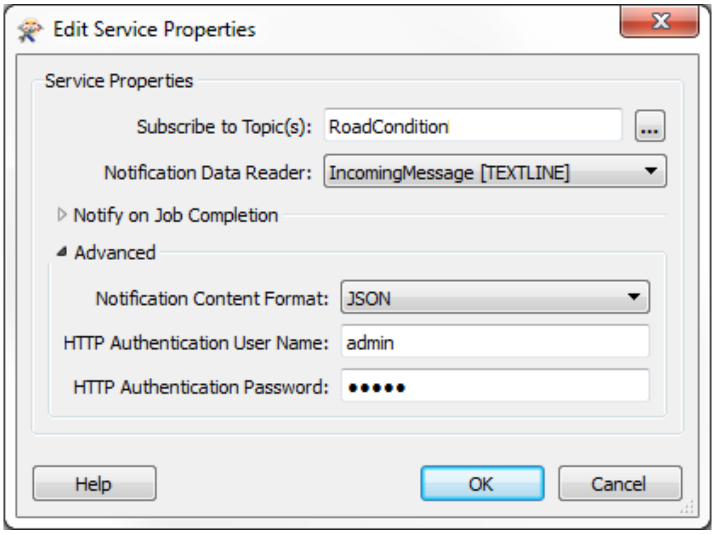


Service	Properties
□ Data Download	Edit...
□ Data Streaming	Edit...
□ Job Submitter	Edit...
□ KML Network Link	Edit...
☑ Notification Service	Edit...
□ OGC Web Feature Service	Edit...
□ OGC Web Mapping Service	Edit...

Two important parameters are Subscribed Topics and Notification Reader

Subscribed Topics tells FME which notification topics should trigger the workspace.

Notification Reader tells FME which Reader should receive the notification content.



Edit Service Properties

Service Properties

Subscribe to Topic(s): RoadCondition

Notification Data Reader: IncomingMessage [TEXTLINE]

Notify on Job Completion

Advanced

Notification Content Format: JSON

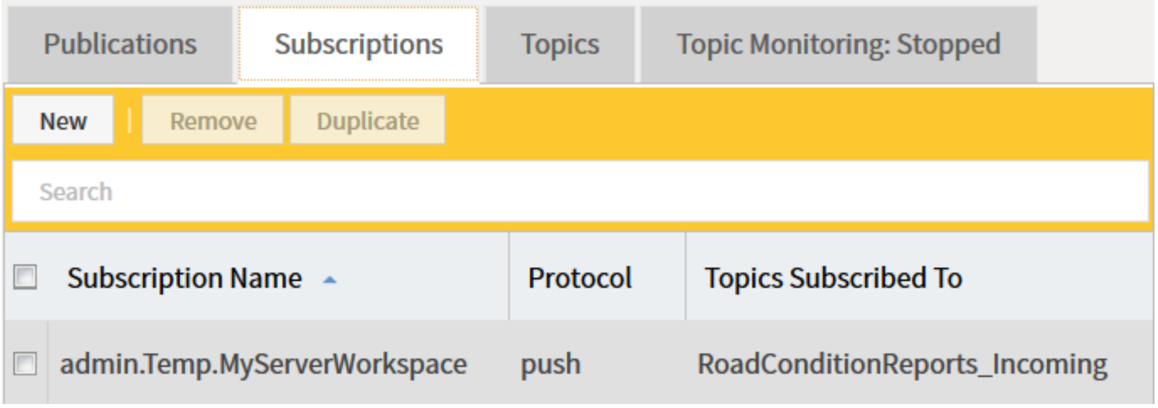
HTTP Authentication User Name: admin

HTTP Authentication Password: •••••

In this example any incoming message that is flagged under *RoadCondition* will cause the workspace to run. The message content will be funneled through a plain text Reader called – appropriately enough – *IncomingMessage*.

Managing Workspace Subscription

Once a workspace is published and registered with the notification server as a subscriber it appears in the Notification section of the Web user interface.

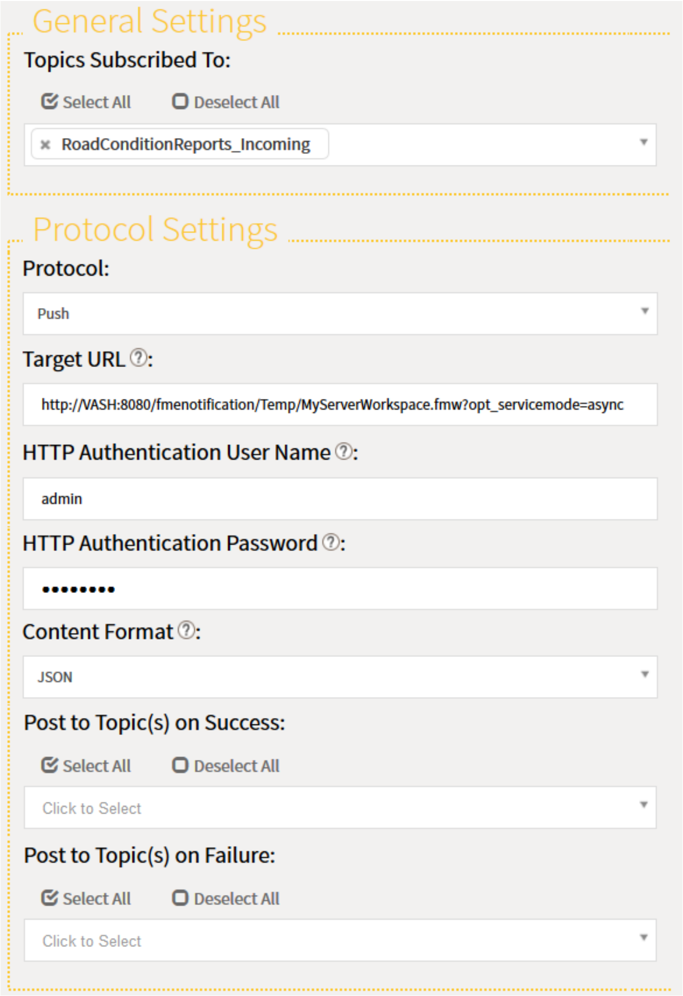


□	Subscription Name ▲	Protocol	Topics Subscribed To
□	admin.Temp.MyServerWorkspace	push	RoadConditionReports_Incoming

The workspace appears as a Subscription because it is literally subscribed to the topic(s) shown.

It is listed with the protocol "push" because FME Server is pushing a notification using HTTP post to run the workspace.

Clicking on the subscription allows you to make edits to the notification without having to republish the workspace; for example topics can be added or removed.



Topics Subscribed To: × RoadConditionReports_Incoming

Protocol: Push

Target URL: http://VASH:8080/fmenotification/Temp/MyServerWorkspace.fmw?opt_servicemode=async

HTTP Authentication User Name: admin

HTTP Authentication Password: ••••••••

Content Format: JSON

Post to Topic(s) on Success: Click to Select

Post to Topic(s) on Failure: Click to Select

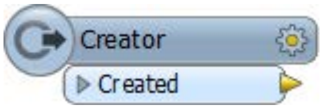
Exercise 4b: Basic Notifications and Email Subscription	
Scenario	Airphoto Data Vendor
Data	GeoTiff Orthophotos
Overall Goal	Provide email-driven self-serve access to image files
Demonstrates	Email Publications and Topic Notification
Starting Workspace	n/a
Finished Workspace	C:\FMEData2015\Workspaces\ServerAuthoring\Exercise4b-Complete.fmw

This exercise continues setting up a system that provides email-driven access to image files. In this part we'll create a workspace that will run in response to the incoming notification.

1. Create a Workspace

Now we have a Publication and a Topic that it will trigger when FME Server discovers a new email in the defined email account. However, at the moment there is nothing subscribed to that publication to act upon it. We'll handle that part of the solution with a workspace.

Start FME Workbench. Begin with a blank canvas and simply add a Creator transformer. This will give us a workspace to run in response to an email; albeit one that doesn't do much yet.

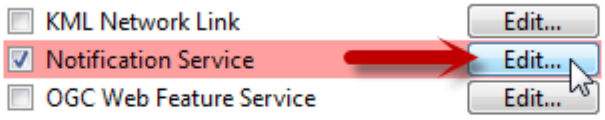


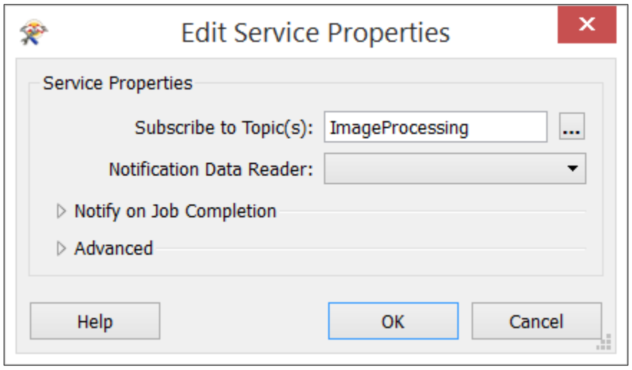
Save the workspace as Exercise4b.fmw

2. Publish Workspace

Now let's publish the workspace to FME Server. Start the Publishing Wizard using File > Publish to Server on the menubar. When prompted enter the same connection parameters as before. Publish the workspace to the Training repository. There are no data files that need to be published along with the workspace.

In the services dialog put a checkmark in the Notification Service's box. Click **Edit** to see the Service Properties for this service.


□ KML Network Link Edit...
☑ Notification Service Edit...
□ OGC Web Feature Service Edit...


Edit Service Properties

Service Properties

Subscribe to Topic(s): ImageProcessing ...

Notification Data Reader:

Click the […] button next to Subscribe to Topic(s) to retrieve a list of available topics from the server.

Select the ImageProcessing topic created in exercise 4a.

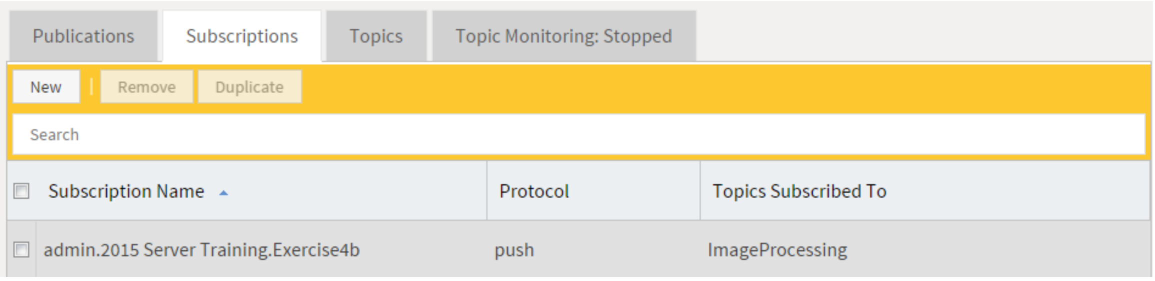
All other parameters can be left as they are, so simply click **OK** to close the dialog.

Then click **Publish** to finish publishing the workspace.

3. Confirm Subscription

Remember, the workspace is now effectively a subscriber to the email publication. To confirm this, return to the web interface. Select Notifications from the Manage menu and then click the Subscriptions tab.

You should see a push subscription for your workspace associated with the topic ImageProcessing. Click on the subscription to view the details.



☐ Subscription Name ▲	Protocol	Topics Subscribed To
☐ admin.2015 Server Training.Exercise4b	push	ImageProcessing

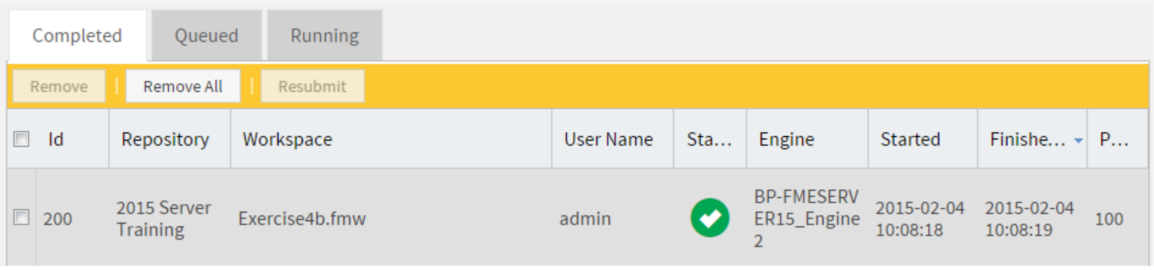
Click Cancel as we don’t need to make any changes to this component.

4. Send an Email

As before, send an email to trigger the notification and monitor the topic to ensure that it arrives.

5. Confirm Notification

To confirm the notification has caused the workspace to run, open the web interface to FME Server and click the Jobs option on the Manage menu. You should see a completed job for workspace Exercise4b.fmw, where FME ran the workspace in response to the email you just sent.



☐	Id	Repository	Workspace	User Name	Sta…	Engine	Started	Finishe… ▾	P…
☐	200	2015 Server Training	Exercise4b.fmw	admin	✓	BP-FMESERVER15_Engine 2	2015-02-04 10:08:18	2015-02-04 10:08:19	100

Congratulations! *You have now triggered an FME Server notification using email.*

Mr. Flibble says …

“If you’re using IMAP then this is probably a good point to go back to Notifications > Subscriptions, select the Exercise4b push subscription, and unsubscribe it from the ImageProcessing topic.

If you don’t do this, then forever more any incoming email to your account will trigger this workspace! And any attachments would get stored in the Resources folder. Wouldn’t that be funny! Ha ha!”

Message Content

The ability to interpret and process message content is a key reason for using FME workspaces as a subscriber. They can receive the message and then transform it in whatever way is required.

Content is received in a workspace by a Reader.

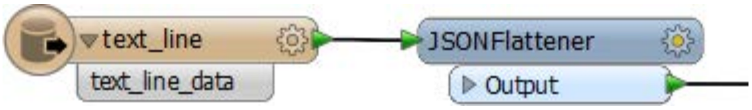
When the workspace is run (in response to a message) the notification content is substituted in place of the source dataset for a Reader feature type, so that a feature emerges from that reader with the message content attached as an attribute.

Content Format

Notification content can be read by a workspace in one of the following formats:

- JSON (Javascript Object Notation)
- XML (Extensible Markup Language)

For example, here a workspace has a Reader that will receive the message content.



Notice that the above workspace uses a Text File Reader.

Reading the incoming message as plain text is the most flexible method because the message is kept intact inside a single attribute and not processed into features. It would also allow you to create one workspace that could deal with both content formats.

The message can then be scanned and processed with a number of different transformers. If the messages are in JSON format there are transformers such as the JSONExtractor and JSONFlattener. Similarly there are XMLFlattener and XMLFragmenter transformers for XML content.

Handling Email Publications in a Workspace

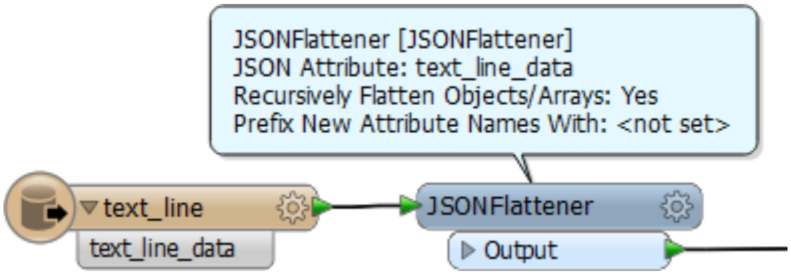
As previously described, content from a notification will be passed into the workspace to a Reader, usually a plain text Reader, and stored in an attribute.

For a JSON email the content will look like this:

```
{
"fns_type": "email_publisher",
"email_publisher_to": "demo@somehost.com",
"email_publisher_subject": "MIME message from sender",
"email_publisher_content{0}": "Testing Email",
"email_publisher_content_type{0}": "text/plain",
"email_publisher_from": "sender@somehost.com",
"email_publisher_received": "Thu May 17 11:15:46 PDT 2012",
"email_publisher_sent": "Thu May 17 11:15:46 PDT 2012",
"email_publisher_attachment{0}": "C:\\Temp\\demo246129673106713_canada.xsd"
}
```

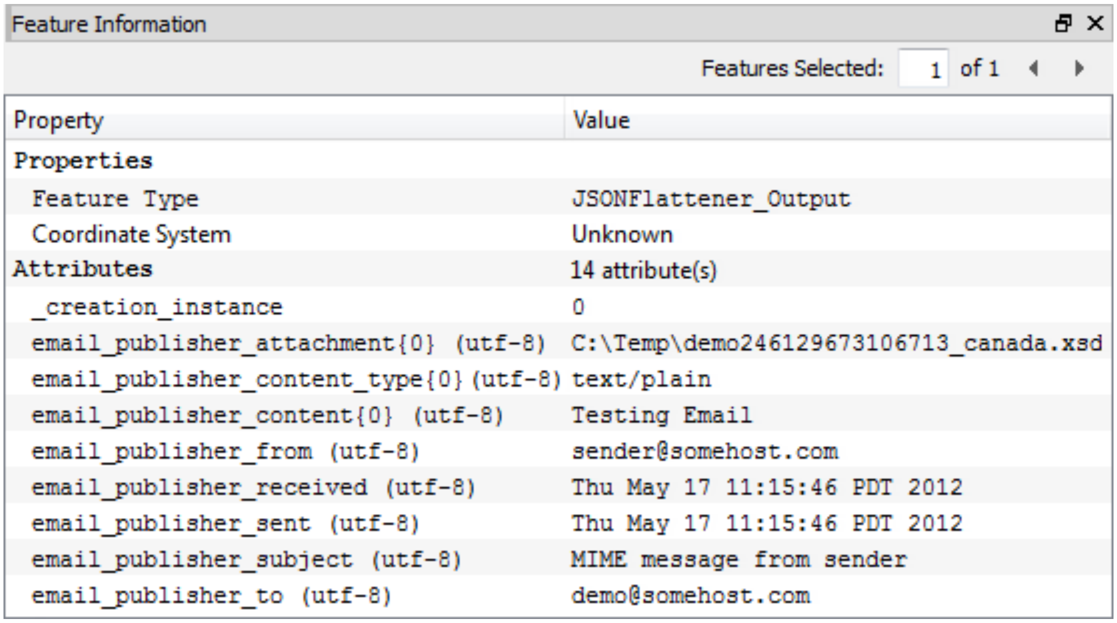
Notice how it includes the email from and to fields, plus the content itself. If there are attachments they will be stored on the file system and a path substituted into the JSON content.

This content can be converted into FME attributes using the JSONFlattener transformer:



JSONFlattener [JSONFlattener]
JSON Attribute: text_line_data
Recursively Flatten Objects/Arrays: Yes
Prefix New Attribute Names With: <not set>

The result – as shown in the FME Data Inspector – will look something like this:



Feature Information

Features Selected: 1 of 1

Property	Value
Properties	
Feature Type	JSONFlattener_Output
Coordinate System	Unknown
Attributes	14 attribute(s)
_creation_instance	0
email_publisher_attachment{0} (utf-8)	C:\Temp\demo246129673106713_canada.xsd
email_publisher_content_type{0} (utf-8)	text/plain
email_publisher_content{0} (utf-8)	Testing Email
email_publisher_from (utf-8)	sender@somehost.com
email_publisher_received (utf-8)	Thu May 17 11:15:46 PDT 2012
email_publisher_sent (utf-8)	Thu May 17 11:15:46 PDT 2012
email_publisher_subject (utf-8)	MIME message from sender
email_publisher_to (utf-8)	demo@somehost.com

Now the content is available to the workspace and can be processed as required.

Exercise 4c: Basic Notifications and Email Publication	
Scenario	Airphoto Data Vendor
Data	GeoTiff Orthophotos
Overall Goal	Provide email-driven self-serve access to image files
Demonstrates	Email Publications and Topic Notification
Starting Workspace	C:\FMEData2015\Workspaces\ServerAuthoring\Exercise4c-Begin.fmw
Finished Workspace	C:\FMEData2015\Workspaces\ServerAuthoring\Exercise4c-Complete.fmw

Having already set up a solution for incoming email notifications (in Exercise 4b) we can now expand upon that to let the user select a neighborhood in the email subject line. This neighborhood will be used to select a feature to clip the data with.

1. Open the Starting Workspace

In FME Workbench open the starting workspace for this exercise.

You **must** open this workspace as the exercise does not continue on from Exercise 4b. This workspace has more content and is similar to those created in Chapter 3; it reads and resamples raster data, clips it to a boundary, and writes it to a JPEG file.

2. Add a Text File (TEXTLINE) Reader

To read information into a workspace, from a notification, requires the use of a Reader. In this case we'll use a Text File Reader.

Select Readers >Add Reader from the menubar and add a Text File reader using the following settings:

Source Format:	Text File
Source Dataset:	C:\FMEData2015\Resources\emailIMAP.json

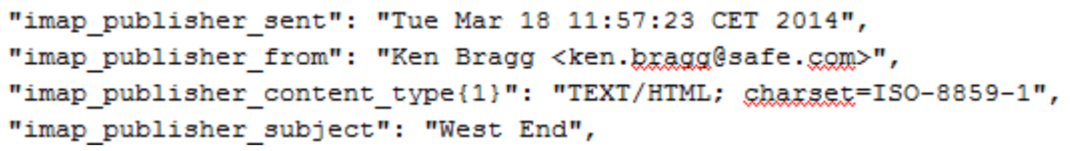
Parameters:
Read Whole File at Once: Yes

The *Read Whole File* parameter is VERY important! Click **OK** and **OK** again to add the Reader.

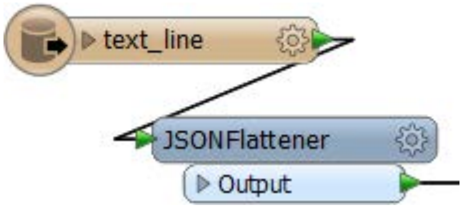
The file being read here is provided because all Readers need a source dataset, but also as an example of what Notification Service sends to a workspace when FME Server receives an email.

Examine the file in a text editor. It contains JSON content. You can see a number of JSON attributes available, which come from an email, and you will be able to read and use any of these in a workspace. In this exercise we'll be using the Subject of the email.

Because you set the Parameter to read the whole file at once, all of the contents will be read into a single attribute in a single feature, rather than creating multiple features.



```
"imap_publisher_sent": "Tue Mar 18 11:57:23 CET 2014",
"imap_publisher_from": "Ken Bragg <ken.bragg@safe.com>",
"imap_publisher_content_type{1}": "TEXT/HTML; charset=ISO-8859-1",
"imap_publisher_subject": "West End",
```



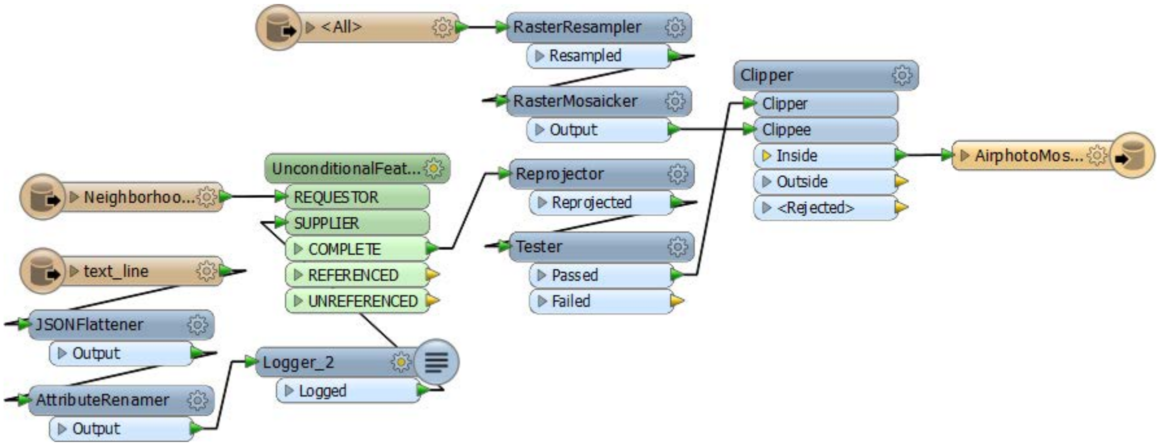
3) Process the JSON Message

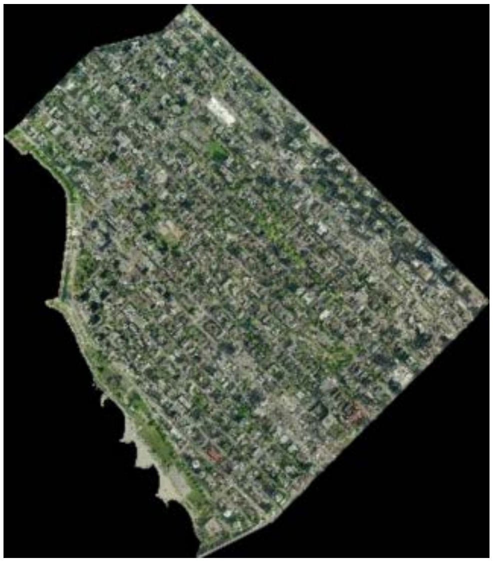
Back in the workspace, connect the Text Line source feature type to the JSONFlattener already in the workspace.

Click the cog wheel icon to open the parameters dialog for the JSONFlattener.

Examine the parameters. This transformer flattens the JSON attributes into FME attributes and exposes all of them.

Browse through the rest of the workspace. It also reads a KML dataset of neighborhood polygons and then filters the neighborhood by the subject of the email to clip by that neighborhood. The UnconditionalFeatureMerger is a custom transformer (available on the FME Store) that simply merges data together without any matching key.





4) Test the Workspace in FME Workbench

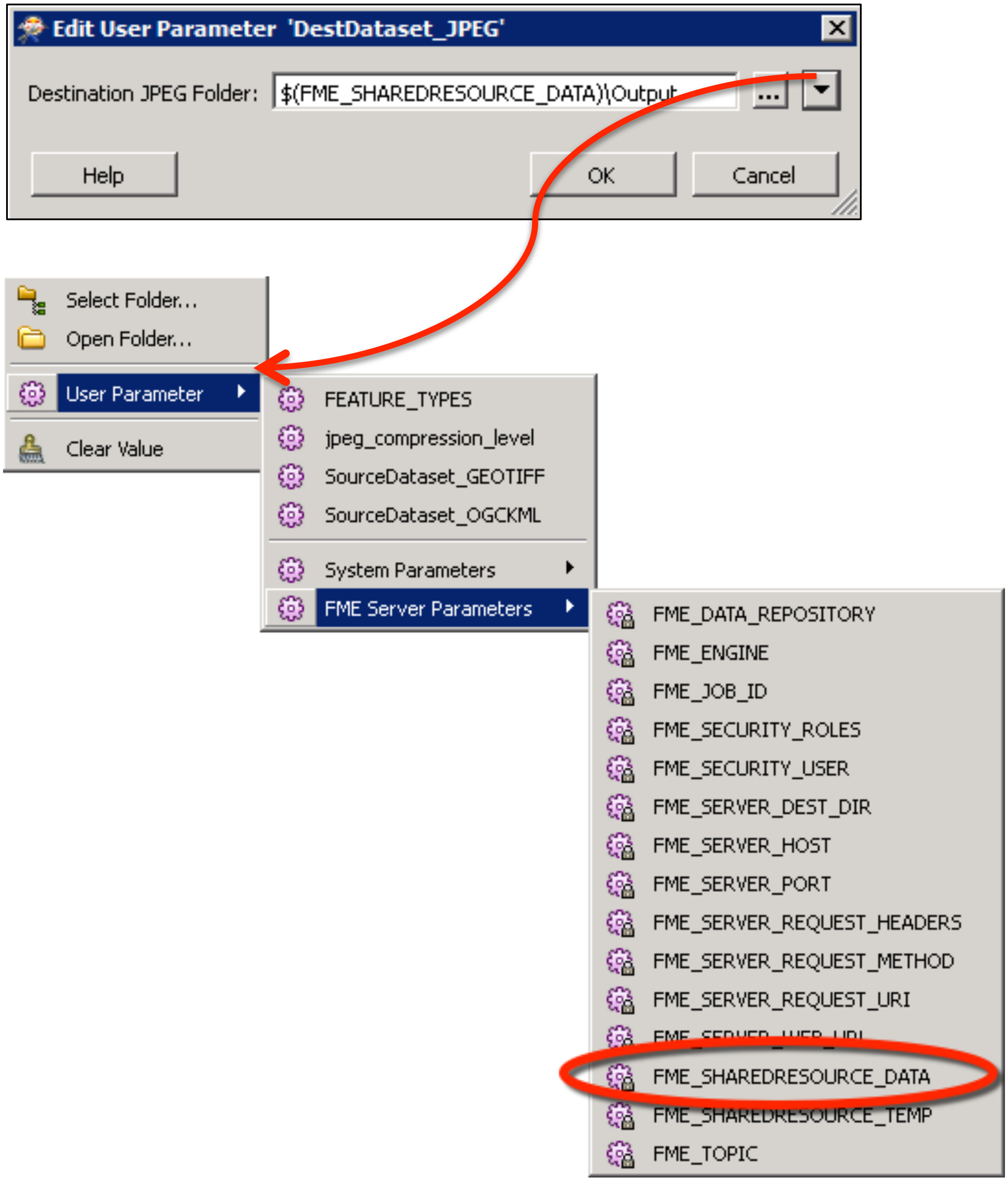
Test the workspace by running it in FME Workbench.

This will work because the text file contains a sample email where the neighborhood is coming from the *email_publisher_subject* attribute. If you wish, you can edit the file and use either “Downtown” or “West End” as the neighborhood.

Inspect the output. It should look like this:

5) Set Writer Path

Set the JPEG writer path to the Output folder of Server Resources.



Edit User Parameter 'DestDataset_JPEG'

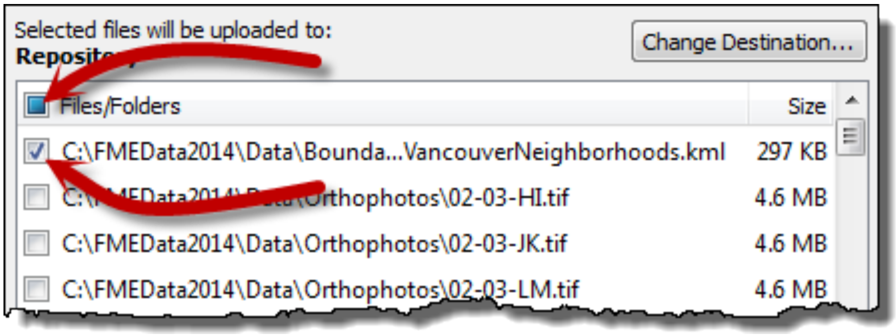
Destination JPEG Folder: $(FME_SHAREDRESOURCE_DATA)\Output

6) Publish to FME Server (Step 1)

Start the Publish Wizard from the menu or toolbar and enter the connection parameters when prompted. Select the Training repository to store the workspace, as usual.

Click the **Select Files** button in the dialog and ensure that **only** the KML dataset VancouverNeighborhoods.kml gets uploaded to the repository.

The easiest way to do this is to click the Files/Folders box to turn off ALL files, and then simply turn the neighborhoods file back on:



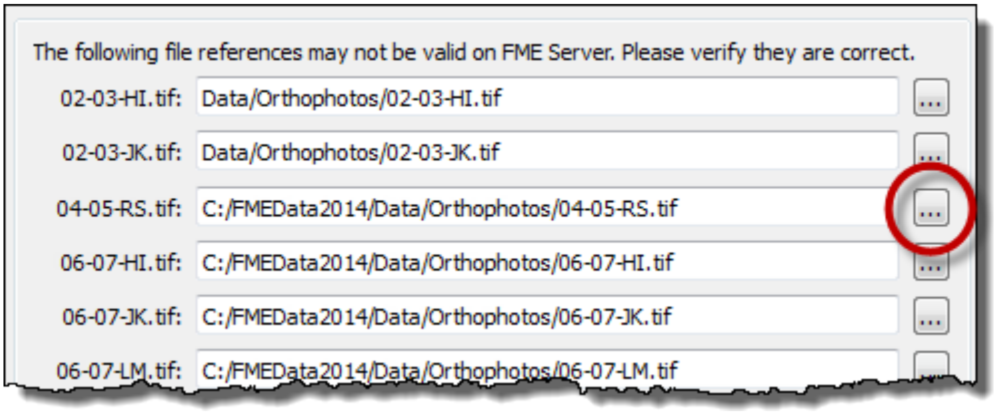
Selected files will be uploaded to:
Reposit

■ Files/Folders	Size
☑ C:\FMEData2014\Data\Bounda...VancouverNeighborhoods.kml	297 KB
☐ C:\FMEData2014\Data\Orthophotos\02-03-HI.tif	4.6 MB
☐ C:\FMEData2014\Data\Orthophotos\02-03-JK.tif	4.6 MB
☐ C:\FMEData2014\Data\Orthophotos\02-03-LM.tif	4.6 MB

Uncheck the GeoTIFF files if they are checked because they are already available in the FME Server Repository.

However, you will still need to point the file location from the workspace to the resource location. This is necessary because there is no end-user to set this – for example in a published parameter.

Therefore, click **OK**. A dialog will appear in which you are warned that the file references are not valid. For each of the Orthophoto references, click the […] ellipsis button and browse to the true file location in the resources folder.



The following file references may not be valid on FME Server. Please verify they are correct.

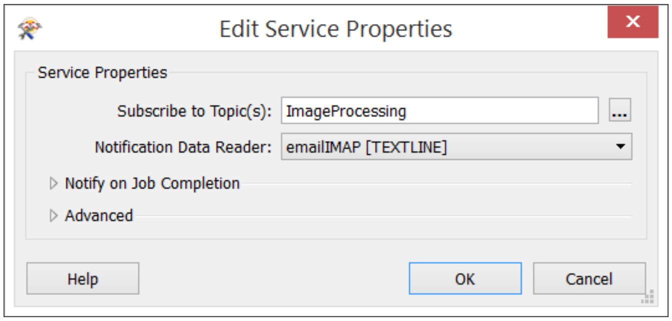
02-03-HI.tif:	Data/Orthophotos/02-03-HI.tif
02-03-JK.tif:	Data/Orthophotos/02-03-JK.tif
04-05-RS.tif:	C:/FMEData2014/Data/Orthophotos/04-05-RS.tif
06-07-HI.tif:	C:/FMEData2014/Data/Orthophotos/06-07-HI.tif
06-07-JK.tif:	C:/FMEData2014/Data/Orthophotos/06-07-JK.tif
06-07-LM.tif:	C:/FMEData2014/Data/Orthophotos/06-07-LM.tif

When a reference is fixed then it will show as pointing to the resources folder; for example *Data/Orthophotos/02-03-HI.tif*

Once complete for all files, click **OK** and then click **Next** to go on to the next publishing stage.

7) Publish to FME Server (Step 2)

In the Services dialog check the Notification Service and click **Edit** to open the Service Properties.



Edit Service Properties

Service Properties

Subscribe to Topic(s): ImageProcessing

Notification Data Reader: emailIMAP [TEXTLINE]

Choose the topic *ImageProcessing* you created in the previous exercise as the Subscribed Topic.

The Notification Reader should be set to the Reader (emailIMAP[TEXTLINE]).

This way the content of any incoming email will be passed directly to that Reader.

Click **OK** and then **Publish** to complete the wizard.

8) Send an Email to FME Server

As before, use any method you like to send an email to your FME Server.

Use the email address you used in Exercise 4a/4b. Enter either Downtown or West End as the subject of the email.

9) Confirm Workspace Triggered

Again, if you are using IMAP it make may take a couple of minutes until FME Server is notified.

Open the Web User Interface and click on the Jobs link and to view the Jobs history. Look for a completed job where FME Server ran Exercise4c.fmw in response to the email. Check the Resources/Data/Output directory for your imagery.

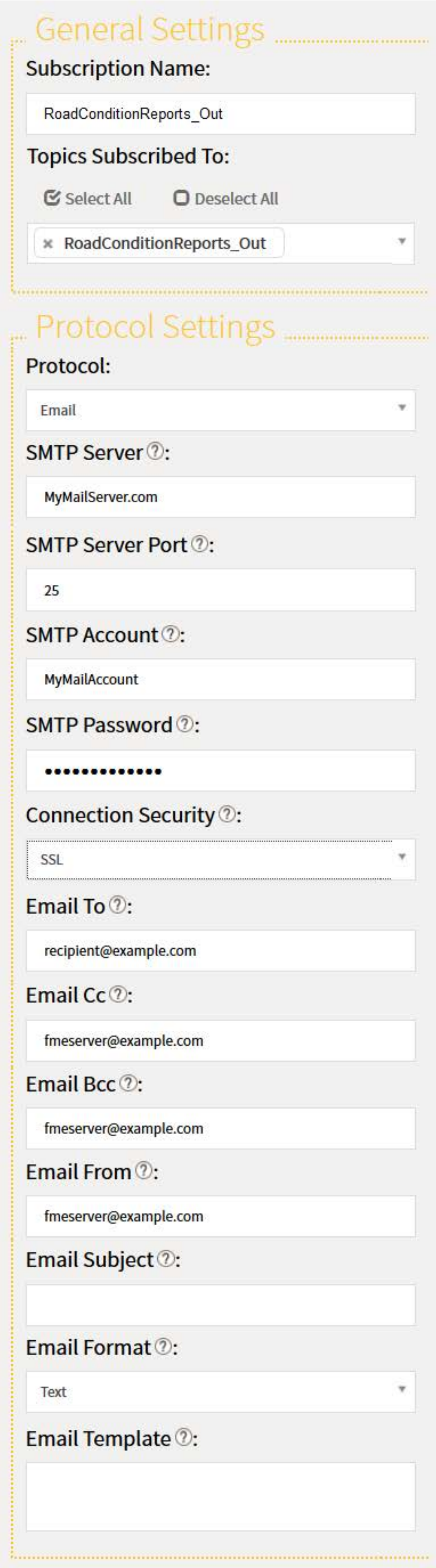
Congratulations! *Now users can email your FME Server with specific information on how to run a workspace - this is truly Self-Serve! In this next exercise you will configure FME Server to email the user back a result.*

Mr. Flibble says …

“Again, you’ll probably want to unsubscribe Exercise4c from the ImageProcessing topic if you’re using IMAP. Unless you want all incoming emails to keep triggering this workspace again and again and again and again and again and again and again …”

Email Subscriptions
Email Subscriptions are when FME Server sends an email in response to a Topic. The in-built mail server in FME Server is only for incoming mail, and so messages need to be sent via an existing SMTP email server.

Setting up an Email Subscription
Creating an Email subscription is done in the Notifications section of the Web User Interface, by choosing the email protocol for a new Subscription.



General Settings

Subscription Name: RoadConditionReports_Out

Topics Subscribed To: ☑ Select All ▢ Deselect All

× RoadConditionReports_Out

Protocol Settings

Protocol: Email

SMTP Server: MyMailServer.com

SMTP Server Port: 25

SMTP Account: MyMailAccount

SMTP Password: •••••••••••••

Connection Security: SSL

Email To: recipient@example.com

Email Cc: fmeserver@example.com

Email Bcc: fmeserver@example.com

Email From: fmeserver@example.com

Email Subject:

Email Format: Text

Email Template:

Here, for example, the author has chosen the Email protocol for a new Subscription.

The Subscription is given a name and an existing topic chosen to be triggered.

There are many more parameters for outgoing mail because the full SMTP server connection parameters need to be defined.

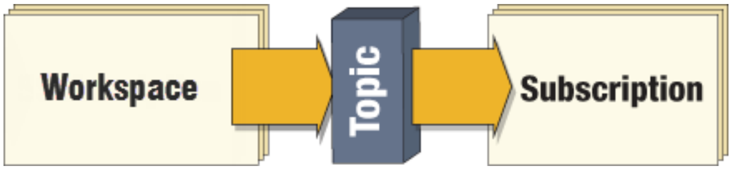
Now when the RoadConditions_Out topic is triggered, an email is sent to recipient@example.com

However, the various fields for the email itself (From, To, Subject, Template) do not need to be hard-coded and can be passed through to the Subscription from a workspace.

One other important parameter is the Email Format. This can either be plain text or HTML.

Workspaces as a Publication
When a workspace acts as a Publication, it triggers a topic when it is run.

In effect, the workspace is publishing to that topic.



In this case the message is constructed in the workspace and passed onto the Subscription.

Publishing notification messages from FME is useful for both sending messages containing content from the data being processed, and for reporting the status of a translation, such as whether it has run successfully or ended in a failure.

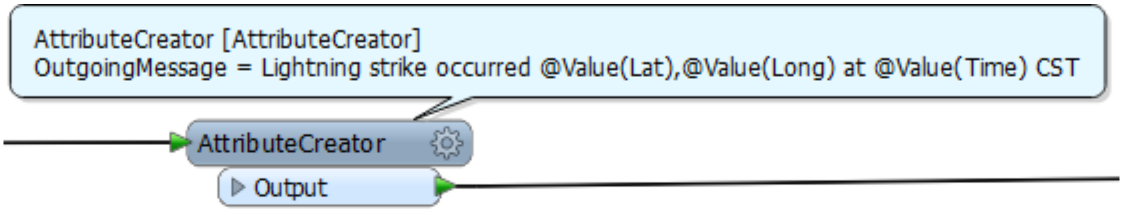
Message Content
The ability to construct message content from a variety of sources – including spatial – is a key reason for using FME workspaces as a publisher. Workspaces can transform data in multiple ways, use it to construct a message, and then dispatch that message to users as a notification.

Content is sent from a workspace by a Writer.

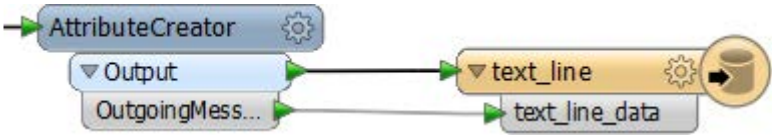
When the workspace is run the notification message is constructed as an attribute and sent to a Writer feature type. The output from that feature type is used by the notification service as the content of the publication.

Content Format
For the purposes of FME notifications, the content of a workspace publication can be in any format. It does not necessarily need to be either JSON or XML, though obviously it would have to be if any client who subscribes to the notifications requires it in that format.

For example, here a workspace constructs a plain-text weather (lightning) alert using an AttributeCreator:


AttributeCreator [AttributeCreator]
OutgoingMessage = Lightning strike occurred @Value(Lat),@Value(Long) at @Value(Time) CST

Now the message attribute is connected to a Text File Writer in order to provide a means for publishing the outgoing message.

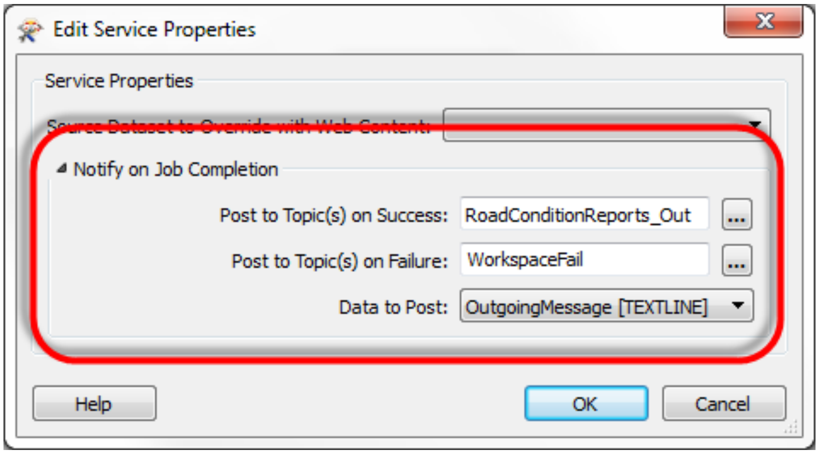


Registering a Workspace

Setting up a workspace to communicate notifications is achieved when the workspace is published to FME Server.

Notifications are sent once the workspace is complete. The way in which the workspace is run is not important. Notifications can be sent for workspaces run using the Job Submitter Service, or the Data Download Service, or any other service.

In other words, a workspace that sends notifications doesn't need to be published to the Notification service; it should be published to the service under which it is to be run. Under the properties for that service will be parameters for setting the notification.



Notify on Job Completion

Post to Topic(s) on Success: RoadConditionReports_Out

Post to Topic(s) on Failure: WorkspaceFail

Data to Post: OutgoingMessage [TEXTLINE]

In this example the author is publishing a workspace to the Job Submitter service and has opened the properties dialog for that service. There are two parameters for controlling the topics – one for success and one for failure – and a parameter for specifying the Writer to use.

Having two Topic parameters is useful because different notifications can be sent depending on whether the workspace succeeds or fails.

In this case, when the workspace runs to completion it will trigger the *RoadConditions_Out* topic. If the workspace fails to complete, it will trigger the *WorkspaceFail* topic.

Either way, the outgoing message is set to be output via a TextLine Writer defined in the workspace.

Ms. Analyst says…

"Four Topics are pre-installed with FME Server to handle workspace notifications. There are two topics for the Job Submitter Service (one for success, one for failure) and two for the Data Download Service (again, one for success, one for failure).

These topics are triggered automatically by FME Server so a system administrator – for example – could subscribe to these topics to receive notifications from all workspaces"

Handling Email Subscriptions in a Workspace

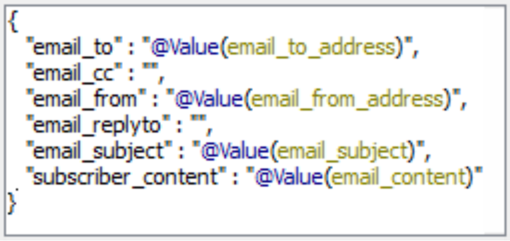
As previously described, content being sent to a notification will be passed out of the workspace using a Writer, usually a plain text Writer that is passed the message in an attribute.

Therefore, to create the content requires only the construction of the attribute with transformers.

For a JSON email you would construct an attribute like this:

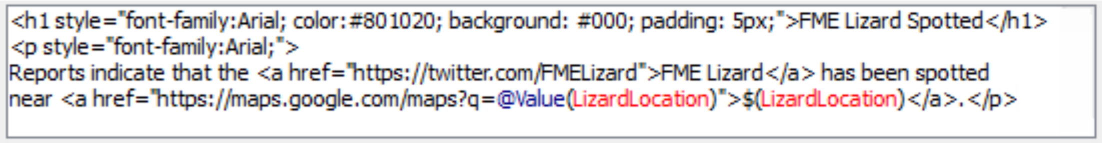
```
{
  "email_to" : "notifications@enduser.com",
  "email_cc" : "",
  "email_from" : "notifications@fmeserver.com",
  "email_replyto" : "",
  "email_subject" : "Notification Subject",
  "subscriber_content" : "Notification Content"
}
```

This could be created using an AttributeCreator transformer, where the email addresses and content can be substituted as required, and would look like this:



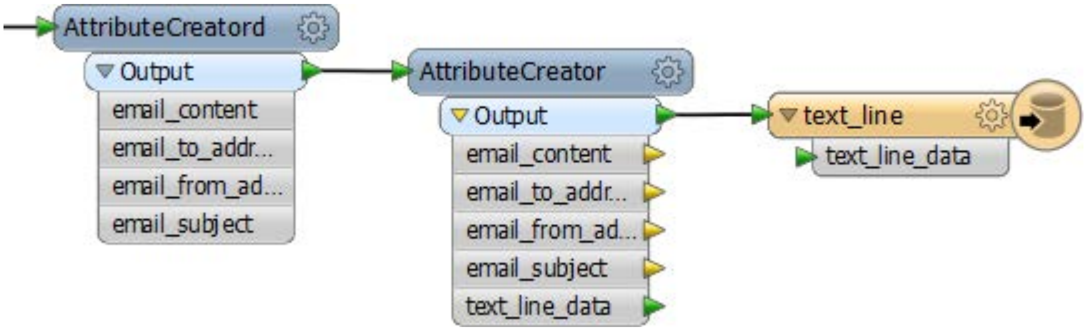
```
{
  "email_to" : "@Value(email_to_address)",
  "email_cc" : "",
  "email_from" : "@Value(email_from_address)",
  "email_replyto" : "",
  "email_subject" : "@Value(email_subject)",
  "subscriber_content" : "@Value(email_content)"
}
```

If the Email format is set to HTML in the Notification GUI, then the email content could be constructed as HTML too, for example:



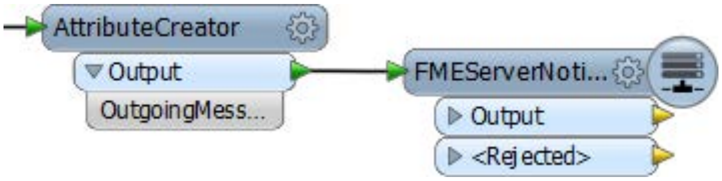
```
<h1 style="font-family:Arial; color:#801020; background: #000; padding: 5px;">FME Lizard Spotted</h1>
<p style="font-family:Arial;">
Reports indicate that the <a href="https://twitter.com/FMELizard">FME Lizard</a> has been spotted
near <a href="https://maps.google.com/maps?q=@Value(LizardLocation)">$(LizardLocation)</a>.</p>
```

The workspace in such a case might look like this:



Publishing with a Transformer

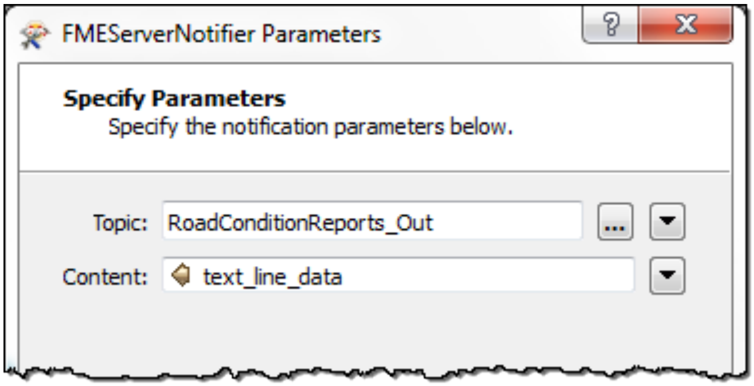
Instead of using a Writer and sending a notification at its completion, a workspace can send a notification through a special transformer called the FMEServerNotifier.



This workspace will trigger the notification wherever it is run.

Notification properties are set in the transformer parameters dialogs (it is a wizard composed of two panes)

The transformer parameters include those for connecting to FME Server, and ones for the topic to post to and the message to be included.


FMEServerNotifier Parameters

Specify Parameters
Specify the notification parameters below.

Topic: RoadConditionReports_Out

Content: text_line_data

Here the author intends to publish the contents of an attribute called *text_line_data* to a topic called *RoadConditionReports_Out*

There are two advantages to issuing a notification this way, over using a Writer:

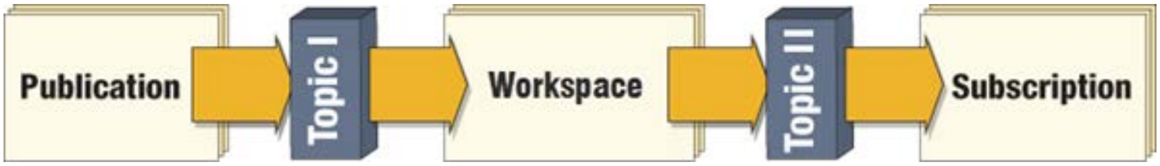
- The workspace can issue a notification *during* a translation, rather than at the end of it.
- The workspace does not need to be run on FME Server to generate an FME Server notification. It will produce the same notification when run using FME Desktop.

Aunt Interop says…

"If you only want to trigger the notification once, then add a Sampler transformer before the FMEServerNotifier and set it up to filter out all but one feature.

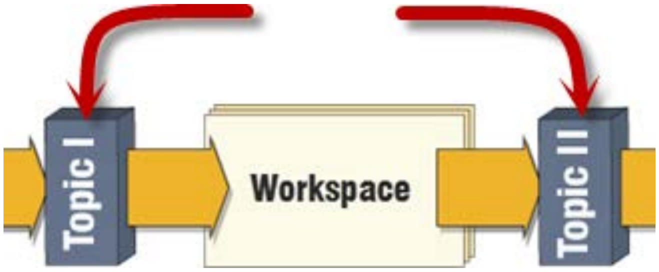
Otherwise a notification is triggered for ***every*** *feature!"*

Workspaces as a Subscription AND a Publication
A system that needs to both receive and send notifications looks like this:



In this scenario the same workspace receives an incoming notification sends an outgoing notification. It is registered with the notification service (in order to receive incoming notifications) and the notification parameters set on a different service (in order to send outgoing notifications).

For example, details of a lightning strike are received via a *LightningStrike* topic. The workspace then processes the information – for example it determines which state/province/county the strike occurred in – creates a new message and dispatches it to subscribers via a *WeatherAlert* topic.



The important thing to notice is that the scenario involves two <u>different</u> topics, here Topic I and Topic II.

A major problem occurs if you set up such a scenario but only use one topic

.

That's because for every notification the workspace receives it will publish back to the same topic, triggering itself again and causing an infinite loop of messages.

Daily Interop Reporter, Chad Pugh-Litzer says …

"Using the same topic for incoming and outgoing notifications is like setting up your email to forward all incoming messages to yourself. You'll end up with a loop that continues until you crash the email system!

To avoid confusion it's a good idea to clarify which topics are for in and out communication through their names. For example, use LighningStrike_In and LightningStrike_Out to differentiate topics that are triggered by incoming alerts and those triggered by outgoing alerts"

Exercise 4d: Email Subscriptions	
Scenario	Airphoto Data Vendor
Data	GeoTiff Orthophotos
Overall Goal	Provide email-driven self-serve access to image files
Demonstrates	Email Subscriptions
Starting Workspace	C:\FMEData2015\Workspaces\ServerAuthoring\Exercise4d-Begin.fmw
Finished Workspace	C:\FMEData2015\Workspaces\ServerAuthoring\Exercise4d-Complete.fmw

Having already set up a solution for incoming email notifications (in Exercise 4b) and used that information to select an area of interest (Exercise 4c) we can now expand upon that to return an email containing the data chosen by the user.

1. Open the Starting Workspace

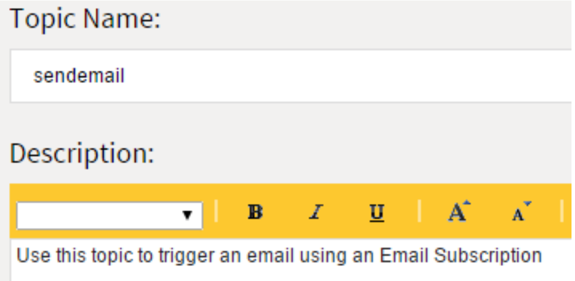
In FME Workbench open the starting workspace for this exercise.

You **must** open this workspace as it is slightly modified from the one in Exercise 4c. A UUIDGenerator transformer and a dataset fanout has been added so that the output JPEG file has a unique name and is written to the FME Server Resources/Data/Output directory.

2. Create an Email Topic

Remember, it's important for a setup like this to use different topics for incoming and outgoing notifications; otherwise the process will end up in an infinite loop.

So open the Web User Interface and click the Notifications link on the Manage menu. In the Topics tab click **New** to create a new Topic. Enter the name *sendemail*. Enter some text in the description to say "Use this topic to trigger an email using an Email Subscription".



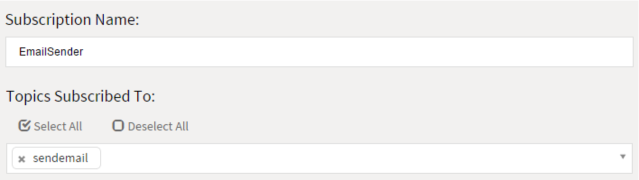
Topic Name: sendemail

Description: Use this topic to trigger an email using an Email Subscription

3. Create an Email Subscription

Continue in the Notifications part of the Web User Interface. Click the **Subscriptions** tab and then click New.

Enter EmailSender as the subscription name and choose the sendemail topic created in step 2.



Subscription Name: EmailSender

Topics Subscribed To: × sendemail

4. Enter Email Settings
Select email as the protocol for the new subscription.

Use the table below to enter your email settings depending on your email service. Please note that if you are using your own internal email system the settings may differ and you may need to ask your IT provider.

	Gmail	Outlook	Yahoo!
SMTP Server	smtp.gmail.com	smtp.live.com	smtp.mail.yahoo.com
SMTP Server Port	465	25	465
SMTP Account	<email address>	<email address>	<email address>
SMTP Password	<password>	<password>	<password>
Connection Security	SSL	TLS	SSL

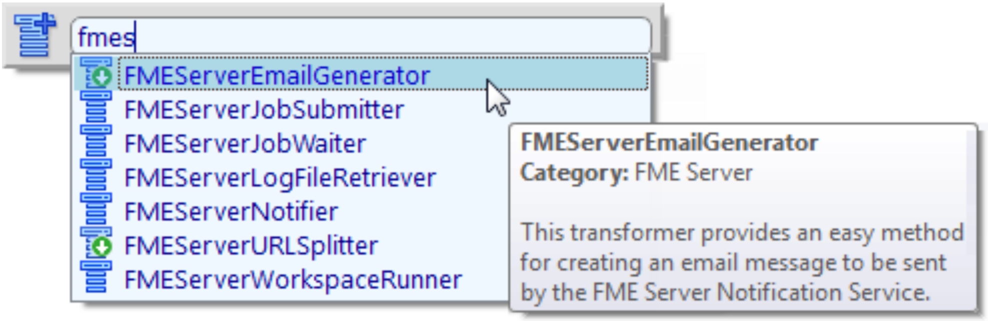
The following settings are same regardless of email service:

Email to:	<leave blank>
Email cc:	<leave blank or enter an admin cc email address>
Email from:	<your email address>
Email Subject:	Your Data is Attached
Email Format:	TEXT
Email Template:	<leave blank>

We can leave some of these fields empty because they will be populated from the workspace when we send the email notification message.

Click OK to accept the settings and create the new Subscription.

5. Add FMEServerEmailGenerator
Back in FME Workbench, click on the workspace canvas and start typing "FMEServer" to locate and add an *FMEServerEmailGenerator* transformer.



FMEServerEmailGenerator
Category: FME Server

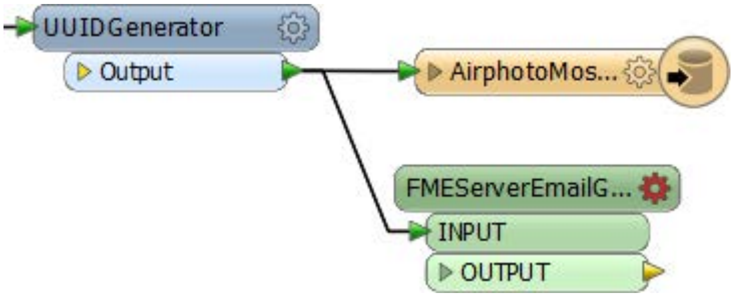
This transformer provides an easy method for creating an email message to be sent by the FME Server Notification Service.

This is a Custom Transformer that will be downloaded from the FME Store – as denoted by the small arrow on its icon in the Quick Add dialog.

The transformer is designed to be a quick and easy way to set up an outgoing email.

6. Connect Custom Transformer

Connect the newly placed custom transformer on a new connection from the UUIDGenerator:



DO NOT connect it between the UUIDGenerator and the Writer feature type.

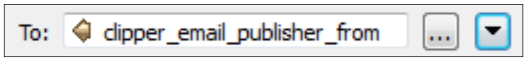
We'll shortly add a new Writer to connect it to.

7. Set Transformer Parameters

Click on the red cog-wheel icon to open the parameters dialog for the FMEServerEmailGenerator transformer. As you'll see, there are many parameters to help us create the JSON message we will be sending to the email Subscription.

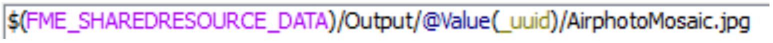
The important parameters should be set as follows:

To: This value depends on whether you are using IMAP or SMTP to send your email to FME Server. Click the drop-down arrow to the right and select Attribute Value > *clipper_email_publisher_from* or *clipper_imap_publisher_from*. This attribute contains the address the incoming email was received from and therefore should be used for who the outgoing email will be sent to.


To: clipper_email_publisher_from

Attachments: Click the drop-down arrow to the right and select Open Text Editor. This value will be slightly more complex than an attribute and we need to construct it with an editor such as this.

Enter the following into the text editor field:
$(FME_SHAREDRESOURCE_DATA)/Output/@Value(_uuid)/AirphotoMosaic.jpg


$(FME_SHAREDRESOURCE_DATA)/Output/@Value(_uuid)/AirphotoMosaic.jpg

This tells FME to use the data created by the JPEG writer as an attachment to the outgoing email message. Click OK to close the dialog.

The other fields can be set as follows:

CC:	blank
From:	blank
Reply to:	blank
Subject:	Your Image Order is Attached
Message:	Please find attached the image you requested.

Note that the blank parameters will come from the Email Subscription we set up in Step 1

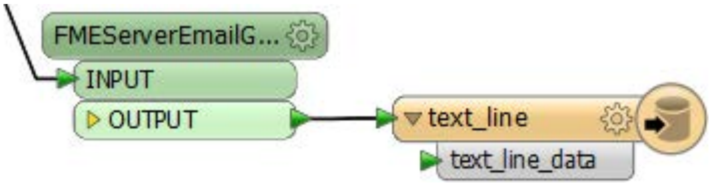
8. Add a Notification Writer

Having created the email to send we now need to add a Writer to pass it to the Notification service.

Select Writers > Add Writer from the menubar. Create a writer with the following parameters:

Writer Format:	Text File
Writer Dataset:	C:\FMEData2015\Output\Email.json

Once added, connect the FMEServerEmailGenerator transformer to the new feature type.



The transformer automatically generates the *text_line_data* attribute required by this Writer.

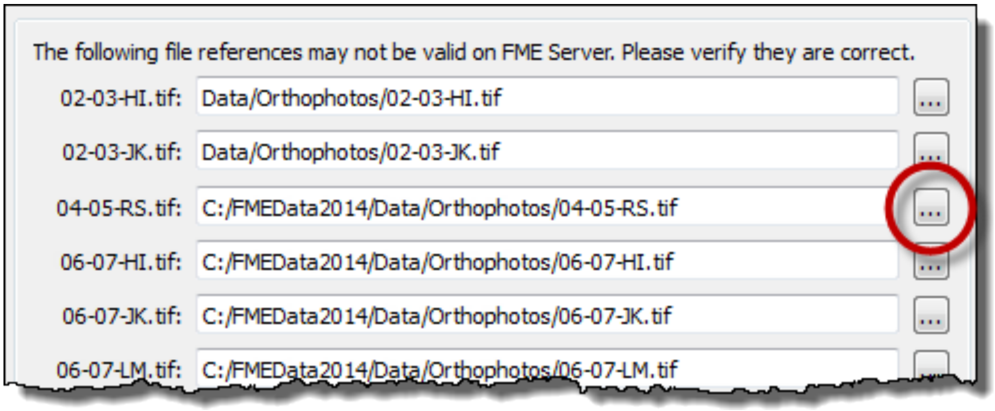
9. Publish the Workspace (Step 1)

This workspace is the heart of a system of incoming and outgoing notification messages; therefore it needs to be published to FME Server as both a Publication and Subscription!

As always, start the Publish Wizard, enter the connection parameters, and choose the Training repository to store the workspace.

As in Exercise 4c, click the **Select Files** button to ensure that the VancouverNeighborhoods.kml file will be uploaded to the repository, and that the image data is not.

Again as in Exercise 4c, click **OK** and then – when warned about invalid references – for each of the Orthophoto references, click the […] ellipsis button and browse to the true file location in the resources folder.

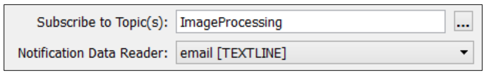


The following file references may not be valid on FME Server. Please verify they are correct.

02-03-HI.tif:	Data/Orthophotos/02-03-HI.tif	...
02-03-JK.tif:	Data/Orthophotos/02-03-JK.tif	...
04-05-RS.tif:	C:/FMEData2014/Data/Orthophotos/04-05-RS.tif	...
06-07-HI.tif:	C:/FMEData2014/Data/Orthophotos/06-07-HI.tif	...
06-07-JK.tif:	C:/FMEData2014/Data/Orthophotos/06-07-JK.tif	...
06-07-LM.tif:	C:/FMEData2014/Data/Orthophotos/06-07-LM.tif	

10. Publish the Workspace (Step 2)

In the Services dialog check the Notification Service and click its Edit button.



Subscribe to Topic(s): ImageProcessing

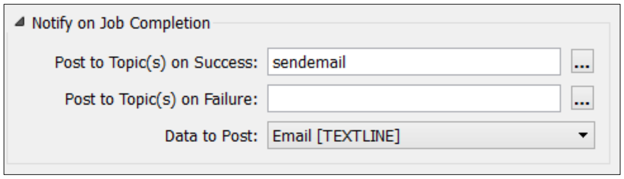
Notification Data Reader: email [TEXTLINE]

Again select *ImageProcessing* as the Subscribed Topic, and set the Notification Reader to the email[TEXTLINE] **Reader**.

In the same Edit Service Properties dialog scroll down and expand the Notify on Job Completion settings.

Set Topics to Publish (Success) to the *sendemail* topic you previously created.

Set the Notification Writer should be set to your Email[TEXTLINE] **Writer**.



Notify on Job Completion

Post to Topic(s) on Success: sendemail

Post to Topic(s) on Failure:

Data to Post: Email [TEXTLINE]

Leave Topics to Publish (Failure) blank, but notice this is where you could handle a failure in workspace if you wanted to.

Click **OK** and then **Publish** to complete the wizard.

11. Send an Email to FME Server

Again send an email to your FME Server using the email address you have been using, and entering either *Downtown* or *West End* as the subject of the email.

After a minute or two you will receive an email back with the image attached. The image will be an orthophoto clipped to the neighbourhood you entered in the subject line.

Congratulations! *Without writing any code you have set-up an email ordering system for imagery.*

Mr. Flibble says …

“Exercise4d. As with 4b and 4c, unsubscribe it from the notification if you are using IMAP and don’t want all incoming emails to keep triggering the workspace.”

Notifications Protocols

Notification Protocols

Available Protocols
Directory Watch Notifications
Mobile Notifications

Different notification protocols need different parameters and set up.

Available Protocols

A protocol is a method of communication. There are many different protocols available in FME Server; some of them are only for use on Publications, some are only for Subscriptions, and some of them can be used with both notification types.

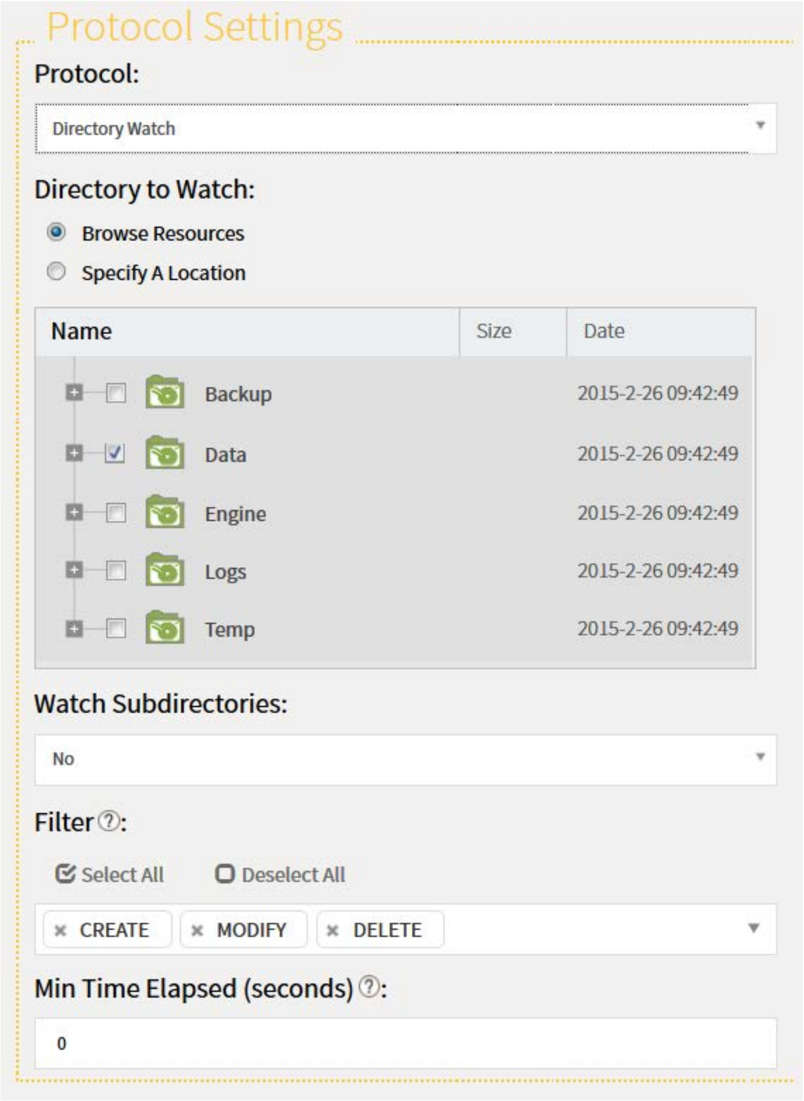
This table lists the different Publication and Subscription protocols and the following pages go into detail on some of the most important, excluding email, which we have already covered.

Protocol	**Description**	**Publications**	**Subscriptions**
Amazon SNS	Communication w/ Amazon's Simple Notification Service	✓	✓
Amazon SQS	Communication w/ Amazon's Simple Queue Service	✓	✓
Amazon S3	Communication w/ Amazon's Simple Storage Service	✓ NEW	✓
Apple Push Notification	Sending Alerts to Apple mobile devices		✓
Directory Watch	Monitoring a folder for new files as a trigger mechanism	✓	
Email (SMTP)	Communication via email	✓	✓
FME Workspaces	Publication via a transformer. Subscription via Job Submitter service	✓	✓
FTP	Writing to an FTP site		✓
Google Cloud Messaging	Sending Alerts to Android mobile devices		✓
Email (IMAP)	Monitoring an email account for new emails	✓	
JMS	Communication with a Java Message Service	✓	✓
Logger	Output to a simple log file		✓
Push	Pushing notifications via HTTP requests	✓	✓
UDP	Communication via a User Datagram Protocol port	✓	
WebSockets	Communication via a WebSockets channel	✓	✓

Directory Watch Notifications
Directory Watch is another Publication-only notification protocol.

Similar to IMAP it monitors for a new object, but instead of email it is monitoring a directory for new files to be added. When a file appears in that directory then the Topic to which it is tied becomes triggered.

The directory being monitored is one of the FME Server resources folders, and this is basically the only parameter you need to set for this notification:



Protocol Settings

Protocol: Directory Watch

Directory to Watch:
● Browse Resources
○ Specify A Location

Name	Size	Date
□ Backup		2015-2-26 09:42:49
☑ Data		2015-2-26 09:42:49
□ Engine		2015-2-26 09:42:49
□ Logs		2015-2-26 09:42:49
□ Temp		2015-2-26 09:42:49

Watch Subdirectories: No

Filter ⓘ:
☑ Select All □ Deselect All
× CREATE × MODIFY × DELETE

Min Time Elapsed (seconds) ⓘ: 0

This Directory Watch notification monitors the *Data* folder for new files.

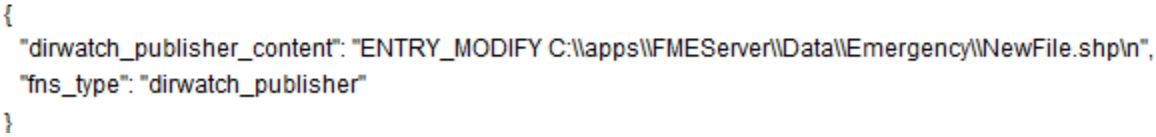
It will not scan subdirectories because the parameter Watch Subdirectories is set to No.

NEW

New for FME2015 is the ability to choose whether to watch for the creation of a new file, or the modification or deletion of an existing file. Also new is the ability to set an elapsed time between notifications. You would want to make sure this setting is longer than the time taken to copy large files onto the file system; otherwise you'll get multiple notifications for the same file update!

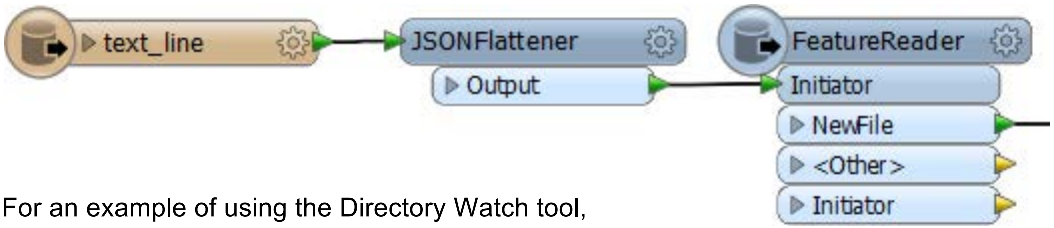
Because FME Server resource folders can be accessed through the file system, it’s possible to put new files into a folder directly, rather than having to upload them via the Web User Interface.

As usual, the notification will be passed into a workspace through a Reader, often a plain text Reader. The result will be a notification that a new (or updated) file exists, with the name of that file:



```
{
  "dirwatch_publisher_content": "ENTRY_MODIFY C:\\apps\\FMEServer\\Data\\Emergency\\NewFile.shp\n",
  "fns_type": "dirwatch_publisher"
}
```

If the filename is already known then the workspace could just read from it directly (it would also need a textfile Reader to accept the notification, even if the content is not used) or it can extract the file name using a JSON transformer and read content from the file using a FeatureReader transformer:



For an example of using the Directory Watch tool, see the exercise in Chapter 5.

Mobile Notifications

Mobile Notifications are Subscription-only protocols that send notifications to mobile devices. The two protocols available are:

- Apple Push Notification
- Google Cloud Messaging

Apple Push Notifications

The Apple Push Subscriber allows delivery of messages to an Apple iOS device, like an iPhone. Messages are sent from FME Server to the device via a cloud service called the Apple Push Notification service (APNs).

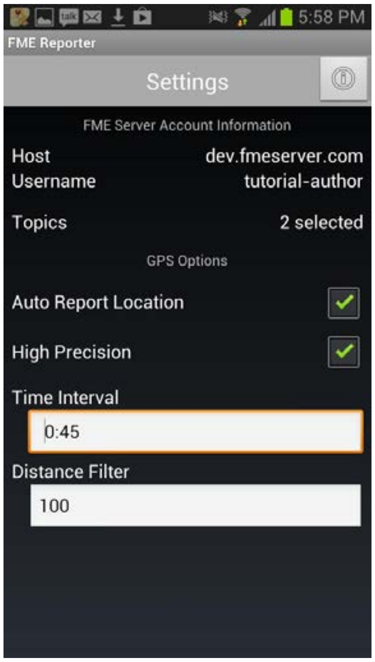
The FME Server Reference Manual explains how to set up such a notification and what is required in terms of Apple "SSL Keystore" and device tokens.

Two free FME apps on the iTunes store exist to provide a starting point for mobile apps; they are the FME Reporter and FME Alerts.

The FME Reporter is a Publication tool used for sending location from a device's GPS to a notification topic on FME Server. Technically it's not a Subscriber so it won't be using the Apple Push Notification protocol.

FME Alerts is a Subscriber tool for receiving notifications from an FME Server using the Apple Push Notification protocol. Location is also sent to the FME Server to allow spatial-based filtering to occur. For example, you can push alerts on local traffic based on the GPS location.

The source code for both apps is made available from Safe Software, so that they can be used as prototypes for your own custom applications.



5:58 PM

FME Reporter

Settings

FME Server Account Information

Host dev.fmeserver.com

Username tutorial-author

Topics 2 selected

GPS Options

Auto Report Location ☑

High Precision ☑

Time Interval

0:45

Distance Filter

100

Google Cloud Messaging

Google Cloud Messaging is the Android equivalent to Apple notifications. Again they involve sending content to a mobile device from an FME Publication.

There are also two free FME apps on the Google Play store to experiment with and use.

Message Streaming

Message Streaming

What is Message Streaming?
Elements of a Message Stream
Message Streaming Architecture
Exercise 4e: Message Streaming

FME Server Basic Training 2015

Whereas Notifications are one-off messages, Message Streaming involves a continuous flow of information

What is Message Streaming?

Message Streaming is a real-time technique that involves a continuous flow of information.

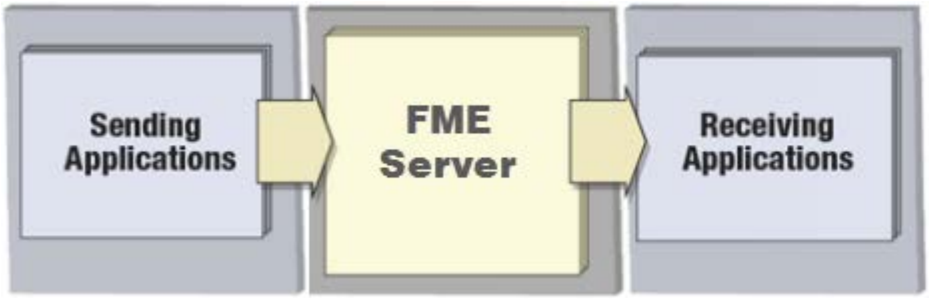
For our purposes, “continuous” means messages arrive at the FME Server at a faster rate than the notification service could handle; say more than one message per second.

At rates faster than this the notification service does not have time to start and run a workspace to process each message. However, a message stream uses a workspace that is constantly running, and doesn’t need to be started for each message, and so can process data at a much faster rate.

When used in this way we call it *High Capacity Message Streaming*, as thousands of messages can be processed every second.

Elements of a Message Stream

Like Notifications, Message Streams can be either into or out of FME, or both:



However, rather than use the Notification service with Readers and Writers, Message Streams are handled by a number of **transformers**. A workspace is set up to run continuously and the transformers within it listen for, and dispatch, messages on a number of different protocols.

Key protocols supported include:

Protocol		**Transformers**
JMS (Java Messaging Service)		JMSReceiver, JMSSender
HTML5 WebSockets		WebSocketReceiver, WebSocketSender
TCP		TCPIPReceiver, TCPIPSender
Twitter API	NEW	TweetStreamer

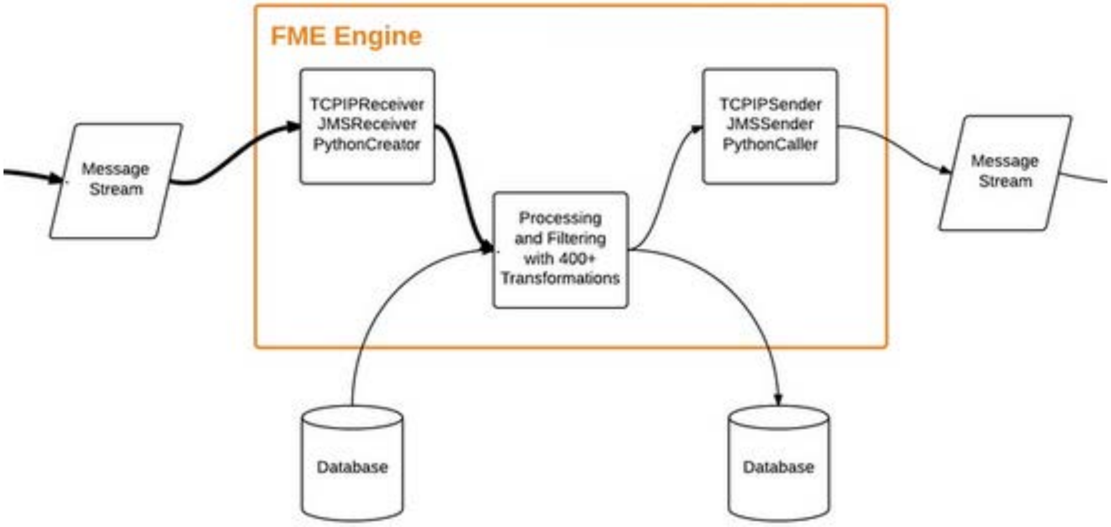
A workspace containing any of the Message Streaming transformers will run continuously because the transformer itself is designed to run continuously.

Receiver transformers will emit a feature in response to an incoming message, but will then return to waiting for more messages. Even if no messages are instantly available, it will not terminate.

Sender transformers will send a message in response to an incoming feature, but will then return to wait for more features. Again it will not terminate until the workspace is deliberately stopped.

Message Streaming Architecture

A Message Streaming architecture that both receives and sends messages looks like this:



- A stream of messages is read into the workspace via one of the available transformers, for example the JMSReceiver.
- Data is processed by any of the available FME transformers, according to the needs of the setup.
- A stream of messages is sent out of the workspace via one of the available transformers, for example the TCPIPSender.

The Database components in this diagram are optional, but probably usual.

Data read from the database is intended to be used to process the incoming message. For example perhaps the message represents a point feature (maybe a vehicle location) that is used to filter database data (maybe traffic conditions) against.

Data written to the database is usually to record a stream of message information. For example, perhaps each incoming message represents a point feature (a lightning strike) that needs to be recorded in a database.

Ms. Analyst says …

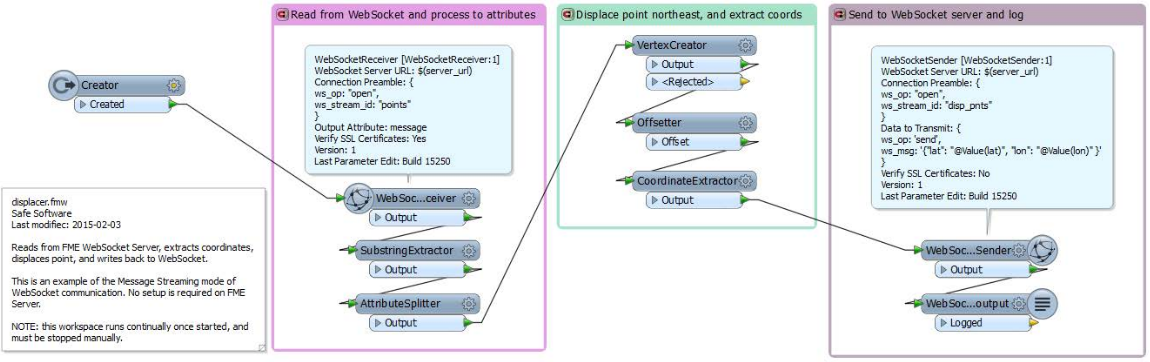
"Writing to a database requires that the transaction interval is set to 1. Otherwise messages won't get committed as they arrive."

Exercise 4e: Message Streaming	
Scenario	Airphoto Data Vendor
Data	GeoTiff Orthophotos
Overall Goal	Web Map Application with WebSockets
Demonstrates	WebSockets
Starting Workspace	C:\FMEData2015\Workspaces\ServerAuthoring\Exercise4e-Begin.fmw C:\FMEData2015\Workspaces\ServerAuthoring\Exercise4e-Begin-Advanced.fmw
Finished Workspace	n/a

This exercise uses FME Server's WebSockets service and WebSockets transformers to power a Web Map application. The method shown here is suitable for high frequency messaging in excess of 1 message per second.

1. Open the Starting Workspace

In FME Workbench open the starting workspace Exercise4e-Begin.fmw



displacer.fmw
Safe Software
Last modifiec: 2015-02-03

Reads from FME WebSocket Server, extracts coordinates, displaces point, and writes back to WebSocket.

This is an example of the Message Streaming mode of WebSocket communication. No setup is required on FME Server.

NOTE: this workspace runs continually once started, and must be stopped manually.

```
WebSocketReceiver [WebSocketReceiver:1]
WebSocket Server URL: $(server_url)
Connection Preamble: {
ws_op: "open",
ws_stream_id: "points"
}
Output Attribute: message
Verify SSL Certificates: Yes
Version: 1
Last Parameter Edit: Build 15250
```

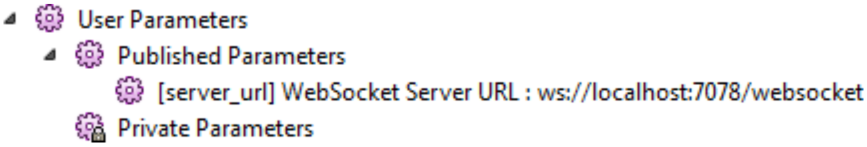
```
WebSocketSender [WebSocketSender:1]
WebSocket Server URL: $(server_url)
Connection Preamble: {
ws_op: "open",
ws_stream_id: "disp_pnts"
}
Data to Transmit: {
ws_op: 'send',
ws_msg: '{"lat": "@Value(lat)", "lon": "@Value(lon)" }'
}
Verify SSL Certificates: No
Version: 1
Last Parameter Edit: Build 15250
```

Notice that it contains two WebSocket transformers – one for receiving and one for sending. The workspace will listen for information coming through the WebSockets channel. Incoming data will be processed with an offset in its coordinates, and then sent back. The workspace will run continuously, processing each incoming message, until stopped manually.

2. Update Server Name

The first task is to update the hostname for the server to match the one you are using.

Locate the published parameter server_url in the Navigator window.



- User Parameters
  - Published Parameters
    - [server_url] WebSocket Server URL : ws://localhost:7078/websocket
  - Private Parameters

Double-click the parameter to open it for editing and replace the host with your own FME Server host name and port. It will need a "ws" prefix to denote the WebSockets protocol.

If you are working on the same machine as your FME Server you can use localhost for example: ws://localhost:7078/websocket

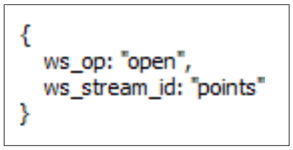
3. Examine the WebSocketReceiver and WebSocketSender Transformers

Open the parameters dialog for the WebSocketReceiver transformer.

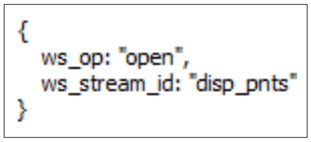
Notice that the Server URL is being obtained from the published parameter you just updated.

Click the […] button to open the Connection Preamble parameter.

What this means is that the transformer will make a connection to the specified FME Server and listen for features coming through a WebSocket stream called “points”.



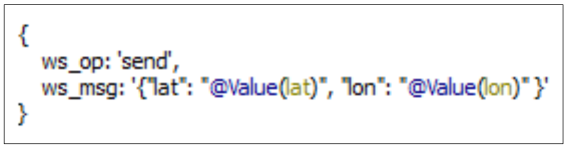
```
{
  ws_op: "open",
  ws_stream_id: "points"
}
```



```
{
  ws_op: "open",
  ws_stream_id: "disp_pnts"
}
```

If you open the parameters for the WebSocketSender transformer, you’ll see it sends features back using a stream id called “disp_pnts”.

The data being sent back through the WebSocket channel is the amended lat/long coordinates of the feature being processed.



```
{
  ws_op: 'send',
  ws_msg: '{"lat": "@Value(lat)", "lon": "@Value(lon)" }'
}
```

4. Start the Workspace

Run the workspace and notice the initial connection messages printed to the log window. Also notice that the workspace keeps running, waiting for messages via the WebSocket stream it is listening to.

5. Edit the Web Map Application

In order to test this setup, a basic web mapping application has been provided in *C:\FMEData2015\Resources\WebSockets*

Some minor updates will be needed before we can use this. In a text editor open the main HTML file *C:\FMEData2015\Resources\WebSockets\www\index.htm*l

Find the two lines below which define the host WebSockets connections for the application. If you are using localhost you can leave this as is, otherwise replace localhost with your own hostname.

```
var sendconn = new WebSocket('ws://localhost:7078/websocket');
var rcvconn = new WebSocket('ws://localhost:7078/websocket');
```

Now find the block of code starting with the comment

```
/*** connect to server input stream ***/
```

Notice that the web application is sending points to FME Server using the stream id “points” – the same one that the workspace is currently listening to.

```
connmsg = '{ ws_op : "open", ws_stream_id : "points" }';
```

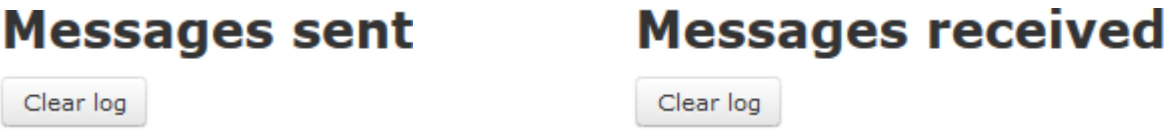
If you look you’ll also find a block of code that listens for data using the stream id “disp_pnts”

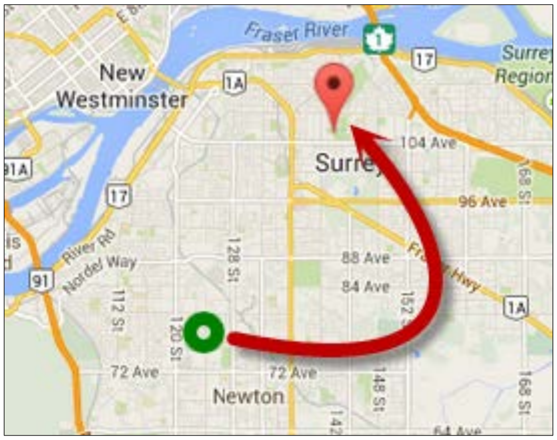
```
connmsg = '{ ws_op : "open", ws_stream_id : "disp_pnts" }';
```

Save the file index.html if you have made any edits.

6. Run the Web Application
Open the index file that we've just inspected in a web browser.

As the Web map loads you will see two message sections logging messages sent to and from the FME Workspace:

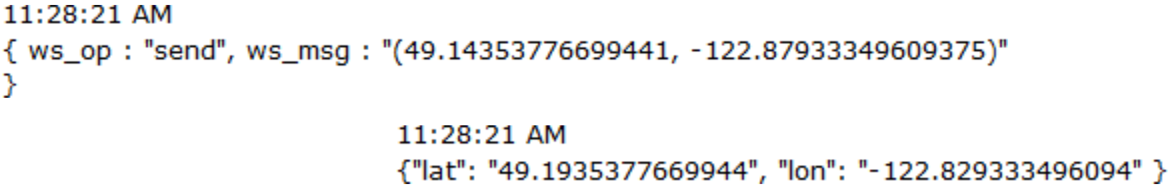




Click on any point of the map.

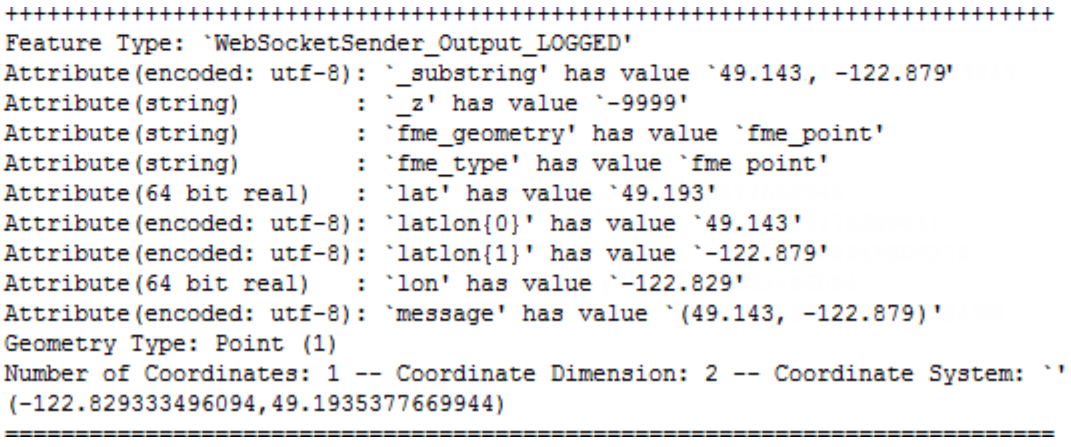
Notice that immediately the point is recorded and returned with a random offset applied (it doesn't matter what happens here, just the fact that something does).

Also notice that messages to and from the workspace are being logged.



```
11:28:21 AM
{ ws_op : "send", ws_msg : "(49.14353776699441, -122.87933349609375)"
}
                                    11:28:21 AM
                                    {"lat": "49.1935377669944", "lon": "-122.829333496094" }
```

If you switch back to the workspace you'll notice that not only is it still running, but a feature was logged as it was received via the WebSocketReceiver, displaced and transmitted out via the WebSocketSender.



```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Feature Type: `WebSocketSender_Output_LOGGED'
Attribute(encoded: utf-8): `_substring' has value `49.143, -122.879'
Attribute(string)        : `_z' has value `-9999'
Attribute(string)        : `fme_geometry' has value `fme_point'
Attribute(string)        : `fme_type' has value `fme_point'
Attribute(64 bit real)   : `lat' has value `49.193'
Attribute(encoded: utf-8): `latlon{0}' has value `49.143'
Attribute(encoded: utf-8): `latlon{1}' has value `-122.879'
Attribute(64 bit real)   : `lon' has value `-122.829'
Attribute(encoded: utf-8): `message' has value `(49.143, -122.879)'
Geometry Type: Point (1)
Number of Coordinates: 1 -- Coordinate Dimension: 2 -- Coordinate System: `'
(-122.829333496094,49.1935377669944)
==========================================================================
```

7. Publish the Workspace to FME Server

Up to now we've just run the workspace on FME Desktop, but it can also be run on FME Server.

Stop the workspace in FME Desktop if it is still running and publish the workspace to FME Server into the Training repository. Simply register the workspace with the Job Submitter Service.

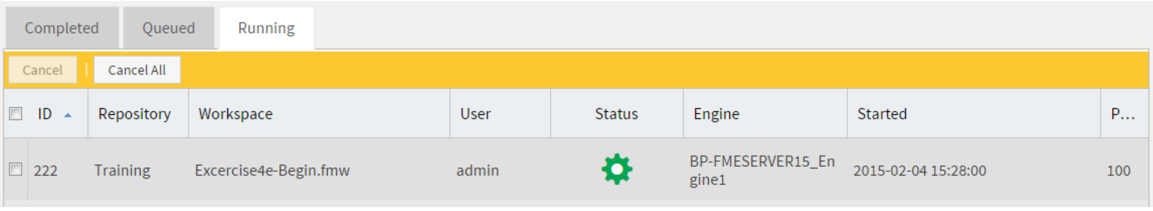
8. Run the WebSockets Workflow on FME Server

In the Web User Interface browse to the workspace in its repository, choose the Job Submitter and click **Run**

The workspace will keep running indefinitely and FME Server's job recovery functionality will restart the job if it ever fails.

Try placing a few more points on the web map application to confirm that points are being displaced. This time the process is being carried out by FME Server.

Go back to the Web User Interface and click on the Manage menu > Jobs and then click the Running tab where you will see your workspace running.



□	ID	Repository	Workspace	User	Status	Engine	Started	P...
□	222	Training	Excercise4e-Begin.fmw	admin	⚙	BP-FMESERVER15_Engine1	2015-02-04 15:28:00	100

Click on the job and then on the Log tab. You'll see the log of the workspace as it is running and can examine it to see your last few map clicks passing through the workspace as features.

NEW

Police Chief Webb-Mapp says …

"The ability to view a log file while the translation is still in progress is a new feature of FME Server 2014"

Advanced Tasks (Exercise 4f)
The use of a continuously running workspace has advantages and disadvantages.

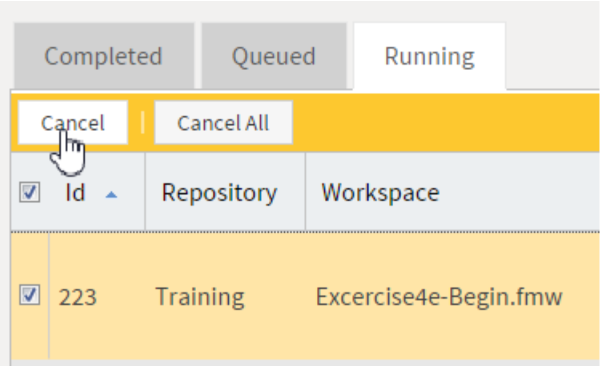
On the plus side, it can process many messages – several thousand – per second. However, it does take up an FME Server engine on a continuous basis too.

If the same workflow was configured to run using the Notification Service it would not use an FME Engine at all times; however it would not be suitable for systems with multiple messages per second.

As an optional task, let's try configuring the system to use notifications instead if WebSockets.

1. Cancel the Job from the Above Exercise

Under Jobs > Running in the Web User Interface, put a checkmark next to the job and press the Cancel button. This will stop the existing job from running, as we no longer need it.



☑	Id	Repository	Workspace
☑	223	Training	Excercise4e-Begin.fmw

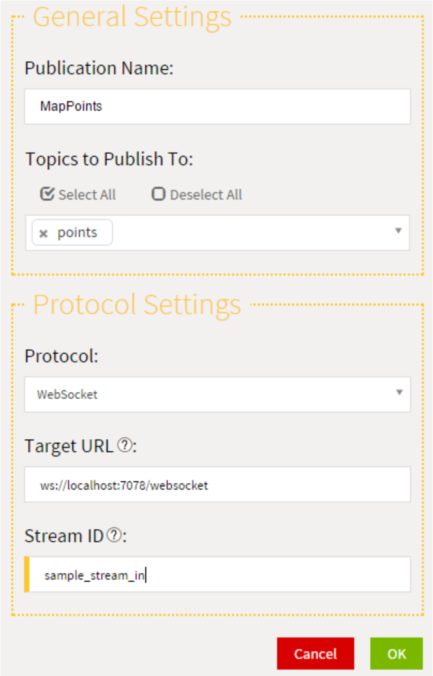
2. Add a Topic

In the Notification part of the Web User Interface, add a Topic named *points*.

3. Add a Web Sockets Publication

Now add a new Publication with the following settings:

Publication Name	MapPoints
Topics to Publish To	points
Protocol	WebSocket
Target Url (or use your own host)	ws://localhost:7078/websocket
Stream ID	sample_stream_in


General Settings

Publication Name: MapPoints

Topics to Publish To:

× points

Protocol Settings

Protocol: WebSocket

Target URL: ws://localhost:7078/websocket

Stream ID: sample_stream_in

4. Edit the Web Map Application

In a text editor open the file *C:\FMEData2015\Resources\WebSockets\www\index.html*

Comment out the default send connection message (line 86) and uncomment the one below:

```
//connmsg = '{ ws_op : "open", ws_stream_id : "points" }';
connmsg = '{ ws_op : "open", ws_stream_id : "sample_stream_in" }';
```

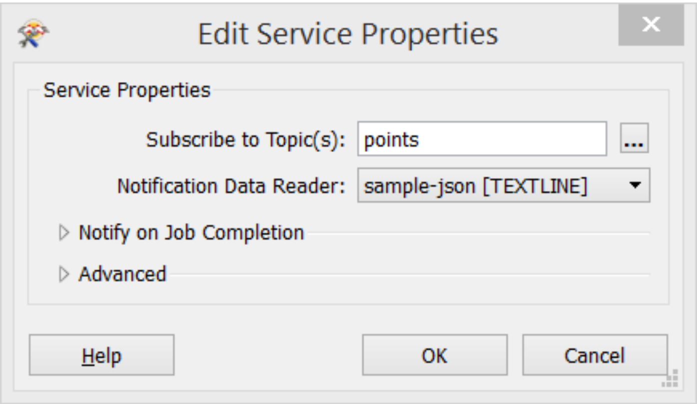
Save the updated file.

5. Open the Advanced Workspace

Open the workspace *Exercise4e-Begin-WebSockets-Advanced.fmw* and edit the server_url parameter if necessary.

Publish this workspace to FME Server and register it with the Notification Service.

In the service settings set the workspace to subscribe to the topic points and ensure the Notification Reader is set to the TEXTLINE reader.



Edit Service Properties

Service Properties

Subscribe to Topic(s): points

Notification Data Reader: sample-json [TEXTLINE]

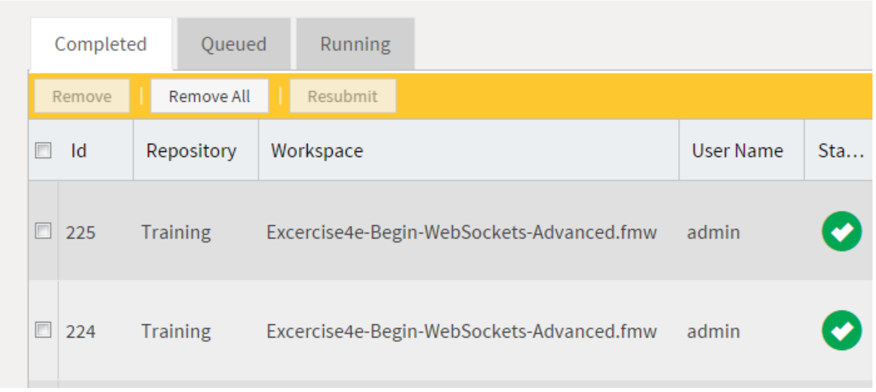
Click **OK** to close the dialog and **Publish** to finish publishing the workspace.

6. Use the Web Application in Notification Mode

Reload the web application in your browser or simply reopen the file *index.html*.

Click some points on the map to ensure they are being processed and offset.

Examine the Jobs history in Web User Interface. You will see a job is run each time you click.



□ Id	Repository	Workspace	User Name	Sta...
□ 225	Training	Excercise4e-Begin-WebSockets-Advanced.fmw	admin	✓
□ 224	Training	Excercise4e-Begin-WebSockets-Advanced.fmw	admin	✓

Troubleshooting for Administrators

Troubleshooting for Administrators

FME Server Fails to Receive Email
FME Server Fails to Send Email
Email Uses Defaults
Cannot Connect to WebSockets Server
Workspace Subscription is Not Triggered by a Topic
Topic Monitoring Fails

This section shows a few basic troubleshooting techniques in case of emergency

FME Server Fails to Receive Email

If you are unable to receive email on FME Server then the following suggestions may be of help.

- Ensure you carry out the post-installation configuration steps as noted in the reference manual. Without this only local mail could be delivered.

- Is the FME Server's email SMTP port open? Typically this will be port 25 or 465.

- Does your FME Server have a DNS name that is available on an internet name server or your local DNS setup? Email cannot be delivered to it if it cannot be found!

- For IMAP, is the FME Server's IMAP port open? Typically this will be port 993.

FME Server Fails to Send Email

If you are unable to send email on FME Server then the following suggestions may be of help.

- Have you created and properly configured the Email Subscriber? The Data Download and Job Submitter services have pre-defined Subscribers, but they will need to be properly configured.

- Is the FME Server's email SMTP port open? Typically this will be port 25 or 465.

- Does the FME Server have an internet connection?

Email Uses Defaults

If email sent by FME Server uses default parameters instead of those you had configured, then the following suggestions may be of help.

- If you are sending an email using the FMEServerNotifier transformer ensure that the Content parameter is set to an attribute that contains the email message information in JSON format (use the custom transformer FMEServerEmailGenerator)

- If you are sending an email from a completed workspace using the Notification Service and Topics to Notify (Success) ensure the Notification Writer is set to the Text File writer which is writing the email information.

Cannot Connect to WebSockets Server

If you cannot connect to a WebSockets server then the following may be of help.

- Ensure the FME Server's WebSockets port (default 7078) is open.
- Ensure you are using the correct *stream_id* for sending and receiving between your applications

Workspace Subscription is not Triggered by Topic

If a workspace set to subscribe to a topic is not triggered by it, then the following suggestions may be of help.

- Check the Push subscription for the workspace in the Web User Interface and make sure the required Topic is in the list of Topics subscribed to.
- Check the Push subscription for the workspace in the Web User Interface and confirm the username and password for the subscription. If you have changed your password recently the password for the subscription needs to be changed as well

Topic Monitoring Fails

If topic monitoring fails then the following may be of help.

- Monitoring requires the use of WebSockets, so ensure the FME Server's WebSockets port (default 7078) is open.

Module Review

Module Review

What you should have learned from this module

FME Server Basic Training 2015

This module introduced you to FME Server's Notification Services.

What You Should Have Learned from this Module

The following are key points to be learned from this module.

Theory

- Real-time systems are message driven and involve small amounts of information. They can be either **Notifications** or **Message Streams**.
- The Notification Service handles individual messages. Topics are keywords that define a notification subject.
- Incoming Notification messages are received from Publishers, outgoing messages are sent to Subscribers
- Protocols are the method of communication (e.g. email, SMS, JMS, etc)
- Message Streams are set up using transformers and involve a workspace that runs continuously receiving/sending messages

FME Skills

- The ability to create Topics, Publications, and Subscriptions
- The ability to use the Email protocol for Publishing and Subscribing
- The ability to create a workspace for handling a message stream

Automation with FME Server

Automation with FME Server

Automation with FME Server

What is Automation?
Scheduling
Automated Data Processing
Workflow Management

FME Server Basic Training 2015

Automation involves workspaces processing data without a user having to manually initiate the process.

What is Automation?

Automation is the ability for FME Server to run a workspace without any user intervention or initiation.

For example, a system administrator sets up a workspace to do a daily update of a corporate database from file-based field updates. The translation can either be set up to start at the same time every weekday, or in response to files being deposited in a folder; and can be set up to run a series of workspaces according to logic inside a control workspace.

These automations are handled in three ways: **Scheduling, Automated Data Processing,** and **Workflow Management.**

Scheduling

Scheduling is the ability to initiate a workspace based on a particular time and/or date.

Schedules can be very simple – for example a one-off translation at a set date/time – or can be more complex and involve setting regular repetition, job priorities, and selection of a particular engine.

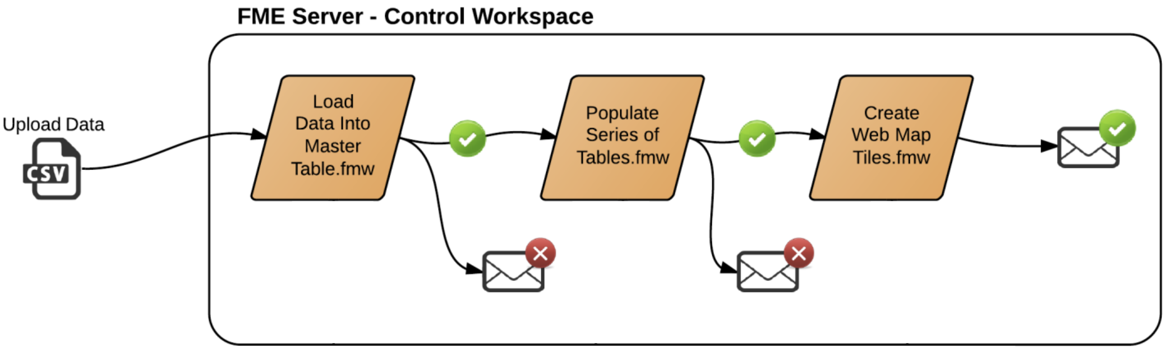
Automated Data Processing

Automated Data Processing is simply starting a workspace in response to an event.

For example, a workspace might be started in response to an email from an administrator, or by connecting FME Server to a cloud automation service such as IFTTT or Zapier.

Workflow Management

Workflow Management is the ability to create a branched translation, where a control workspace automatically determines other workspaces to run based on a set of inbuilt logic.


FME Server - Control Workspace
Upload Data
CSV
Load Data Into Master Table.fmw
Populate Series of Tables.fmw
Create Web Map Tiles.fmw

Scheduling

Scheduling

What is Scheduling?
Managing Scheduled Tasks
Creating a Scheduled Task
Exercise 5a: Scheduling

FME Server Basic Training 2015

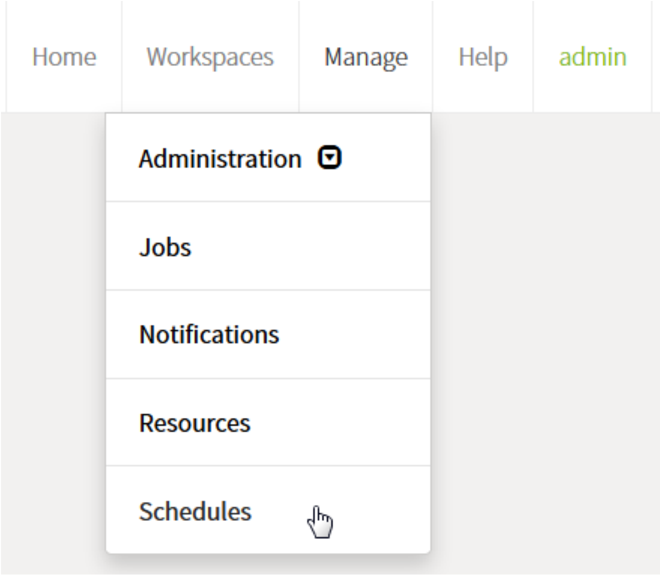
Scheduled Translations are the best way to kick off a workspace at a particular time or date

What is Scheduling?

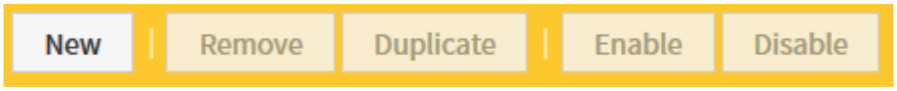
Scheduling is the ability to program FME Server to run a workspace in a repository, at a specific time in the future. The schedule can cause the workspace to run once or on a repeating basis.

Managing Scheduled Tasks

Scheduled tasks are set up in a the Web User Interface. Apart from a quick link on the main page for adding a scheduled task, there is a section under the Manage menu:

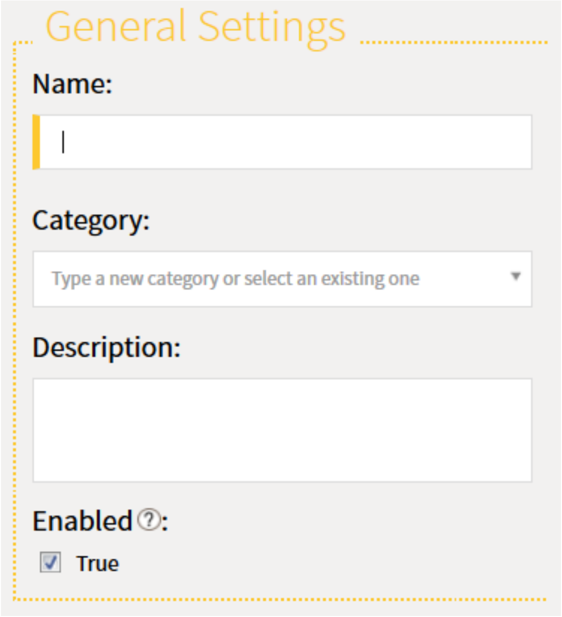


The interface supports all the capabilities you would expect, including the ability to create, delete, copy, enable and disable scheduled tasks:



Creating a Scheduled Task

There are a number of parameters involved in creating a scheduled task



General Settings

Name:

Category:

Type a new category or select an existing one

Description:

Enabled:

☑ True

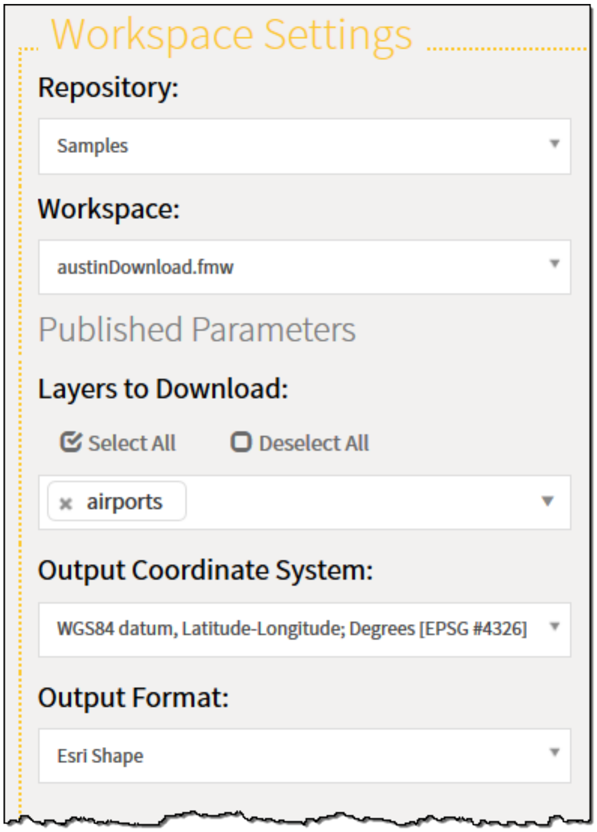
The first parameters are very simple ones for naming and describing the schedule.

Notice that each schedule can be assigned to a particular category.

There is also a parameter for deciding whether the schedule will be enabled straight away.

The next few parameters concentrate on the workspace to be run.

Once a workspace is selected there will be a short pause while FME retrieves information about the workspace. It will then expose any published parameters that exist in the workspace:



Workspace Settings

Repository: Samples

Workspace: austinDownload.fmw

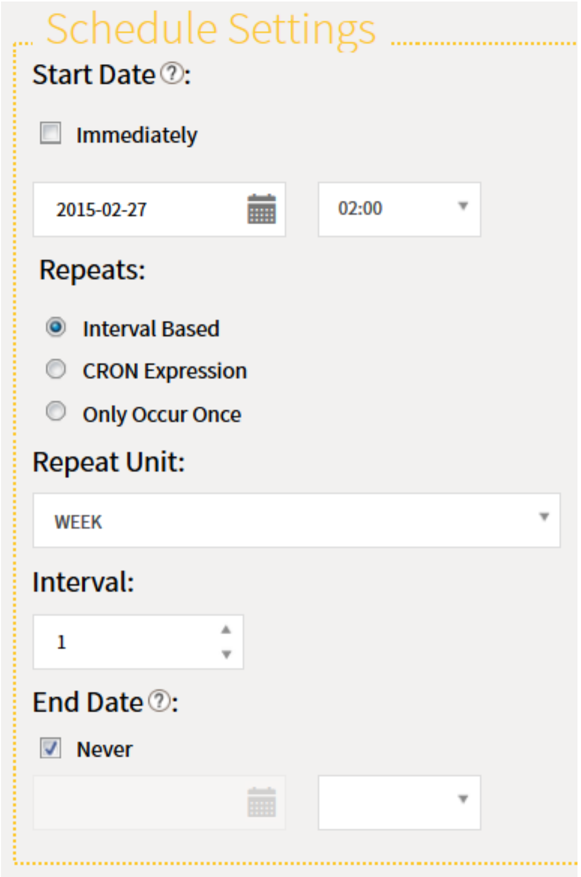
Published Parameters

Layers to Download:

airports

Output Coordinate System: WGS84 datum, Latitude-Longitude; Degrees [EPSG #4326]

Output Format: Esri Shape



Schedule Settings

Start Date:

□ Immediately

2015-02-27 02:00

Repeats:

● Interval Based
○ CRON Expression
○ Only Occur Once

Repeat Unit: WEEK

Interval: 1

End Date:

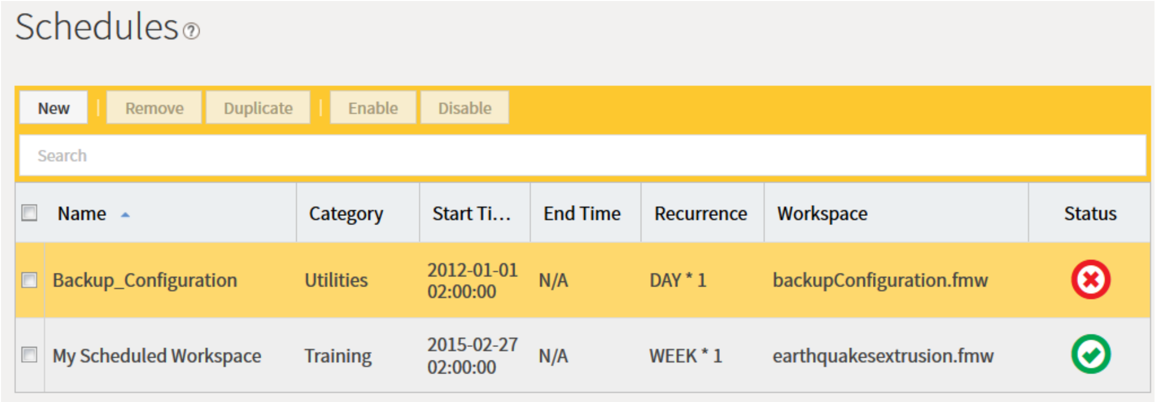
☑ Never

The key parameters, of course, are for setting the actual schedule. Here the workspace is set to run every week starting on the 27th February at 2:00am

There are also optional parameters for notification topics to trigger on completion of the scheduled task. These could be used to inform an administrator of the success or failure of the translation.

Finally there are options to control job priority, job routing (which engine it should use) and job expiry (for jobs that are time-sensitive and would be no longer useful if held back past a certain time by higher priority tasks).

Once the parameters are set for a scheduled task, it is added to the main Scheduling interface.



□	Name	Category	Start Ti...	End Time	Recurrence	Workspace	Status
□	Backup_Configuration	Utilities	2012-01-01 02:00:00	N/A	DAY * 1	backupConfiguration.fmw	✗
□	My Scheduled Workspace	Training	2015-02-27 02:00:00	N/A	WEEK * 1	earthquakesextrusion.fmw	✓

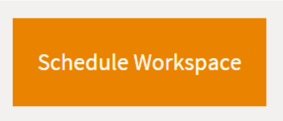
Notice that there are a number of pre-defined jobs, automatically installed with FME Server for the purpose of backing-up and maintaining various components.

Exercise 5a: Scheduling	
Scenario	City of Austin
Data	Apartment Data
Overall Goal	Schedule a Translation
Demonstrates	Automated Scheduling
Starting Workspace	n/a
Finished Workspace	n/a

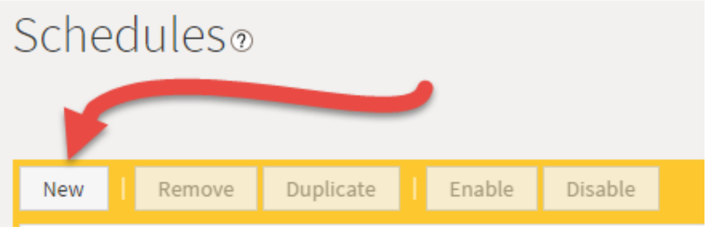
This is a simple exercise to schedule a workspace to translate data.

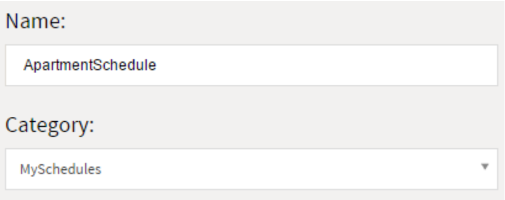
1. Open Web User Interface
Open the FME Server Web User Interface and log in.

2. Select Schedules
Select Schedule Workspace from the homepage.



3. Create Schedule
Click **New** on the schedules menu to start creating a new schedule.

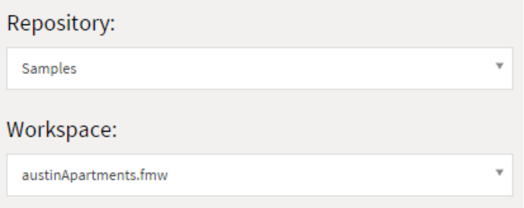



Name: ApartmentSchedule

Category: MySchedules

Enter a name for the schedule (something like *ApartmentSchedule*) and enter a name for a new category (like *MySchedules*)

For the workspace to execute select the repository called Samples and in there select the workspace austinApartments.fmw


Repository: Samples

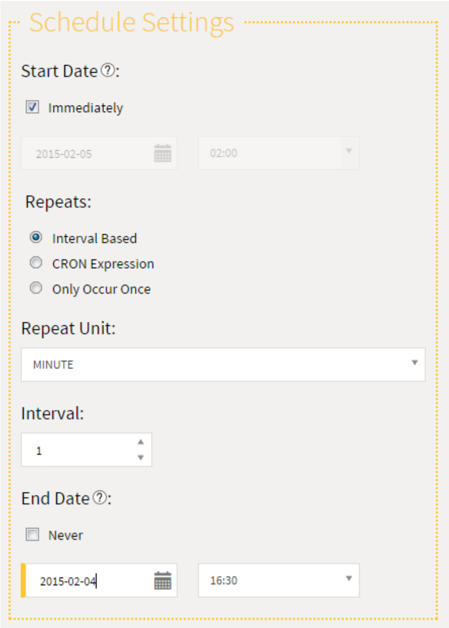
Workspace: austinApartments.fmw

4. Set Schedule
For the schedule itself, use the following settings:

Start date:	Check *Immediately* in order to verify that the schedule works.
Repeats:	Interval Based.
Repeat Unit:	Minute
Interval:	1
End Date:	Choose a date 5 minutes in the future.

Be aware that times are in the 24-hour clock and are in the same time zone as the client.

Click OK to create the schedule.



Schedule Settings

Start Date ⓘ:

☑ Immediately

2015-02-05 02:00

Repeats:

● Interval Based
○ CRON Expression
○ Only Occur Once

Repeat Unit:

MINUTE

Interval:

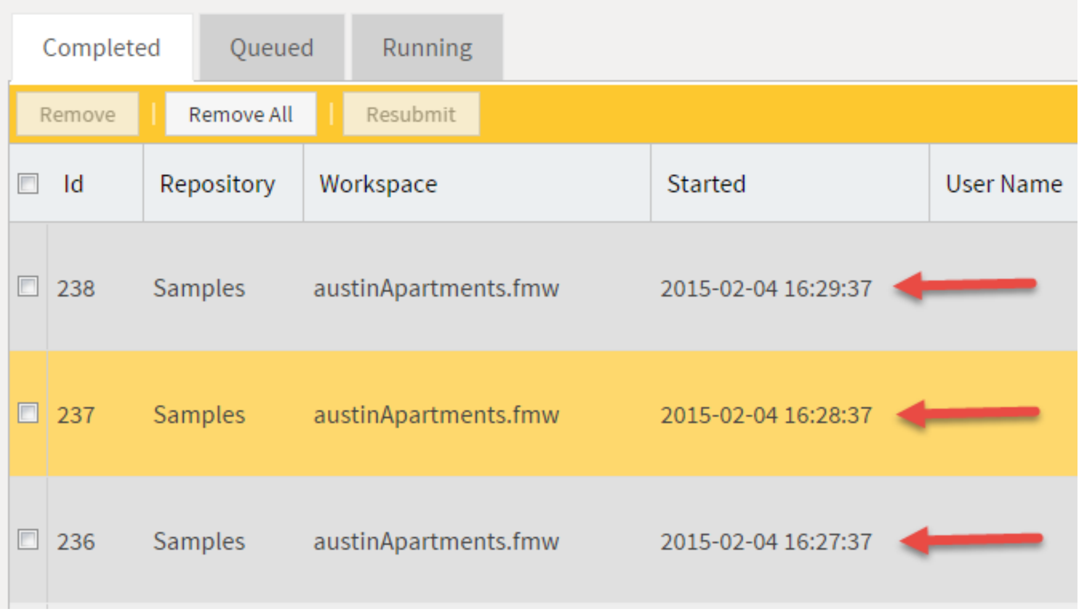
1

End Date ⓘ:

☐ Never

2015-02-04 16:30

5. Check Results
Click on the Jobs menu entry and the Completed tab. You will see the results of the completed scheduled jobs occurring at one minute intervals.



☐	Id	Repository	Workspace	Started	User Name
☐	238	Samples	austinApartments.fmw	2015-02-04 16:29:37	
☐	237	Samples	austinApartments.fmw	2015-02-04 16:28:37	
☐	236	Samples	austinApartments.fmw	2015-02-04 16:27:37	

Advanced Task

As an advanced (and optional) task, why not try to issue a notification for when a workspace fails.

1. Add Topic

Under the Notification dialogs add a new Topic called *schedule_notification*

2. Add Subscription

Now add a new Subscription called *schedule_notification_email.* It should be an email protocol and it should be subscribed to the *schedule_notification* topic

Use the same email (SMTP) details as you used in the Chapter 4 exercises. The other fields should be as follows:

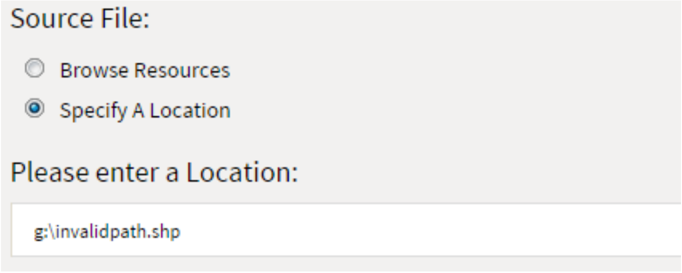
Email To:	<Your own email address>
Email CC:	<Leave empty>
Email Subject:	Schedule Failed
Email Template:	<Leave empty>

3. Create Schedule

Return to the Schedules dialogs. Create a new schedule with a name and category of your choosing. Select the Samples repository and choose the workspace easyTranslator.fmw

When you select a workspace with published parameters – like this one – they will now appear in the Schedules dialog. They can therefore be set at this time.

To force this workspace to fail, we'll deliberately set an incorrect parameter. Locate the Source File parameter and click **Specify A Location**. In the location field enter g:\invalidpath.shp. As this file does not exist it will cause the workspace to fail when run.



Source File:

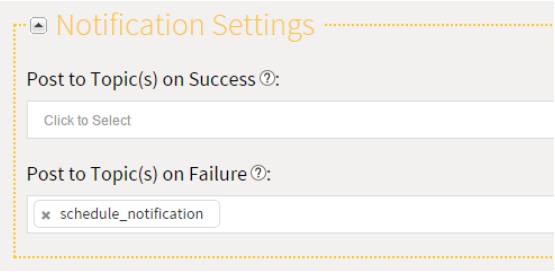
○ Browse Resources

● Specify A Location

Please enter a Location:

g:\invalidpath.shp

Under the Notification settings *Post to Topic(s) on Failure* select the topic *schedule_notification*



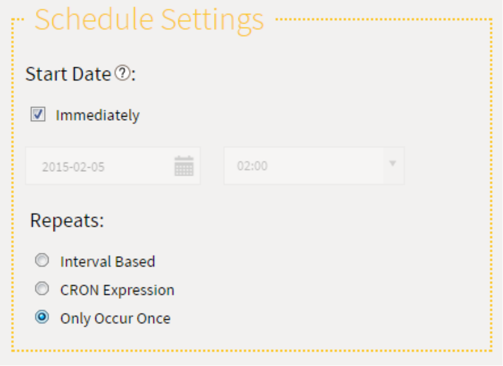
Notification Settings

Post to Topic(s) on Success ⓘ:

Click to Select

Post to Topic(s) on Failure ⓘ:

× schedule_notification


Schedule Settings

Start Date:

☑ Immediately

2015-02-05 02:00

Repeats:

○ Interval Based

○ CRON Expression

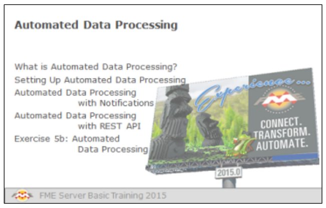
● Only Occur Once

For the schedule itself, set the Start Date to Immediately and the Repeat to Only Occur Once.

We’re only testing this so there’s no need to set up a repeating one.

4. Accept Schedule
Click OK to create the schedule, which should run immediately. The job will fail – as will be confirmed on the Jobs page. You should shortly afterwards receive an email reporting the error.

Automated Data Processing



Automated Data Processing

What is Automated Data Processing?
Setting Up Automated Data Processing
Automated Data Processing with Notifications
Automated Data Processing with REST API
Exercise 5b: Automated Data Processing

FME Server Basic Training 2015

Automated Data Processing is an unscheduled, but automatic, workspace process.

What is Automated Data Processing?

Automated Data Processing is when a workspace is run in response to an event, rather than on a schedule. Because the time of the event cannot be predicted, the workspace is not run at a set time. It is automatic, but unscheduled.

Setting up Automated Data Processing

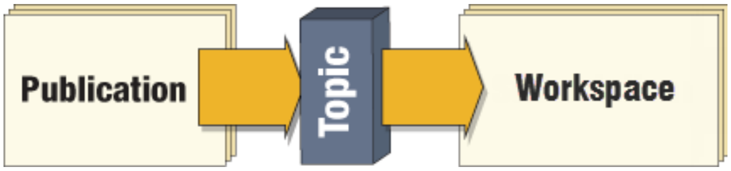
There are two common methods by which to automate FME Server in response to an event:

- Notifications
- REST API

Automated Data Processing with Notificiations

If you've just finished Chapter 4 – which covered notifications – the most obvious way to set up Automated Data Processing will be through a notification.

In this scenario, a Publication is set up to trigger a topic. The workspace is tied to the topic so that it is run as the topic is triggered.



Although the workspace can also send an outgoing notification, here we're concentrating on the idea of starting the workspace, not what it does.

Here we aren't really talking about events that need to be processed; this is more the case that a given event starts a larger processing task.

For example, a system administrator might send an email to FME Server to inform it a daily update of database tables is ready to run. The content of the email isn't important, just the fact that it is triggering a translation.

In this way a workspace can be started through protocols such as Email, Directory Watch, or JMS.

Ms. Analyst says…

"Even if you just want a notification to start a workspace, you'll need a Reader in there to handle that notification.

The Reader can be connected to nothing, and do nothing, but it needs to be there and selected when publishing the workspace."

Automated Data Processing with the REST API

Outside of the FME Server Web User Interface, it's possible to run a workspace using a URL (obtained from the Developer Info) or the REST API.

Therefore, any other application or web service that can issue a HTTP request can be used to automate an FME Server translation.

Database Triggers

A database trigger is an action (or piece of code) that is carried out in response to an event, such as a new record being added or an existing one being modified.

If the trigger were to send an HTTP request to FME Server then it could be made to run a workspace – or other tasks – in response to the database trigger.

For example, Oracle triggers use PL/SQL, which can be made to send a HTTP callout using the UTL_HTTP package. Therefore FME Server workspaces can be executed in response to events that occur on an Oracle database.

Mrs. Dee B. Emess says…

"You can set up a trigger to call FME Server for each individual addition/edit to a record, or you can use a Bulk Mode, which is useful when a large number of edits are made that can be processed by a single workspace.

FMEpedia has some great examples of Single and Bulk processing for Oracle, SQL Server, and PostGres/PostGIS."

Web Automation Services

Web Automation Services are those such as IFTTT and Zapier, which tie together different web-based applications using conditional statements.

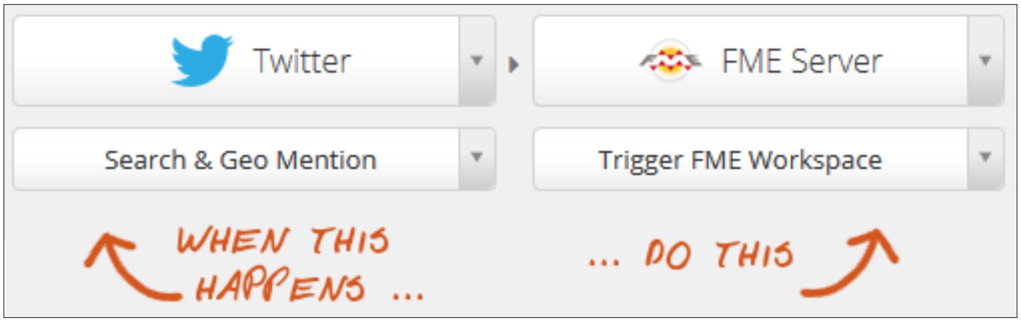
Such services allow you to define a statement using a trigger (the If part) and an action (That).

For example, here is a pre-defined IFTTT "recipe" to send an email when congress is about to vote on a bill.

if
then
Congress is scheduled to vote on a bill
Send me an email

Typically intended to support tools like Gmail, Dropbox, and Facebook; it's also possible to connect FME Server into a workflow like this.

For example, a Zapier "zap" looks like this:

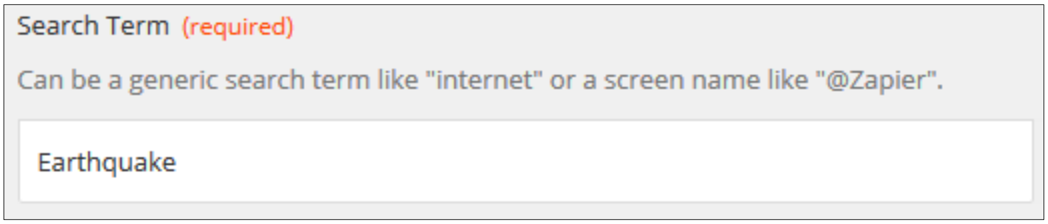


Twitter ▾ ▸ FME Server ▾

Search & Geo Mention ▾ | Trigger FME Workspace ▾

Here, whenever a tweet with a specific keyword and location is discovered, a workspace is run on FME Server.

In this example the search term is for the word "earthquake" and the location is defined (not shown here) as within 50km of the city of Vancouver, BC.

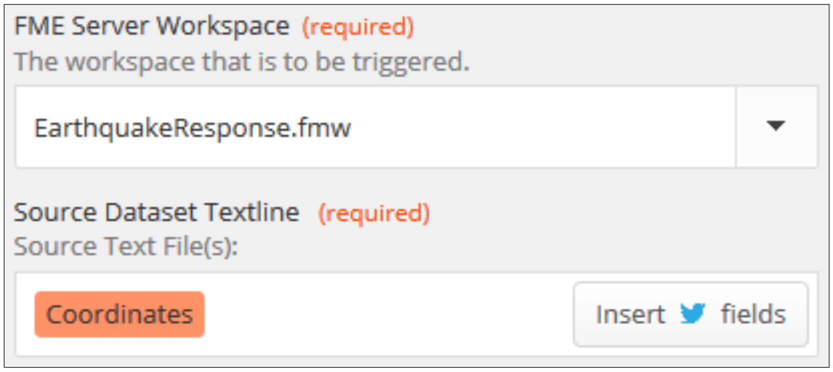


Search Term (required)

Can be a generic search term like "internet" or a screen name like "@Zapier".

Earthquake

The FME workspace being run is called EarthquakeResponse.fmw and the Text File Reader within it is being fed the coordinates of the tweet:



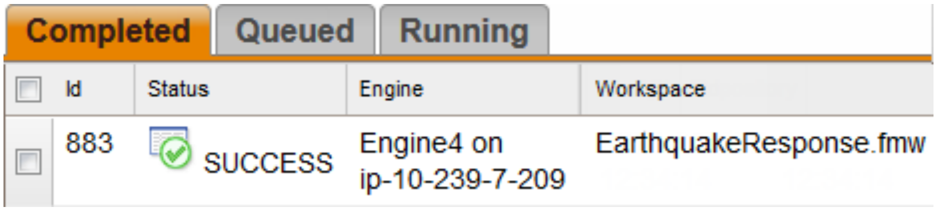
FME Server Workspace (required)
The workspace that is to be triggered.

EarthquakeResponse.fmw ▾

Source Dataset Textline (required)
Source Text File(s):

Coordinates

Now Zapier will check for such a tweet (by default every 15 minutes) and run the workspace on FME Server in response:



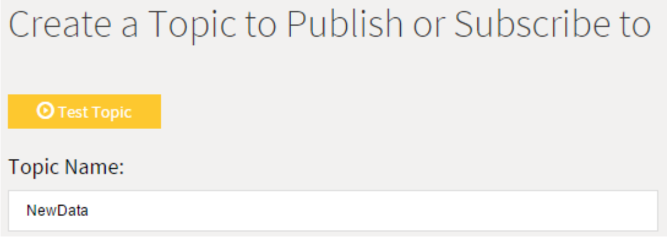
□	Id	Status	Engine	Workspace
□	883	✓ SUCCESS	Engine4 on ip-10-239-7-209	EarthquakeResponse.fmw

Exercise 5b: Automated Data Processing	
Scenario	Airphoto Data Vendor
Data	GeoTIFF Orthophotos
Overall Goal	Run a translation workspace whenever new data is added
Demonstrates	Automated Data Processing
Starting Workspace	C:\FMEData2015\Workspaces\ServerAuthoring\Exercise5b-Begin.fmw
Finished Workspace	C:\FMEData2015\Workspaces\ServerAuthoring\Exercise5b-Complete.fmw

As a member of the Airphoto Data team, you would like to have FME start processing data once an update becomes available, and an email to be sent to alert you of this. Availability of new data will be denoted by new files in a folder. The Directory Watcher protocol can be used to do this.

1. Create New Topic

In the Notifications dialog create a new Topic called *NewData*.



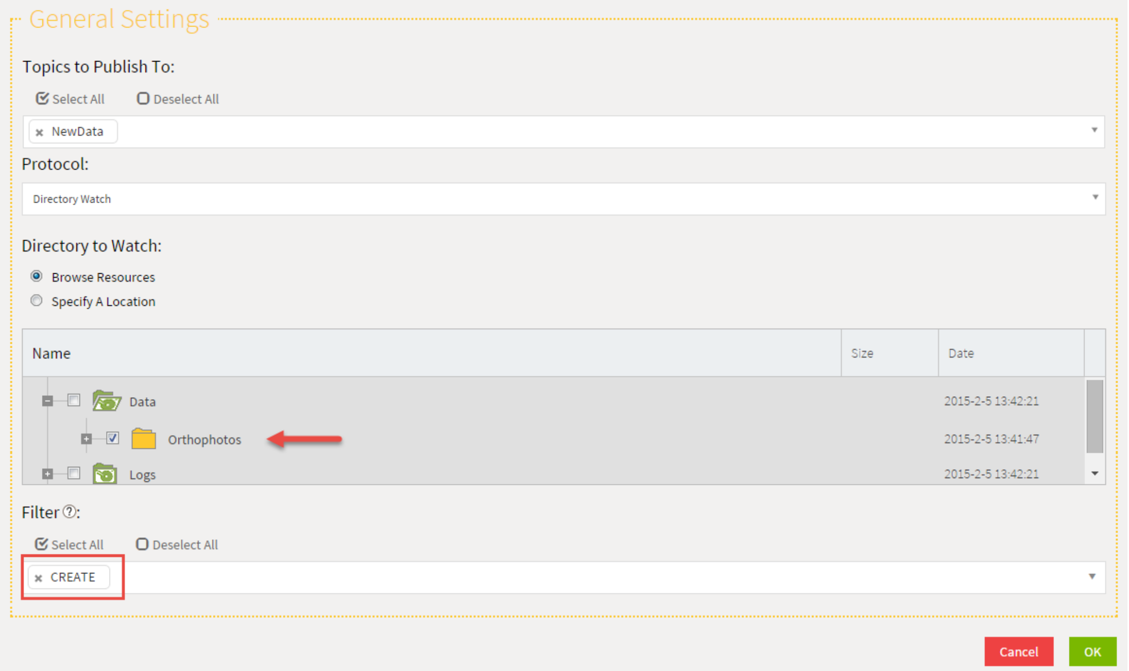
Create a Topic to Publish or Subscribe to

Topic Name: NewData

2. Create a New Publication

Now create a new Publication. Use the following settings:

Publication Name:	New Data Watcher
Topic to Publish to:	NewData (the topic you created in step 1)
Protocol:	Directory Watch
Directory to Watch:	Browse Resources \Orthophotos and check it
Filter:	Create (remove Modify and Delete)



General Settings

Topics to Publish To:

☑ Select All ☐ Deselect All

× NewData

Protocol:

Directory Watch

Directory to Watch:

● Browse Resources

○ Specify A Location

Name	Size	Date
☐ Data		2015-2-5 13:42:21
☑ Orthophotos		2015-2-5 13:41:47
☐ Logs		2015-2-5 13:42:21

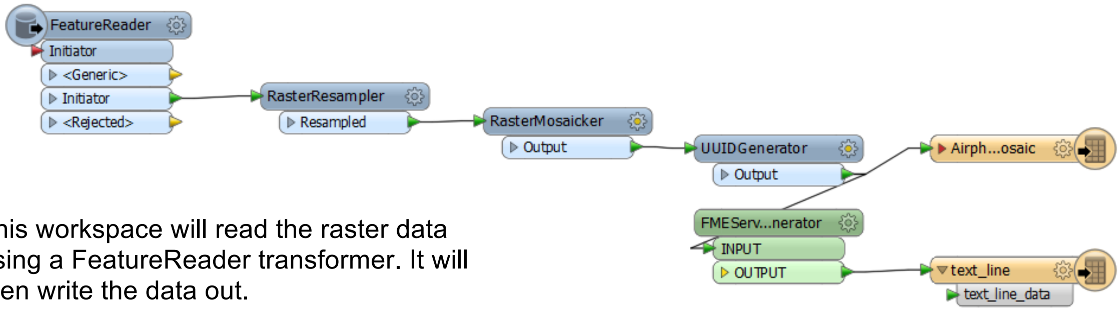
Filter ?:

☑ Select All ☐ Deselect All

× CREATE

3. Open Workspace

In FME Workbench open the starting workspace.



This workspace will read the raster data using a FeatureReader transformer. It will then write the data out.

Notice the workspace also creates an email to send as an output notification.

4. Add a Text File (TEXTLINE) Reader

Select Readers >Add Reader from the menubar to add a Text File Reader. This Reader is necessary to receive the incoming information from the Directory Watcher. Use the settings:

Reader Format:	Text File
Reader Dataset:	C:\FMEData2015\Resources\NewFile.json

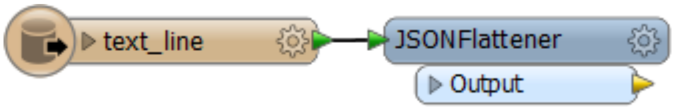
Parameters:

Read Whole File at Once:	Yes

Click **OK** and then **OK** again to add the reader. Don’t connect it to anything in the workspace.

This JSON file is provided as example of what the Notification Service sends to a workspace when the Director Watcher picks up a change to the directory. You can examine the JSON file in a text editor to see how the message is structured.

5. Add JSONFlattener



Add a JSONFlattener transformer and connect it to the Text File feature type.

Edit the properties of the transformer as follows:

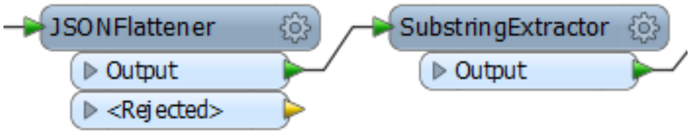
JSON Attribute:	text_line_data
Recursively Flatten Objects/Arrays:	Yes
Prefix New Attribute Names With:	<Leave blank>
Attributes to Expose:	Type in the following two entries onto two lines: fns_type dirwatch_publisher_content

6. Format the File Name for Joining

The attribute *dirwatch_publisher_content* now looks something like this:
ENTRY_CREATE C:\apps\FMEServer\Data\\Orthophotos\10-11-JK.tif\n

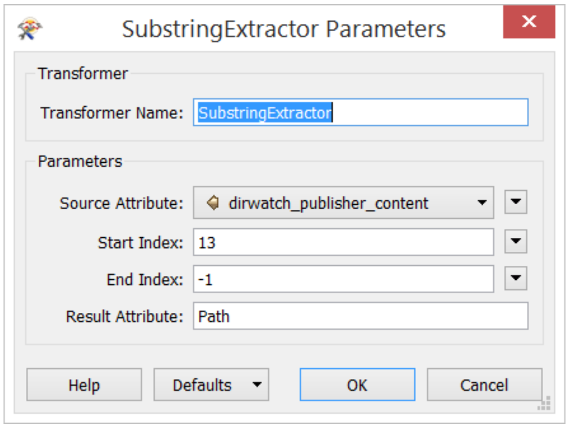
We need to format this attribute in order to extract the file path.

Add a SubstringExtractor transformer, to the Tester Passed port



Open the parameters dialog and set the following:

Source Attribute:	dirwatch_publisher_content
Start Index:	13
End Index:	-1
Result Attribute:	Path


SubstringExtractor Parameters

Transformer

Transformer Name: SubstringExtractor

Parameters

Source Attribute: dirwatch_publisher_content

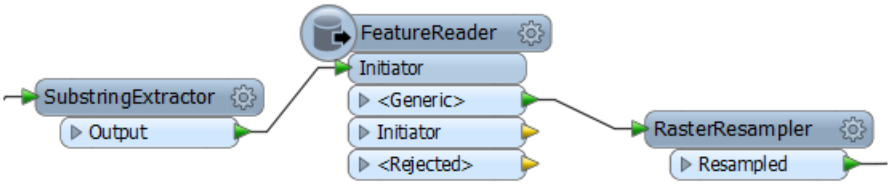
Start Index: 13

End Index: -1

Result Attribute: Path

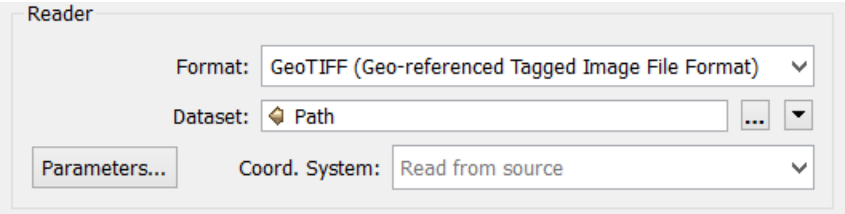
7. Read the New File

Connect the SubstringExtractor transformer to the FeatureReader that already existed in the workspace.



8. Set FeatureReader Parameters

Open the FeatureReader parameters dialog. Ensure that the Reader Format is set to *GeoTIFF* and that the Dataset File parameter is set to the value of the *Path* attribute:



Reader

Format: GeoTIFF (Geo-referenced Tagged Image File Format)

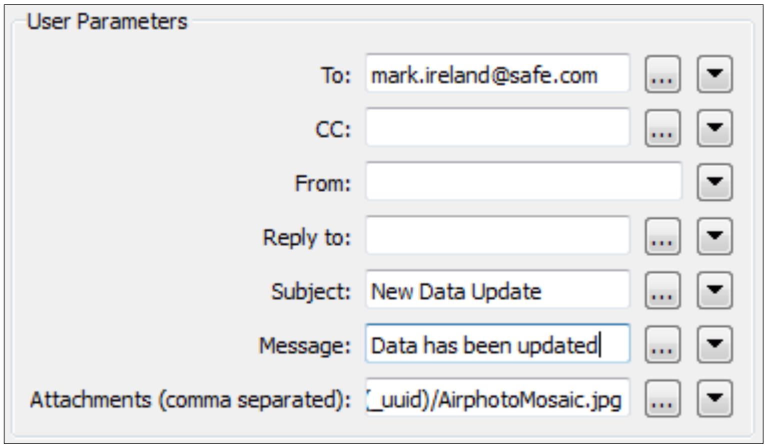
Dataset: Path

Coord. System: Read from source

9. Set the Email Address

In this case the emails will only go to you or another manager so you can set a fixed email address. Open the parameters dialog of the *FMEServerEmailGenerator* and enter your address in the To: field.

Change the subject and message fields as well so they tell you this is new data and not a data order.



User Parameters

To: mark.ireland@safe.com

CC:

From:

Reply to:

Subject: New Data Update

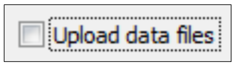
Message: Data has been updated

Attachments (comma separated): _uuid)/AirphotoMosaic.jpg

10. Publish the Workspace (Step 1)

Start to publish the workspace to FME Server. As usual use the same connection parameters and select the Training repository.

Uncheck the Upload Data Files checkbox to ensure that no files will be uploaded to the repository and click **OK** and **Next**.

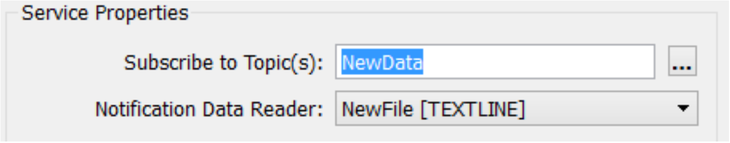


□ Upload data files

11. Publish the Workspace (Step 2)

In the Services dialog check the Notification Service. Open the Service Properties dialog and choose the Topic NewData that was created earlier.

Make sure the Notification Reader is set to your NewFile[TEXTLINE] Reader only.



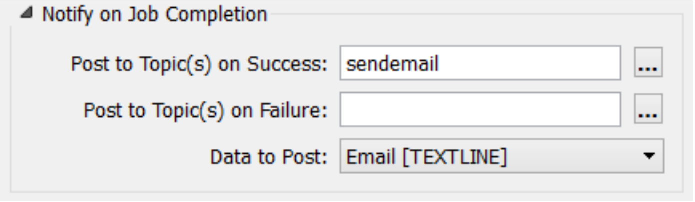
Service Properties

Subscribe to Topic(s): NewData

Notification Data Reader: NewFile [TEXTLINE]

In the same Edit Service Properties dialog expand the Notify on Job Completion settings and set Topics to Publish (Success) to the *sendemail* topic created in a previous exercise.

The Notification Writer should be set to your Email[TEXTLINE] writer.



Notify on Job Completion

Post to Topic(s) on Success: sendemail

Post to Topic(s) on Failure:

Data to Post: Email [TEXTLINE]

Leave Topics to Publish (Failure) blank, but notice this is where you could handle a failure in the workspace if you needed to.

Click **OK** and **Publish** to complete the wizard.

12. Add some New Data

Use the Web User interface to add a new orthophoto file to the Resources folder.

To do so, click the Manage Menu > Resources and expand the Data > Orthophotos folder. Browse to the folder C:\FMEData2015\Data\Orthophotos and select another GeoTIFF file for upload, for example 08-09-PQ.tif

After a few minutes you should receive an email with the resampled image attached.

Congratulations! You have created a solution that automatically processes new data files and alerts users to their presence through an email notification.

Workflow Management

Workflow Management

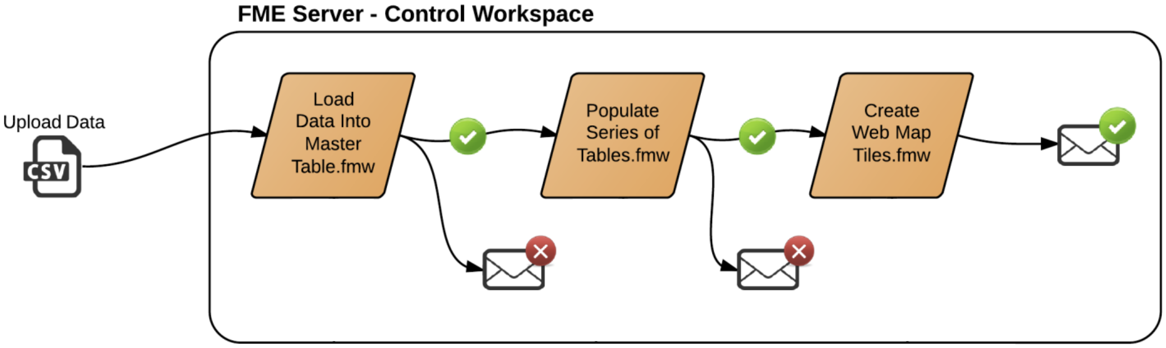
What is Workflow Management?
Implementing Workflow Management
Exercise 5c: Workflow Management

FME Server Basic Training 2015

Workflow Management is great for controlling workspaces in sequence or branching with in-built logic

What is Workflow Management?

Workflow Management is a technique where a set of workspaces can be run in a particular sequence, and where branching can take place to run one workspace or another depending on the result of a set of pre-defined decisions.

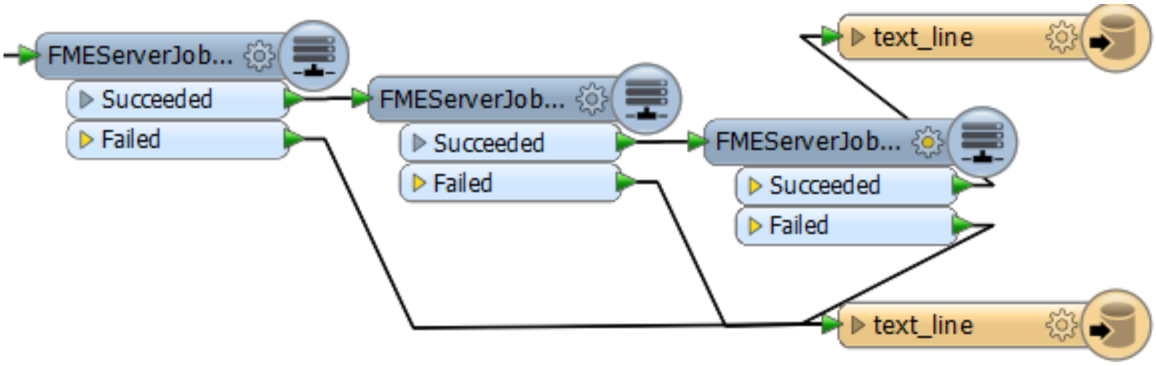


FME Server - Control Workspace
Upload Data
CSV
Load Data Into Master Table.fmw
Populate Series of Tables.fmw
Create Web Map Tiles.fmw

Workflows like this are set up using a control (parent) workspace, which defines the logic for any decisions that needs to be made, and runs subsequent (child) workspaces according to that logic.

Implementing Workflow Management

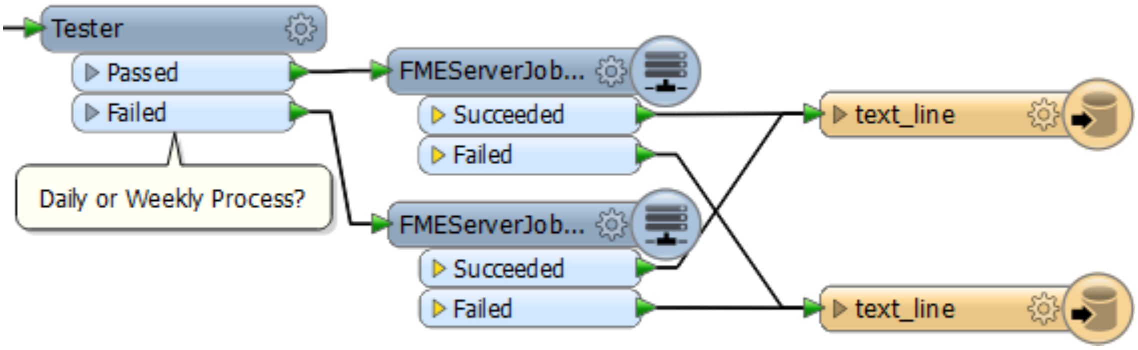
A master workspace calls other workspaces by using one of a number of transformers. For FME Server the most commonly used transformer is the FMEServerJobSubmitter.



Here a control workspace is using the FMEServerJobSubmitter to run three further FME workspaces. Maybe each workspace is a separate step in a database update process.

If a particular task fails then the output is routed to a text file Writer – meaning this could be used in a notification system to send an email to an administrator alerting them to the failure.

If there are several workspaces, but only one of which should be run, then the logic for deciding which one can be made using a Tester, or other filter transformer.



For example, here an organization runs a daily process to upload field updates into a database. However, once a week the process needs to carry out extra actions; such as archiving the source files to Dropbox, or exporting the database contents to Shape files for use by another team.

Each process is defined in a separate workspace, which is called by an FMEServerJobSubmitter transformer.

The Tester transformer prior to the FMEServerJobSubmitters contains the logic over which process is to be carried out. For example it may be testing which day of the week it is.

If it is Monday to Thursday then the test passes and it will run the usual daily update process.
If it is a Friday then the test fails and it will run the weekly update process with the extra tasks.

Again, text file writers are being used to record the result of the translation and whether it was a success or failure. If necessary these could be used to send notifications to a system administrator.

Monsieur D. Server says…

"The Control Workspace may start with a Reader that reads a source dataset, where each source feature triggers the job submitter. For example, the source data may be a list of files that are to be translated.

But more often just a single feature is needed to trigger the child process once. In this case a Creator transformer can be used instead of a Reader."

Exercise 5c: Workflow Management	
Scenario	Airphoto Data Vendor
Data	GeoTiff Orthophotos
Overall Goal	Chain Three Workspaces Together
Demonstrates	Workflow Management
Starting Workspace	C:\FMEData2015\Workspaces\ServerAuthoring\Exercise5c-Begin-Data-Load.fmw C:\FMEData2015\Workspaces\ServerAuthoring\Exercise5c-Begin-Data-Process.fmw C:\FMEData2015\Workspaces\ServerAuthoring\Exercise5c-Begin-Data-Rasterize.fmw
Finished Workspace	C:\FMEData2015\Workspaces\ServerAuthoring\Exercise5c-Complete-Control1.fmw C:\FMEData2015\Workspaces\ServerAuthoring\Exercise5c-Complete-Control2.fmw C:\FMEData2015\Workspaces\ServerAuthoring\Exercise5c-Complete-Control3.fmw

This example shows how to chain together 3 separate workspaces into a complete workflow. The workflow loads data, performs a join and then creates web map tiles in three separate steps.

1. Publish Workspaces

The three workspaces that are to be chained together already exist and need to be published to FME Server. So create a new repository called workflow_management and publish the following three workspaces:

- Exercise5c-Begin-Data-Load.fmw
- Exercise5c-Begin-Data-Process.fmw
- Exercise5c-Begin-Data-Rasterize.fmw

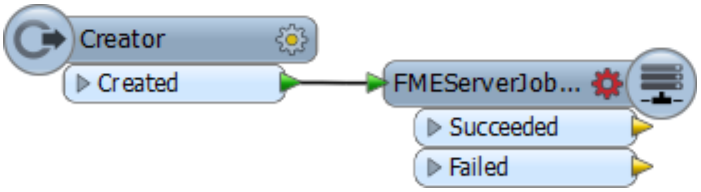
They can simply be published with no resources and registered only with the Job Submitter.

2. Create Control Workspace

Now we can create a control workspace to manage the three child workspaces.

Start with an empty canvas in Workbench and add a Creator transformer. A Creator will simply create a single non-geometry feature that can be used to trigger the rest of the workspace.

Now add an FMEServerJobSubmitter transformer after the Creator:



Save the workspace as *workflow_management_controller.fmw*

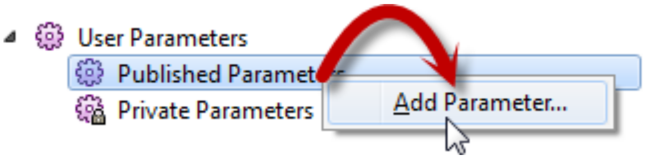
3. Add Published Parameters

Published Parameters are how information can be passed from a master workspace to a child workspace. The child workspace has existing parameters and new ones can be created in the master in order to receive information that is to be passed in.

The first child workspace (for loading data) has two Readers to read two input datasets; therefore we can specify the source data using published parameters.

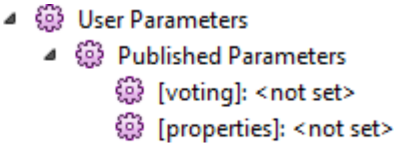
In the Navigator window locate the section named Published Parameters.

Right-click on Published Parameters and choose Add Parameter.



Create two new parameters:

A. Type:	Filename (Existing)		
Name:	voting		
Prompt:	voting:		
Configuration:	Description:	shapefile,	
	Filter:	*.shp	
B. Type:	Filename (Existing)		
Name:	properties		
Prompt:	properties:		
Configuration:	Description:	shapefile,	
	Filter:	*.shp	



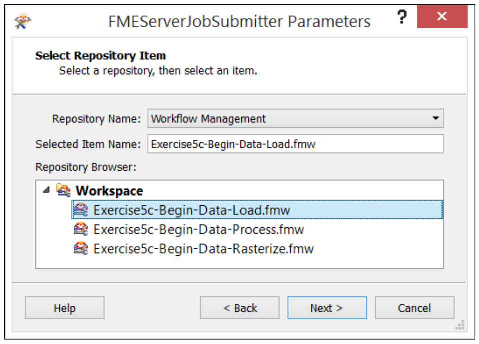
4. Set FMEServerJobSubmitter Parameters

Open the parameters dialog for the FMEServerJobSubmitter. Here we can define the workspace that is to be run by this master process.

Enter the usual connection parameters.

For the workspace to run choose the *workflow_management* repository and from there select the workspace *Exercise5c-Begin-Data-Load.fmw.*

Click **Next** to continue.


FMEServerJobSubmitter Parameters

Select Repository Item
Select a repository, then select an item.

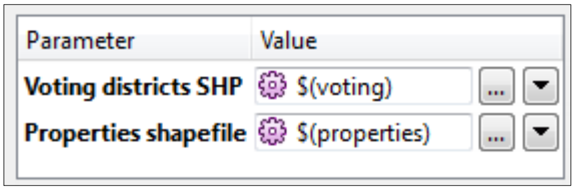
Repository Name: Workflow Management

Selected Item Name: Exercise5c-Begin-Data-Load.fmw

Repository Browser:

- **Workspace**
  - Exercise5c-Begin-Data-Load.fmw
  - Exercise5c-Begin-Data-Process.fmw
  - Exercise5c-Begin-Data-Rasterize.fmw

Set Wait for Server Job to Complete to Yes.

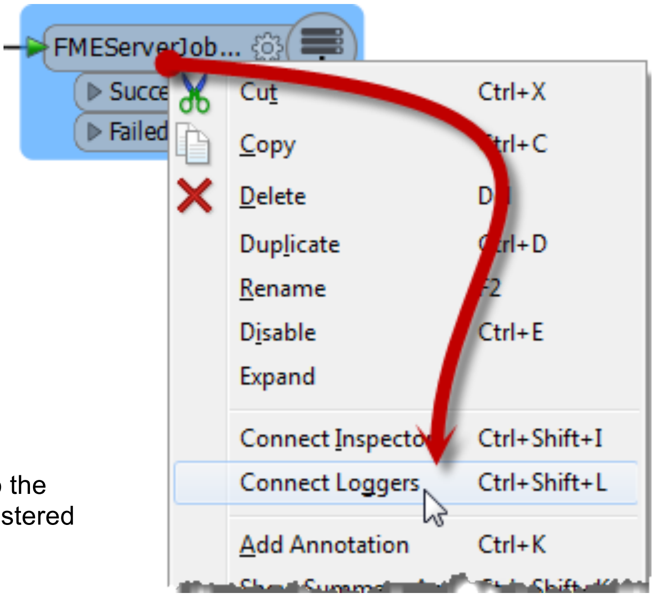


Parameter	Value
Voting districts SHP	$(voting)
Properties shapefile	$(properties)

For the first published parameter, click the drop-down arrow (to the right) and choose User Parameter > voting

For the second, click the drop-down arrow and choose User Parameter > properties

Click **Finish** to close the parameters dialog.



5. Add Logger Transformers
Right click the FMEServerJobSubmitter and select Connect Loggers.

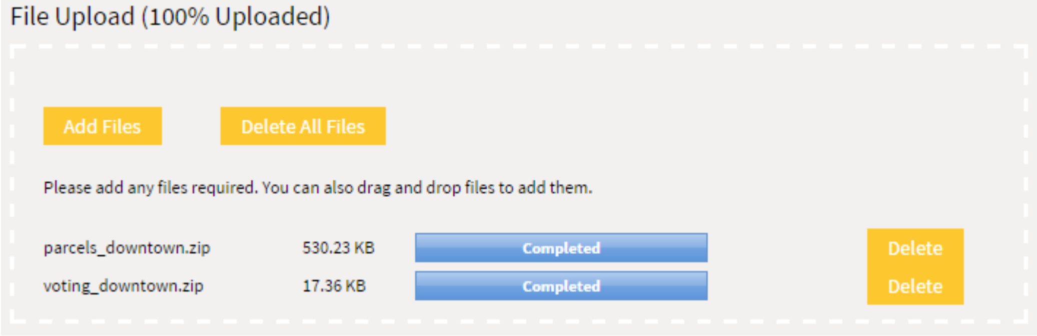
This will add Logger transformers to the workspace so we will know the status of the completed jobs.

6. Publish the Workspace
Publish the workspace to FME Server, into the workflow_management repository and registered with the job_submitter service.

7. Run Workspace
Locate the workspace in its repository in the web interface. Select Job Submitter

Use the File Upload “Add Files” button to upload the files:

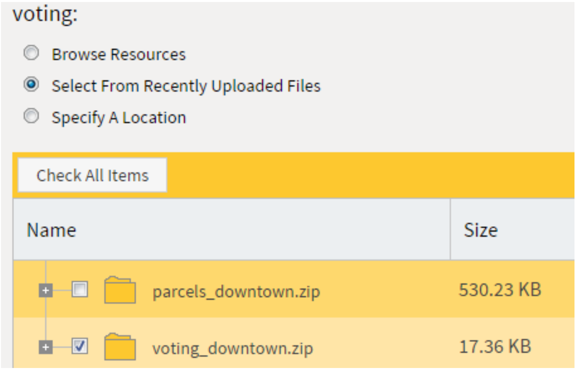
C:\FMEData2015\Resources\workflow_management\parcels_downtown.zip and
C:\FMEData2015\Resources\workflow_management\voting_downtown.zip files



File Upload (100% Uploaded)

Please add any files required. You can also drag and drop files to add them.

parcels_downtown.zip	530.23 KB	Completed	Delete
voting_downtown.zip	17.36 KB	Completed	Delete



voting:

○ Browse Resources
● Select From Recently Uploaded Files
○ Specify A Location

Name	Size
☐ parcels_downtown.zip	530.23 KB
☑ voting_downtown.zip	17.36 KB

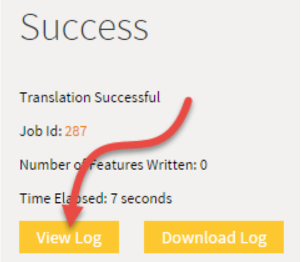
For the two published parameters, first click *From Recently Uploaded Files*.

Then, for the voting parameter, choose the newly uploaded dataset *voting_downtown.zip*

For the properties parameter choose *parcels_downtown.zip*

Click **Run Workspace** to run the workspace.

8. View Log
Once the workspace completes click the View Log button.



Success

Translation Successful

Job Id: 287

Number of Features Written: 0

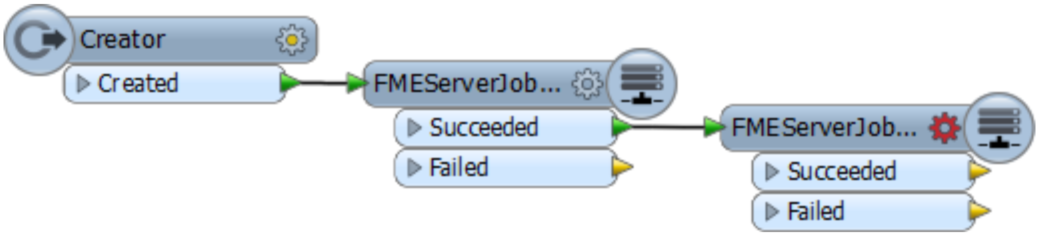
Time Elapsed: 7 seconds

Scroll down the log until you see the logged feature.

If you see the Feature Type of the logged feature reported as FMEServerJobSubmitter_Succeeded_LOGGED then you know the child workspace was successfully triggered.

9. Chain Workspaces – Part 1
Now we can start to chain multiple workspaces together.

Go back to FME Workbench where the control workspace is open. Delete the Logger transformers and then add a second FMEServerJobSubmitter transformer. It should be connected to the Succeeded port of the first FMEServerJobSubmitter transformer.



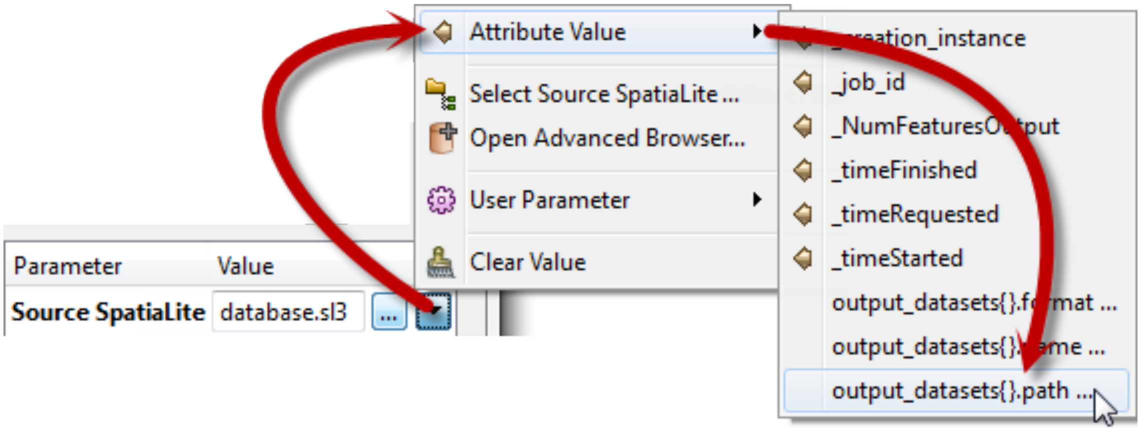
Open the parameters dialog of the newly added FMEServerJobSubmitter and enter the usual connection details. This transformer will be used to execute the second workspace of the three, so select *Exercise5c-Begin-Data-Process.fmw* as the workspace to run.

Again, Set Wait for Server Job to Complete to Yes.

This is where it gets interesting. One of the attributes added to the features output from an FMEServerJobSubmitter transformer records the path to the dataset written by that workspace.

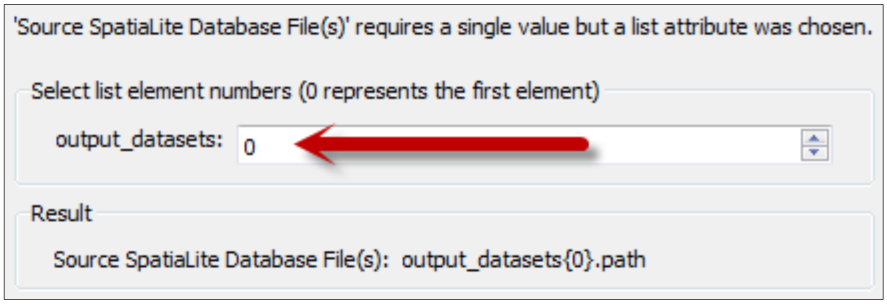
In other words, the first child process returned information about where it wrote its output data, and we can use that information to provide the input for the second child process.

In the Source SpatiaLite Database File(s) parameter, click the drop-down arrow. Select Attribute Value > output_datasets{}.path



Parameter	Value
Source SpatiaLite	database.sl3

Because this is a list attribute – i.e. there may be more than one value – you'll be asked which value you want to use. It will be the first one; element 0



'Source SpatiaLite Database File(s)' requires a single value but a list attribute was chosen.

Select list element numbers (0 represents the first element)

output_datasets: 0

Result

Source SpatiaLite Database File(s): output_datasets{0}.path

So select 0 (if it isn't already) click **OK** to close this dialog, and **Finish** to close the main dialog.

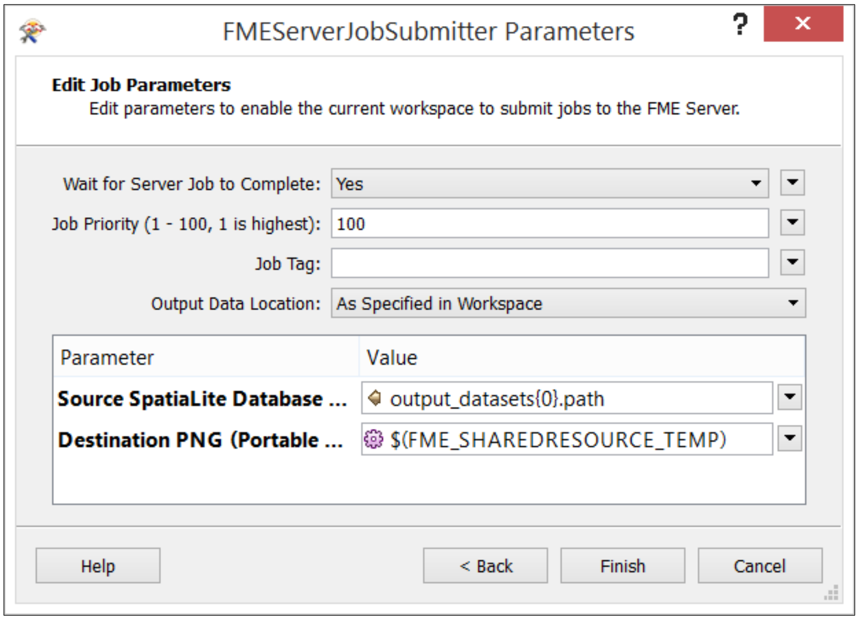
9. Chain Workspaces – Part 2

Now add a third FMEServerJobSubmitter transformer – again connected to the Succeeded port of the previous – in order to run the third child workspace.

Again enter the connection properties and again choose the same repository as before. This time the workspace to select is *Exercise5c-Begin-Data-Rasterize.fmw*

Again, set Wait for Server Job to Complete to Yes and set the source SpatialLite Database to use the output_dataset{}.path parameter (element 0).

Finally, set the destination directory to be the Temp resources folder. To do this use $(FME_SHAREDRESOURCE_TEMP). Using a resources location allows the data to be accessed via the web interface and REST API.



FMEServerJobSubmitter Parameters

Edit Job Parameters

Edit parameters to enable the current workspace to submit jobs to the FME Server.

Wait for Server Job to Complete: Yes

Job Priority (1 - 100, 1 is highest): 100

Job Tag:

Output Data Location: As Specified in Workspace

Parameter	Value
Source SpatiaLite Database ...	output_datasets{0}.path
Destination PNG (Portable ...	$(FME_SHAREDRESOURCE_TEMP)

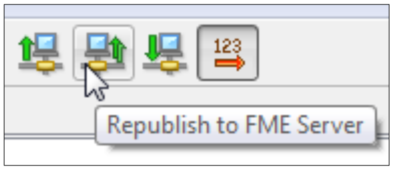
Once more, click **Finish** to close the main transformer parameters dialog.

Then add some Logger transformers again, so we can keep track of the results.

10. Publish Workspace

Publish the workspace back to FME Server, again into the *workflow_management* repository and registered with the Job Submitter service.

In fact, since no service changes need to be made, you can simply use the Republish button.



11. Run Workspace

Run the workspace in the Web Interface, as in step 7. Remember to assign the uploaded data to the two published parameters.

12. View Log

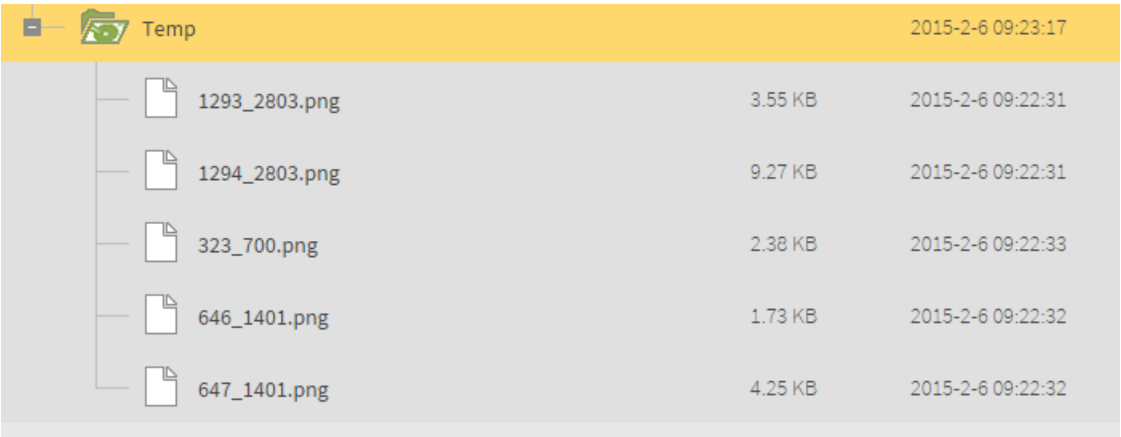
Once the workspace completes, click the button **View Log**

Again you should be able to find the same success message, indicating all three child workspaces have run to completion.

13. View Data

Once complete, click **Resources** on the sidebar menu and expand the Temp directory. You will see the five PNG files that were created by this workflow.

You can now download these files and access them via the REST API.



Temp		2015-2-6 09:23:17
1293_2803.png	3.55 KB	2015-2-6 09:22:31
1294_2803.png	9.27 KB	2015-2-6 09:22:31
323_700.png	2.38 KB	2015-2-6 09:22:33
646_1401.png	1.73 KB	2015-2-6 09:22:32
647_1401.png	4.25 KB	2015-2-6 09:22:32

Troubleshooting for Administrators

Troubleshooting for Administrators

A Scheduled Workspace Fails to Run
A Notification Fails to Start a Workspace
The REST API Fails to Start a Workspace
The FMEJobSubmitter Fails to Start a Workspace

This section shows a few basic troubleshooting techniques in case of emergency

A Scheduled Workspace Fails to Run

If a scheduled workspace appears to have not been run at the expected time, then the following suggestions may be of help.

- Ensure an engine is available, and that the scheduled job is not in a queue.
- Check the date and time **very** carefully to ensure the correct values were entered
- Check the timezone is correct. The web interface operates on local timezone, which is not necessarily the same timezone as the server is physically located.

A Notification Fails to Start a Workspace

If a notification fails to start a workspace as expected then check the troubleshooting suggestions at the end of chapter 4 (Real-Time).

The REST API Fails to Start a Workspace

If a workspace fails to run using the REST API then check the troubleshooting suggestions at the end of chapter 2 (Running Workspaces).

The FMEServerJobSubmitter Fails to Start a Workspace

If a workspace fails to run using the FMEServerJobSubmitter transformer then check the troubleshooting suggestions at the end of chapter 2 (Running Workspaces).

Module Review

Module Review

What you should have learned from this module

FME Server Basic Training 2015

This module introduced you to FME Server's Automation Functionality.

What You Should Have Learned from this Module

The following are key points to be learned from this module.

Theory

- Automation is the ability for FME Server to run a workspace without any user intervention or initiation.
- Scheduling is the ability to program FME Server to run a workspace in a repository, at a specific time in the future.
- Automated Data Processing is when a workspace is run in response to an event, rather than on a schedule.
- Workflow Management is a technique where a set of workspaces can be run in a particular sequence, and where branching can take place to run one workspace or another depending on the result of a set of pre-defined decisions.

FME Skills

- The ability to schedule a workspace to run at a particular time or date, either once or on a repeating schedule.
- The ability to start a workspace automatically with either a Notification or through the REST API
- The ability to create a parent workspace to control and execute a set of child processes

Course Wrap-Up

Product Information and Resources

Product Information and Resources

The Safe Software Web Site
The Safe Software Support Team
The Safe Software Blog
FME Manuals and Documentation

FME Server Basic Training 2015

Although your FME training is now at an end, there is a good supply of expert information available for future assistance.

Safe Software Web Site

Our web site is the official information source for all things FME. It includes information on FME products, Safe Software services, FME solutions, FME support and Safe Software itself.

www.safe.com

Safe Support Team

Behind FME are passionate, fun, and knowledgeable experts, ready to help you succeed, with a support and services philosophy built on the principle of knowledge transfer.

www.safe.com/support

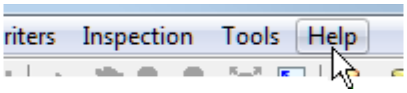
Safe Software Blog

The Safe Software blog provides thoughts on spatial data interoperability from the folks who make it their passion.

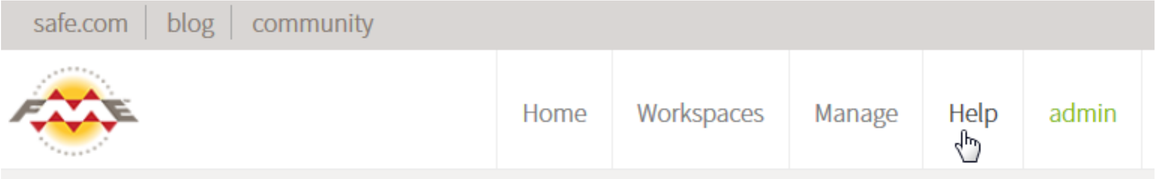
blog.safe.com

FME Manuals and Documentation

Use the Help function in FME Workbench to access context-sensitive help, and various manuals and documentation for FME Desktop.



Use the Help function in the FME Server web interface to access documentation, a knowledgebase, and developer information for FME Server.



Community Information and Resources

Community Information and Resources

- FMEpedia
- The FME KnowledgeBase
- FME Community Answers
- The FME YouTube Channel

Safe Software actively promotes users of FME to become part of the FME Community.

FME Knowledge Center

The FME Knowledge Center is our community web site - a one-stop shop for all community resources, plus tools for browsing documentation and downloads.

Knowledge Center
knowledge.safe.com

Here you will find a wealth of information; including tips, tricks, examples, and FAQs. There are sections on both FME Desktop and FME Server, with articles on topics from installation and licensing to the most advanced translation and transformation tasks.

FME Community Answers

FME community members post FME related messages and questions and share in answering other users' questions. Top users are known as "heroes". Come and see how they can help with your FME projects!

The FME Channel

This FME YouTube channel is for those demos that can only be properly appreciated through a screencast or movie. Besides this there are a host of explanatory and helpful movies, including recordings of most training and tutorials.

www.youtube.com/user/FMEchannel

Course Feedback

Course Feedback

Course Feedback Forms

FME Server Basic Training 2015

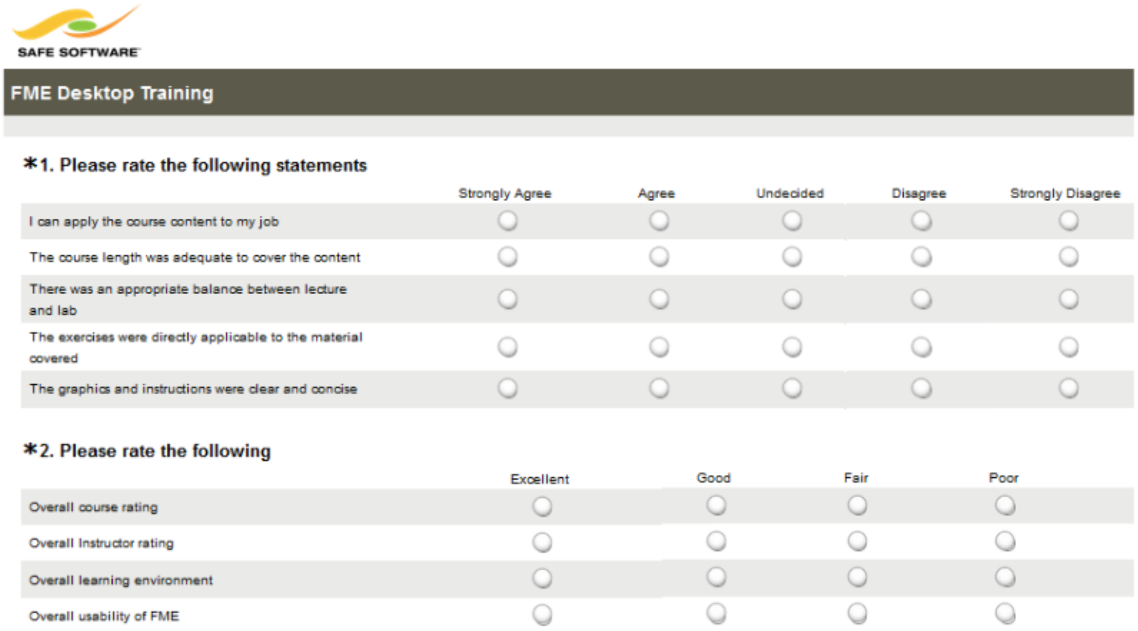
The format of this training course undergoes regular changes prompted by comments and feedback from previous courses.

There's one final Q+A to go – and this time you'll be telling us the answers!

Safe Software Inc. greatly values feedback from training course attendees. This is your chance to tell us what you really think about how well we're meeting your training goals.

The current course structure has been determined by attendee comments and we appreciate your feedback more than ever.

You'll be sent a link to a feedback form after your course.



SAFE SOFTWARE

FME Desktop Training

***1. Please rate the following statements**

	Strongly Agree	Agree	Undecided	Disagree	Strongly Disagree
I can apply the course content to my job	○	○	○	○	○
The course length was adequate to cover the content	○	○	○	○	○
There was an appropriate balance between lecture and lab	○	○	○	○	○
The exercises were directly applicable to the material covered	○	○	○	○	○
The graphics and instructions were clear and concise	○	○	○	○	○

***2. Please rate the following**

	Excellent	Good	Fair	Poor
Overall course rating	○	○	○	○
Overall Instructor rating	○	○	○	○
Overall learning environment	○	○	○	○
Overall usability of FME	○	○	○	○

Certificates

Certificates

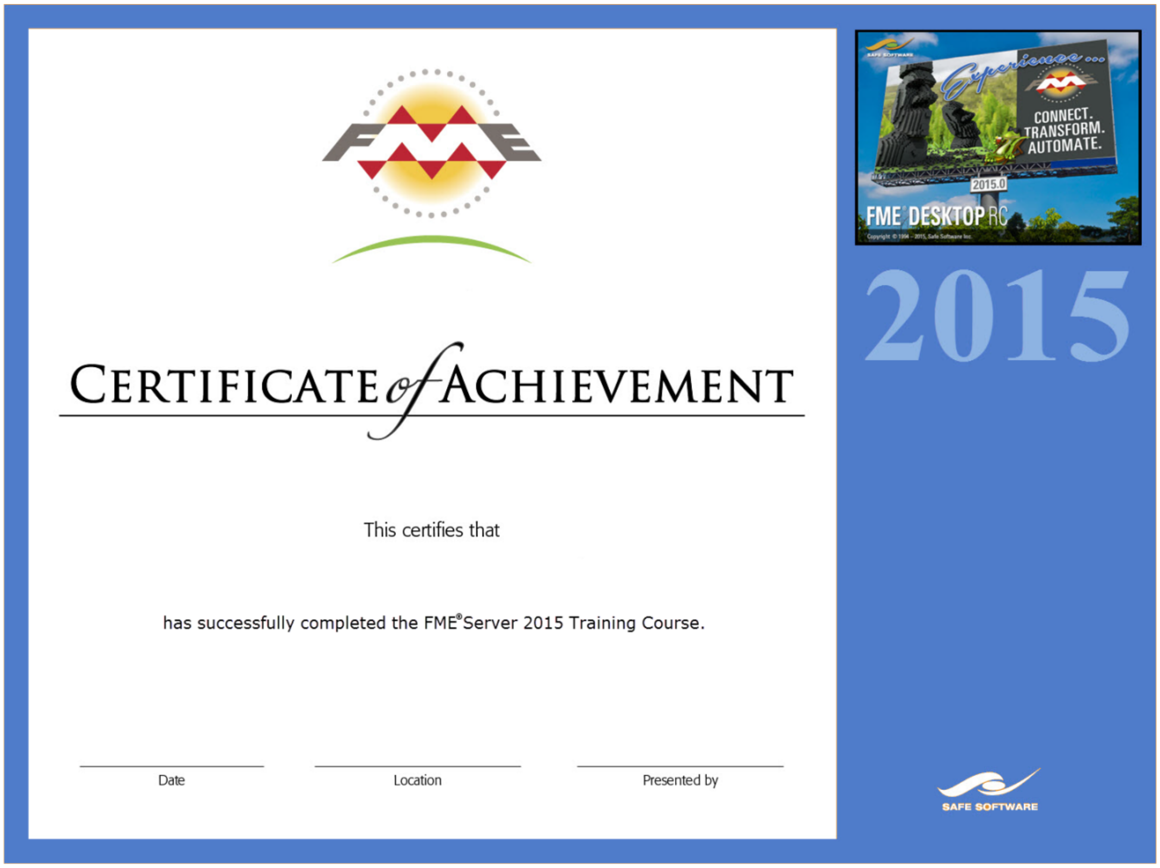
Congratulations!
Thank You

FME Server Basic Training 2015

All FME training course attendees receive a certificate.

Congratulations!

With the presentation of your certificate of achievement, you have now officially completed the FME Server 2015 Training Course.



CERTIFICATE of ACHIEVEMENT

This certifies that

has successfully completed the FME® Server 2015 Training Course.

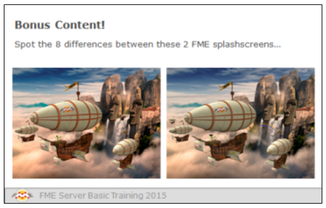
Date

Location

Presented by

FME
Experience...
CONNECT.
TRANSFORM.
AUTOMATE.
2015.0
FME DESKTOP RC
2015
SAFE SOFTWARE

Thank You



Bonus Content!

Spot the 8 differences between these 2 FME splashscreens...

FME Server Basic Training 2015

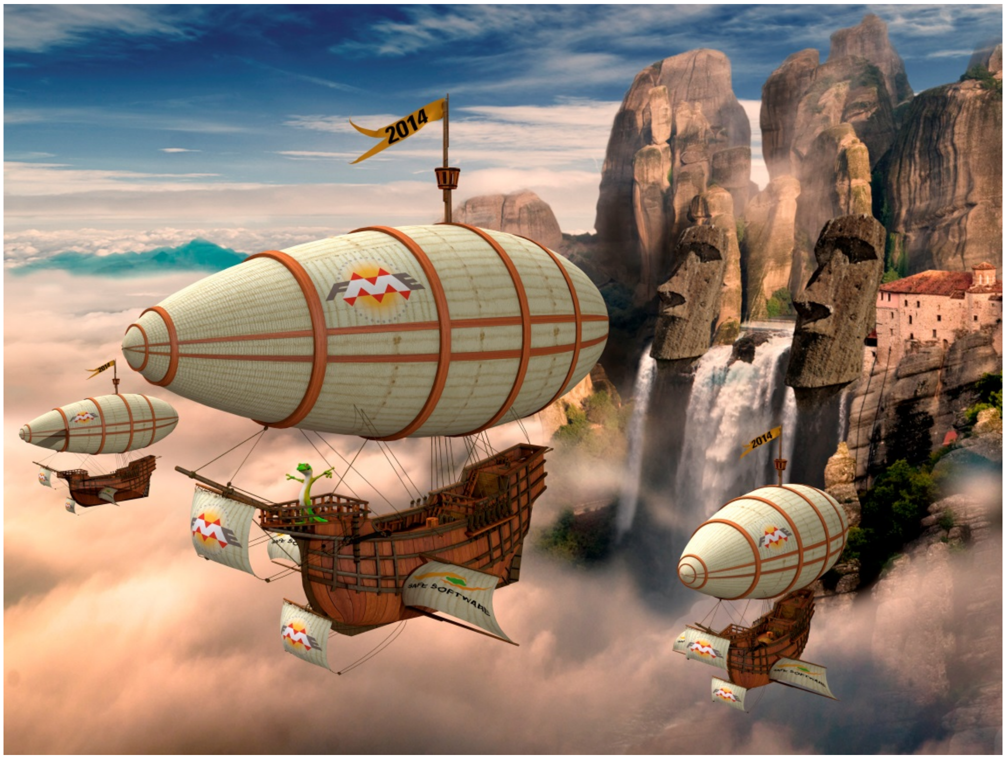
Thank you for attending this FME training course.

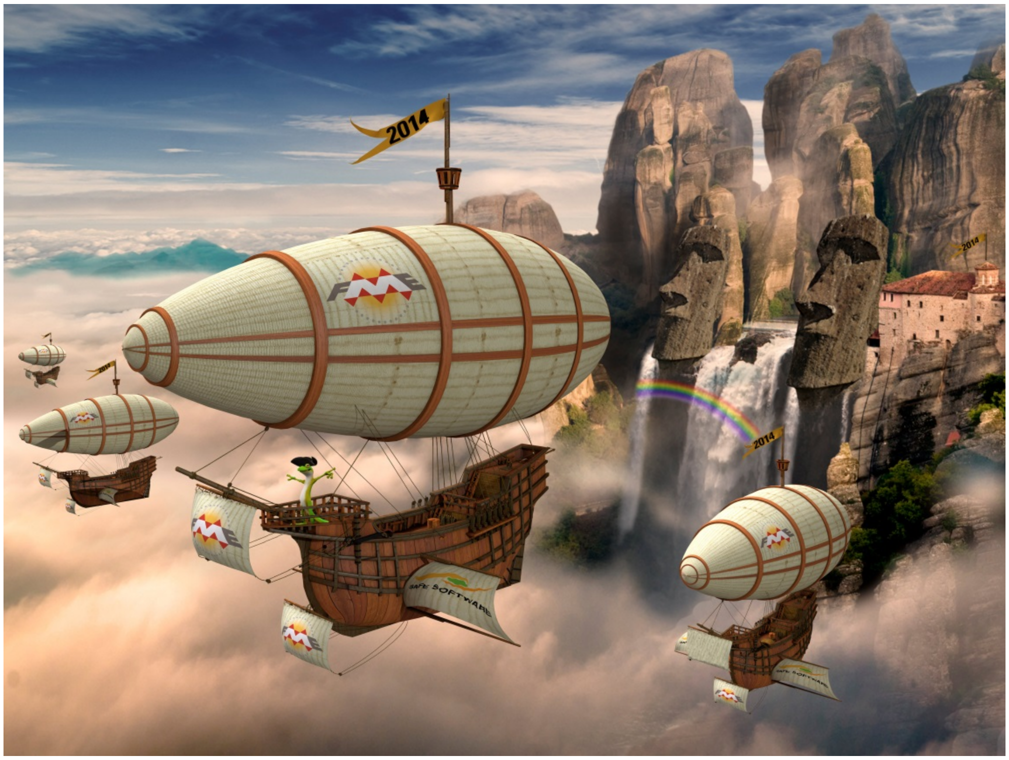
all of us at Safe Software wish you good luck and lots of fun as you build exciting new solutions using fme

Congratulations!

Judge GIS says…

"As a reward for reading this far, here's a small puzzle for you to try out. There are 8 differences between these two splashscreens. Can you spot them all?"


2014
FME
SAFE SOFTWARE
2014


2014
FME
SAFE SOFTWARE
2014
2014