



What are Automations?

Learning Objectives

After completing this unit, you'll be able to:

- Explain how Automations help you achieve enterprise data integration.
- Define an Automation.
- Decide when you need to use an Automation for your own workflows.
- Distinguish Automations from Streams.

Why Automation?

Enterprise Integration tackles two main problems that make it difficult for an organization to freely share data:

1. Data is stored in a format or system that restricts users from gaining access at the source.
2. Data is complex and expert knowledge is required to navigate the system or understand the data itself.

[Data Integration](#) and [Application Integration](#) help make information physically accessible and easier to understand, but total Enterprise Integration is a tall order for many organizations. This is where Automation comes into play. Automation is the process of building a task or service that can run with minimal human interaction. Whether you are a team of 10 or 10,000, the ability to employ automation is essential to enterprise growth.



Learn more about the business problems automation can solve with our [Enterprise Integration Wins: 5 Reasons to Add Automation](#) blog post.

What Are Automations?

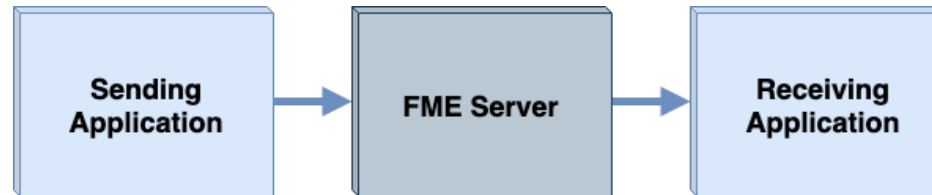
Automations provide a way for data to be pushed to FME Server in the form of short messages and react to these messages by triggering actions either internally or to external clients.

Automations are built on the concept of the FME Server Notification Service using a simple message (sometimes called an “alert”) that informs someone or something that a particular event has happened.

Automations in FME Server are made up of two different components: **incoming triggers** and **outgoing actions**.

Triggers alert FME Server to an event that has taken place either on FME Server or on an external application.

Actions are divided into two categories: Internal and External. Internal actions submit jobs to run on FME Server whilst External actions either send notifications to a client that an event has taken place on FME Server, or push result data out to a location.



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.

Automations are the part of the FME Server architecture that handles all incoming and outgoing notifications.



Throughout the documentation and within the FME Server web interface, you might see mention of Publications, Subscriptions, and Topics. These objects are still present in FME Server (Automations > Notifications (Classic)) and can be directly modified. However, they have been mostly replaced by Automations as of FME Server 2021.0. If you are a new user, we recommend focusing on learning Automations.

When to Use Automations

Automations allow you to incorporate a variety of triggers and actions into a single workflow. Automations should be used when you want to chain a number of different notification elements together in FME or build multiple reactions to a single event.

As well as using Automations to trigger an FME Server response to something that happened outside of FME Server, they are also useful when you want to send a message about something that happened on FME Server to an external client.

In either case, an automation is meant for sending a short message, usually in order to trigger an action from the recipient.

Automations and Real-Time Data

Real-time systems are those that act on events as they happen, and send information as it becomes available.

FME Server Demos

[Fork GitHub](#)[Download Zip](#)

Powered by FME

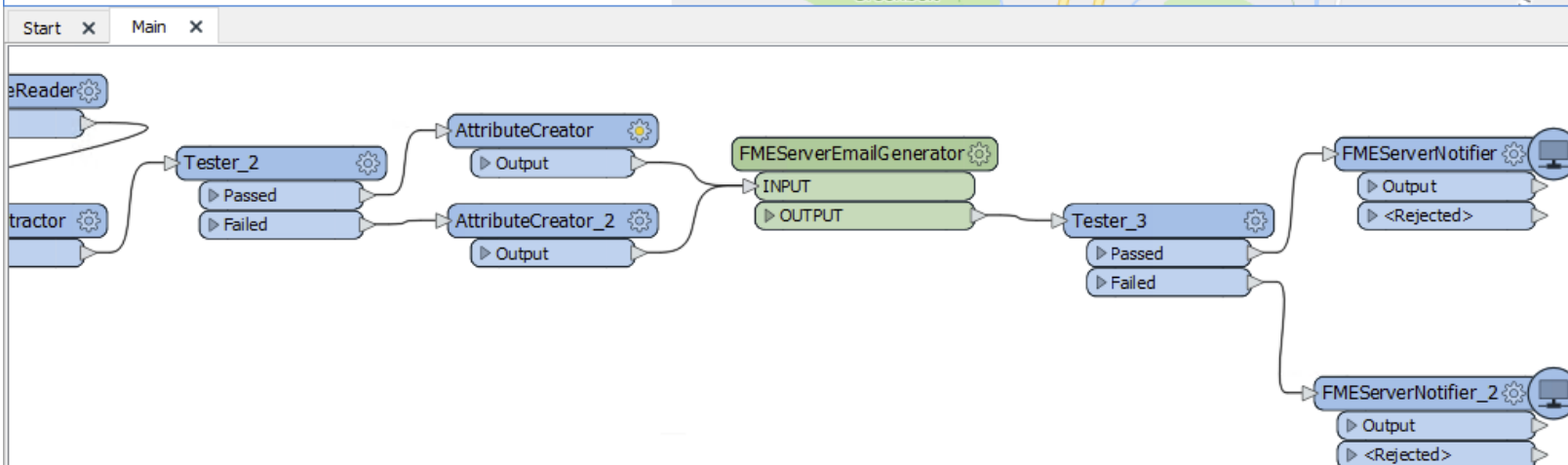
Real-time Display

Upload CAD data and watch it appear in real-time

Step 2

Email file to FMEServer

Use one of the DWG files you downloaded in the previous step. The demo is not designed for generic CAD drawings.

Database readycadqa@demos-safe-software.fmecloud.com[Next step](#)

Real-time is often implemented to monitor sensor networks, run event-driven processing, and/or to send alerts to external clients such as to an application or as an email. Therefore real-time data processing is commonly used to handle small amounts of information (rather than large datasets) that arrive either as individual events or as a continuous communication stream.

Real-world examples of data that would arrive as individual events include processing simple location data from a vehicle tracking system or sending an email to a system administrator in response to database table updates. An example of continuous data would be handling a stream of data being sent from a temperature sensor or lightning detector.

Real-time data in FME Server is handled in two ways: Automations and Streams.

Automations

Automations are how FME Server handles individual message streaming and alerts.

FME Server allows you to build Automation workflows that can be set up to listen for an incoming message from outside of FME and trigger a certain action in response or to send a single alert in response to an event that takes place on FME Server.

Triggers and actions include email, FME Server system events, directory watchers, WebSockets, and more.

You should use Automations if you need to listen to incoming messages at a rate of less than one per second.

Streams

Streams are a real-time technique like Automations. However, where Automations receive/send one-off messages, Streams involve a continuous flow of information.

Instead of a workflow being run once for each message, a Stream runs continuously, receiving messages as soon as they are sent. Because of this reduced overhead, this technique can process data at a much faster rate than Automations. For our purposes, “continuous” means that messages arrive at the FME Server at a faster rate than Automations could handle; say more than one message per second.

You should use Streams if you need to listen to incoming messages at a rate of more than one per second.



[Learn more about FME and stream processing.](#)



FME Academy Feedback Survey Legal Request On-Demand Virtual Machine



English

