E: hello@locus.co.nz
W: www.locusglobal.com

FME Form Basic Training Course Outline

Overview

Gain practical skills to confidently connect, transform, and integrate data using FME Form. Learn to build efficient, well-documented, and high-performing workspaces through expert-led, hands-on training.

Course Objectives

- Understand FME Form's role in data integration
- Connect to, inspect, and manage diverse data sources
- Build workspaces using transformation techniques
- Apply best practices for modular, documented workspaces
- Debug and optimise workspaces for quality outputs

Modules, Key Objectives, and Exercises

Why Data Integration?

- Define data integration and data silos
- Understand spatial data concepts (vector, raster, point, line, polygon)
- Explore the FME Platform along with FME user roles, industries, and data types

Connect To Data

- Connect to and create datasets
- Understand feature types and schemas
- Inspect, document, and manage workspace components
- Use Visual Preview for 2D/3D data interaction and verification

Transform Data

- Define and use transformers (e.g., HTTPCaller, JSONFlattener)
- Edit and manage schemas
- Speed up workspace development with feature caching and partial runs
- Find and filter data using appropriate transformers for tasks

Join Tables

- Differentiate appending vs. joining features
- Understand SQL-style joins: left, inner, full
- Use FeatureJoiner, DatabaseJoiner, SQLCreator, SQLExecutor, InlineQuerier

Create and Modify Attributes

- Manage user and FME attributes
- Create, rename, and manipulate attributes
- Use BulkAttributeRenamer effectively.

Filter Data

- Perform attribute and spatial filtering
- Use transformers: Tester, TestFilter, AttributeFilter, GeometryFilter
- Group and process features using Group By

Use Conditional Values

- Reduce workspace complexity with conditional attribute values
- Complete Parameter Condition Definitions

Work with Multiple Data Models Using Lists

- Understand and manipulate list attributes
- Compare relational vs. object-oriented data models

Design Workspaces for Advanced Reading and Writing

- Define workspace components: reader, writer, feature types
- Prototype, incrementally develop, and apply version control
- Manage reader/writer parameters and updates

Document Your Workspace

- Organize layouts with grids, guides, auto-layout, and connection styles.
- Use bookmarks and annotations for clarity and efficiency

Debug Workspaces

- Debug effectively using log messages and feature counts
- Understand rejected features and spatial/textual logs
- Apply best practices for identifying and solving issues