

3 FME Desktop Komponenten

FME Desktop besteht aus mehreren Applikationen, sog. Komponenten. Die drei Hauptkomponenten sind FME Workbench, FME Data Inspector und FME Quick Translator. Diese werden in Kapitel 3.1 ff. einführend erläutert. Ergänzend gibt es zwei weitere Komponenten: FME Integration Console und FME Licensing Assistant.

Mithilfe der FME Integration Console (vgl. Kapitel 2.4) lässt sich FME als Erweiterung für andere Applikationen (z. B. Esri ArcMap) freischalten. Die Lizenzierung von FME erfolgt durch den FME Licensing Assistant. Umfangreiche Informationen zur Lizenzierung sowie zur Installation und Konfiguration finden Sie in Kapitel 2.

Für die Entwicklung neuer Transformer und die Anbindung weiterer Datenformate gibt es ein Plug-in SDK. Die Integration von FME in andere Anwendungen kann über das FME Objects SDK (vgl. Kapitel 3.4) erfolgen. Für eine Reihe von Anwendungen existieren FME Erweiterungen, z. B. die FME Extension for ArcGIS (vgl. Kapitel 2.4).

FME Workbench	FME Integration Console	FME Plug-In SDK
FME Data Inspector	FME Licensing Assistant	FME Objects SDK
FME Quick Translator	FME Universal Viewer	Erweiterungen

Abb. 3.1: Übersicht über die FME Desktop Komponenten

Alle verfügbaren Komponenten können aus dem Windows-Startmenü (*Startmenü > FME Desktop [Version]*) aufgerufen werden.



Hinweis: In Windows 7 befinden sich die Komponenten FME Quick Translator, FME Integration Console und FME Licensing Assistant im Unterordner *Utilities*.

3.1 FME Workbench

Die FME Workbench ist die zentrale Komponente von FME Desktop. In ihr werden FME Workspaces zur Datenkonvertierung und -transformation modelliert und ausgeführt. Die Erstellung der Workspaces in dieser grafischen Oberfläche kann vollständig mithilfe der PC-Maus erfolgen. Nützliche Tastenkombinationen, auf die an verschiedenen Stellen hingewiesen wird, erleichtern die Arbeit. Eine Liste der wichtigsten Tastenkombinationen finden Sie im Anhang B.

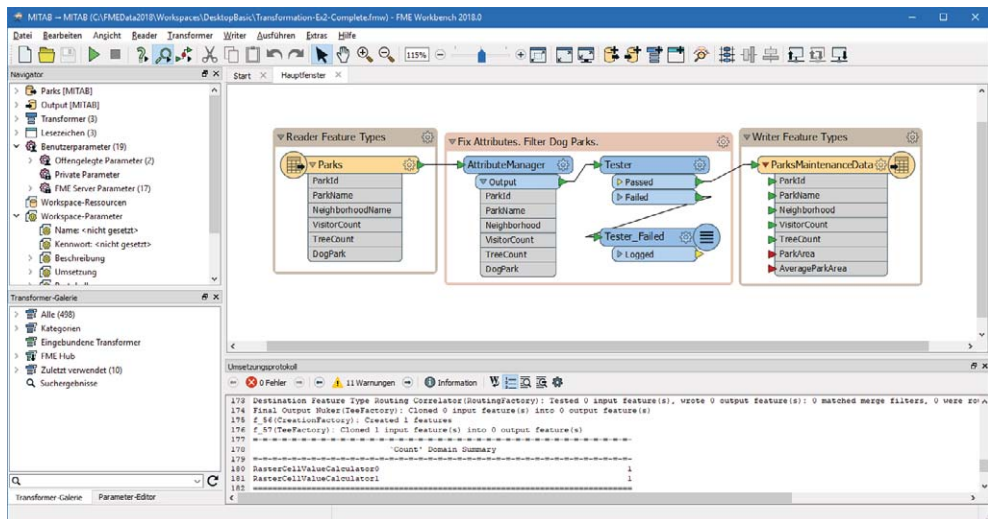


Abb. 3.2: FME Workbench

Die grundlegende Arbeit mit der FME Workbench basiert auf einem einfachen Prinzip. Mithilfe formatspezifischer **Reader** werden die zu verarbeitenden Daten in die Workbench eingelesen, durch **Transformer** verarbeitet und mittels formatspezifischer **Writer** neue Daten erzeugt.



Abb. 3.3: Schematische Darstellung Reader, Transformer und Writer

Reader, Transformer und Writer bilden das Grundprinzip der FME-Technologie **Extract, Transform und Load** (kurz: **ETL**) ab:

- Extract (Daten einlesen mittels Readern)
- Transform (Daten manipulieren mittels Transformern)
- Load (Daten laden bzw. schreiben mittels Writern)

Von entscheidender Bedeutung ist, dass die einzulesenden Daten zunächst in einem ersten Schritt durch den jeweiligen Reader automatisch in ein **internes Datenformat** bzw. **Datenmodell** überführt werden. Diese initiale Konvertierung macht ein übergreifendes Arbeiten mit unterschiedlichen Formaten und Datentypen möglich. Es spielt dabei keine Rolle, ob die ursprünglichen Daten aus einer Datenbank, einer Datei oder von einem Webdienst stammen. Die gesamte Verarbeitung der Daten erfolgt im internen Datenformat. Die formatspezifischen Writer sorgen dafür, die Daten nach abgeschlossener Transformation in das gewünschte Zielformat zu überführen. Die Wandlung aus dem internen Format in das gewählte Zielformat geschieht automatisch.

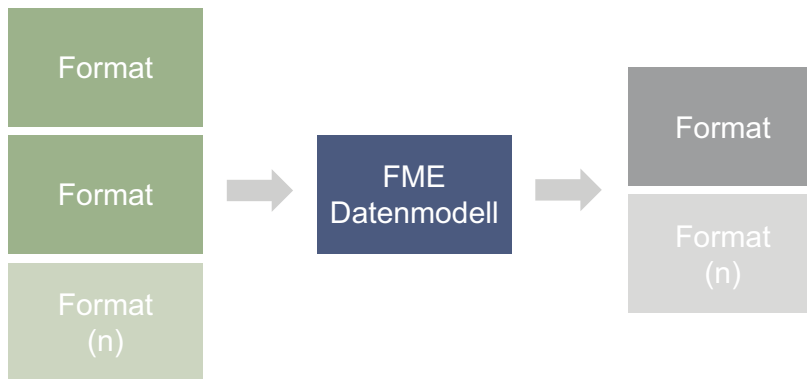


Abb. 3.4: Grundprinzip des internen Datenformats

Die Verarbeitung innerhalb von FME erfolgt objektbasiert. Die zu verarbeitenden Daten werden in FME Features überführt. Je nach Datenformat bzw. Namenskonvention in den unterschiedlichen Anwendungsdomänen entspricht ein FME Feature einem (Geo-) Objekt, Datensatz, Record oder einer Entität.

Mithilfe von Transformatoren werden FME Features verändert. Das Grundprinzip sieht vor, dass ein Transformer genau eine Art von Operation, z. B. eine räumliche Verschneidung oder die Berechnung einer Pufferfläche, ausführt. Aus Gründen der Arbeitseffizienz steht mittlerweile in vielen Fällen eine Reihe von Standardfunktionen innerhalb eines Transformators zur Verfügung. Grundlegendes zur Arbeit mit Transformatoren erfahren Sie in Kapitel 5.7. Eine Übersicht über die verschiedenen Transformer und Transformergruppen bietet Kapitel 6.

Ein erstellter Prozess kann als sog. Workspace (Dateiendung: **.fmw*) gespeichert, ausgeführt, weitergegeben oder beispielsweise als FME Server Prozess veröffentlicht werden.

Alles zum Arbeiten mit der FME Workbench, den Umgang mit Readern, Writern und Transformatoren finden Sie in Kapitel 5.

3.2 FME Data Inspector

Der FME Data Inspector ist die Komponente, mit der Daten in jedem von FME unterstützten Format angezeigt und untersucht werden können. Der FME Data Inspector wird verwendet, um Daten vor, während und nach der Umsetzung mit der FME Workbench zu betrachten und die Resultate zu verifizieren.

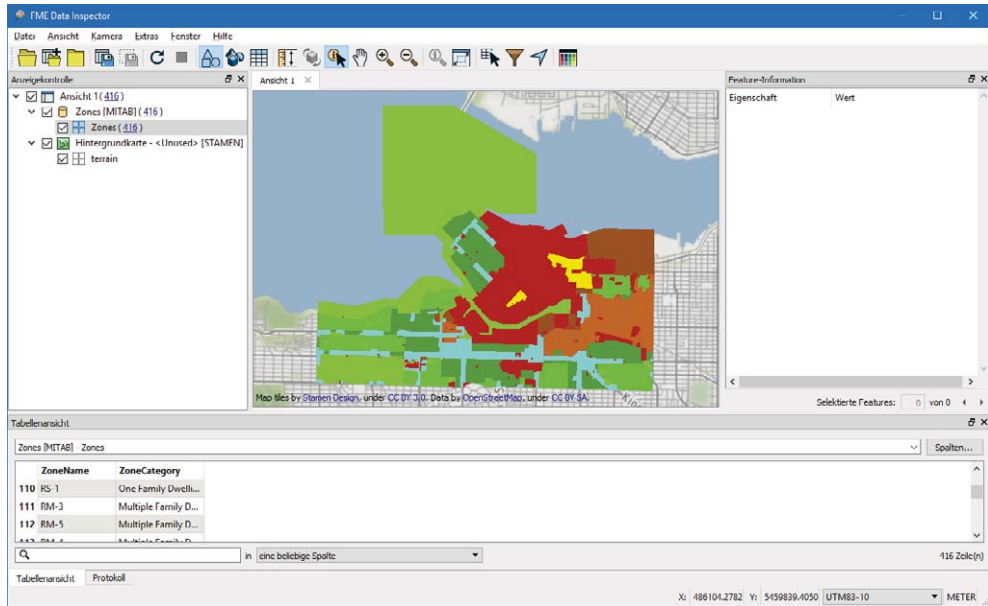


Abb. 3.5: FME Data Inspector

Genau wie in der FME Workbench werden die darzustellenden Daten beim Einlesen automatisch in das FME-interne Format gewandelt. Hierfür werden dieselben Reader wie in der FME Workbench verwendet. Formatspezifische Eigenschaften der Daten, wie Linienstärke oder Textart, bleiben auch nach der Wandlung in das interne Format in Form von sog. Formatattributen erhalten. Ausgewählte Merkmale, wie Farbe oder Textur, werden auch grafisch dargestellt.

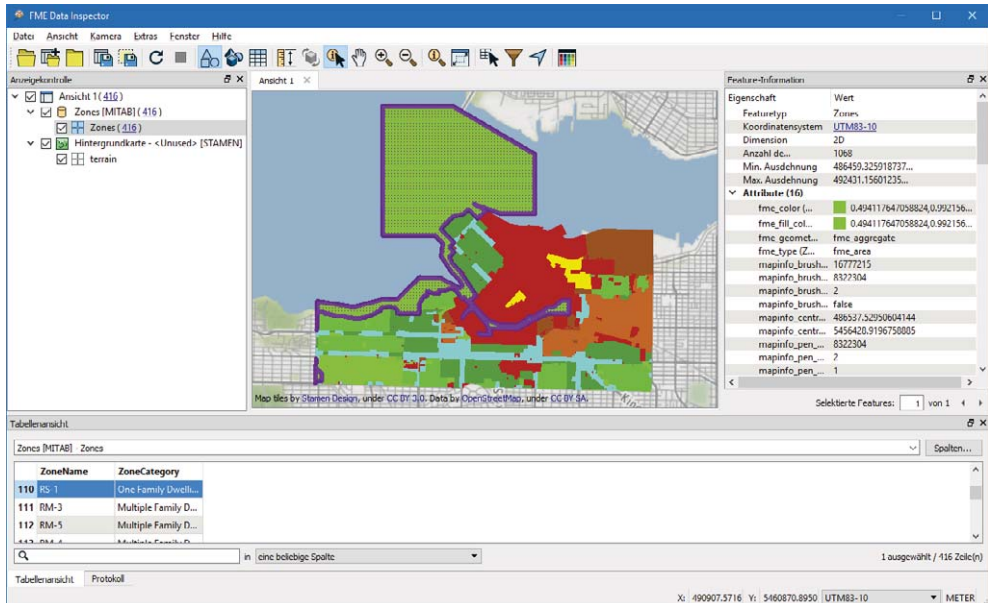


Abb. 3.6: Flächenfarbe einer MapInfo-Datei im FME Data Inspector

Der FME Data Inspector verfügt sowohl über eine 2D- als auch über eine 3D-Ansicht. Es können gleichermaßen Vektor-, Raster- und Punktwolkendaten angezeigt und abgefragt werden.

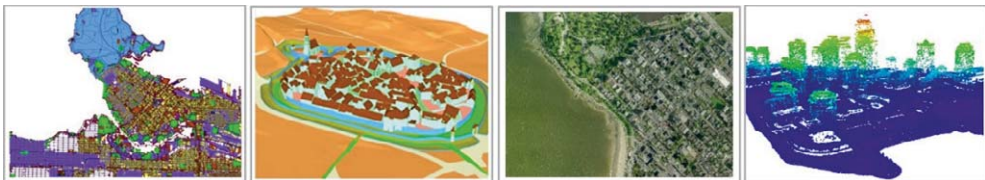


Abb. 3.7: Beispiele für 2D- und 3D-Vektordaten, Rasterdaten und Punktwolken

Ausführliche Erläuterungen zur Arbeit mit dem FME Data Inspector finden Sie in Kapitel 4. Dort erhalten Sie auch nützliche Hintergrundinformationen zum FME-Feature-Modell.

3.3 FME Quick Translator

Der FME Quick Translator ist die dritte Hauptkomponente und bietet die Möglichkeit zur Ausführung von automatischen 1:1-Datenkonvertierungen und Workspaces, die zuvor mit der FME Workbench erstellt wurden.

Einfache 1:1-Datenkonvertierungen zwischen zwei Formaten können über die Option *Umsetzen* konfiguriert und ausgeführt werden (vgl. Abb. 3.8). Eine 1:1-Datenkonvertie-

ung ist eine vollautomatische Umsetzung zwischen den beteiligten Formaten. Dabei gilt es zu beachten, dass eine derartige Umsetzung in Abhängigkeit der beteiligten Formate nicht zwingend verlustfrei ist. Je nach Anforderung an die Datenkonvertierung ist der Einsatz der FME Workbench aufgrund der erweiterten Möglichkeiten zur Datentransformation vorzuziehen.

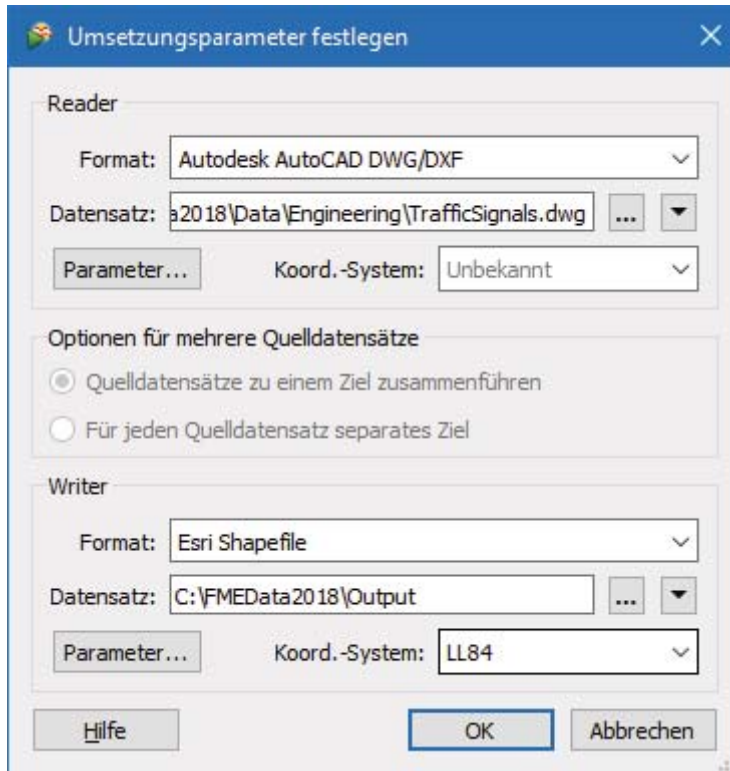


Abb. 3.8: Beispiel einer 1:1-Datenkonvertierung im FME Quick Translator

Der Aufruf eines Workspaces kann mithilfe der Option *Ausführen* erfolgen. Dabei werden vor der Ausführung zunächst die offengelegten Parameter (vgl. Kapitel 5.8.1) abgefragt.

Bei beiden Ausführungsoptionen wird das Protokoll der Umsetzung im Hauptfenster angezeigt und kann optional als Textdatei gespeichert werden. Im Dateimenü unter *Extras* > *Batch erstellen* gibt es zudem die Möglichkeit, ein Batch-Skript für eine 1:1-Datenkonvertierung zu erstellen. Mehr zum Thema Batch-Verarbeitung lesen Sie in Kapitel 7.4.2.

Sehr nützlich ist der Parameter *Optionen für mehrere Quelldatensätze* bei der Konvertierung von Datenbeständen, die in einem oder mehreren Verzeichnissen abgelegt sind. Die Option bietet die Möglichkeit, die Verzeichnisstruktur der Eingangsdaten im gewählten Zielverzeichnis nachzubilden. Praktisch ist dies beispielsweise dann, wenn vorhandene Geodaten in verschiedenen Raumbezugssystemen parallel in der gleichen Verzeichnisstruktur vorgehalten werden sollen (vgl. Kapitel 7.4.6).

3.4 Programmierschnittstellen (APIs)

FME kann um neue Formate und Transformer erweitert werden. Dazu bietet Safe Software verschiedene Programmierschnittstellen an.

Das **Plug-in Software Development Kit** (kurz: SDK) bietet die Möglichkeit, neue Reader, Writer und Transformer in C++, Java und Python zu entwickeln. Alle notwendigen Informationen zum Umgang mit den Schnittstellen befinden sich im Installationsverzeichnis von FME Desktop im Ordner *pluginbuilder*. Dazu gehören die Dokumentation der Schnittstellen, Beispiele mit Testdaten sowie für C++ und Java jeweils „Getting Started“-Dokumente zum Einstieg.

Hilfreiche Unterstützung bieten auch verschiedene Artikel im FME Knowledge Center. Als Einstieg bietet sich der Artikel „Developing a new transformer using the FME Plug-in SDK“ an (https://knowledge.safe.com/articles/How_To/Developing-a-new-transformer-using-the-FME-Plug-in-SDK).



Hinweis: Beachten Sie, dass sich die Schnittstellen teilweise im Umfang unterscheiden. Safe Software empfiehlt die Verwendung von C++.

Da die Entwicklung eigener Transformer zeitaufwendig ist und eine gewisse Einarbeitungszeit erfordert, empfiehlt es sich genau zu prüfen, ob die gewünschte Funktionalität nicht doch bereits als Transformer vorhanden ist oder durch Zusammenführen von verschiedenen Transformern zu einem benutzerdefinierten Transformer (vgl. Kapitel 7.5) realisiert werden kann. Außerdem lohnt sich auf der Suche nach einem passenden Transformer auch ein Blick in den FME Hub (<https://hub.safe.com>). Eine mögliche Alternative zur Entwicklung neuer Transformer ist die Verwendung von Python-Skripten innerhalb der Transformer *PythonCreator* und *PythonCaller*. Die Skriptsprache, die derzeit im Bereich Geodatenverarbeitung das Mittel der Wahl ist, bietet umfangreiche Funktionen, die mit diesen beiden Transformern auf FME Features angewendet werden können. Mehr zur Verwendung von Python in FME lesen Sie in Kapitel 9.8.

Bei der Überlegung, ein neues Format anzubinden, sollte zunächst geprüft werden, inwieweit bei textbasierten Datenformaten die vorhandenen Reader und Writer für Textdateien (CSV, Text File u. v. m.) verwendet und angepasst werden können. Die Anpassung bestehender Reader und Writer ist erfahrungsgemäß weniger zeitintensiv als die Entwicklung neuer Reader und Writer. In Kapitel 7.6 lesen Sie dazu, wie benutzerdefinierte Formate erstellt werden können.

Zur Integration der FME-Technologie in andere Applikationen steht das **FME Objects SDK** zur Verfügung. Auch hier befinden sich alle relevanten Informationen (Dokumentation der Schnittstellen, Beispiele mit Testdaten etc.) im Installationsverzeichnis von FME Desktop im Ordner *fmeobjects*. Wie beim Plug-in SDK werden auch hier verschiedene Programmiersprachen unterstützt. Safe Software empfiehlt hier ebenfalls die Verwendung von C++.