Iowa DOT FME Best Practices Guide

Public Version 1.1 | 06/2019

SANITIZED PUBLIC VERSION

Contents

Guide created by David Runneals, GIS Developer at HNTB, for the Iowa DOT using a combination of existing knowledge and workflows of FME at the Iowa DOT, FME UC presentations, FME knowledge forum posts by the greater FME community, and other FME documentation. For additional tips and tricks on improving performance and other best practices, check out the following links: Performance Tuning FME, Let the Database Do the Work, Layouting Your Workspace, Best Practice Validation Project, FME Desktop Training Best Practices, Code Smell and FME Golf, and Advanced Tricks when using Bookmarks.

Document History

Version	Description	Date
1.0	First Version	20190620
1.1	Minor updates. Moved some content to appendices. Merged some of Mark's Best Practices in here as well.	20190625

Notes

FME does NOT currently support SAML authentication with Portal reader/writer, however it is in the works for FME 2020. This means that you will NOT be able to use your Azure AD account account to access ArcGIS Portal. A suggested workaround is to write to a file geodatabase and then upload that.

For this public version, I have highlighted and set generalized specific references to the following (which you can replace with your own):

- FME Production server url: yourprodfmeserver.gov
- FME Test server url: yourtestfmeserver.gov
- Generic UNC paths: \\yourfmeserver\
- Generic FME user account that runs your FME server: yourfmeserveruser
- Additional generalized references do exist and all of them are highlighted in magenta

General Tips

- Use the newest version of FME possible EXCEPT when publishing to FME server when you should use the version that matches FME server. You should NEVER use beta versions in a production environment.

  By using the latest version of FME, some previous issues may have been resolved and performance will likely increase due to evolving transformers and modifications to the software.

- Look at the FME Desktop Training Best Practices (which has been partially included in this document)
- Determine if you require either an all or nothing approach (if a record fails, then don't write any data at all) or if invalid features can be filtered out (see the truncate tip under the writer section).
- Implement validation checks for error trapping, so workspaces won't fail. Log features (if logs will be reviewed regularly) or send email notification alerts to the data owner, so that they can fix it.
- Keep both test and production writers in the workbench, but just disable the ones that aren't needed before uploading to server.
- If your workbench file is excessively large, you may have to copy over everything to a new file to reduce the size and may resolve issues with corrupted workspaces.

Workspace Organization

- Keep it Simple Silly! Keep it clean and understandable!

  You should be able to show your workspace to a co-worker and have them understand what is occurring. Reduce "Code Smell". **GOAL: Have a co-worker be able to understand and troubleshoot an issue in the workspace in less than an hour.**

- Use bookmarks to group objects together that distinctly define different sections, components, or sub-tasks

  Logically use bookmarks to collapse complicated sub-tasks into a smaller footprint. A recommended best practice is having between 5-10 objects per bookmark. Don't over use bookmarks! A bookmark should NOT have less than 2 objects in it.

  Label bookmarks with a short and sweet description (name) that describes what is happening within the bookmark.

  Appropriately color code your bookmark (see the color coding sub-section below)

  Bookmarks can be SUPER useful when used appropriately. See some advanced tricks when using bookmarks here and bookmark documentation here.

- Rename transformers to give a 1-2 word description of what you're doing.

  Convert a transformer with a name like "AttributeManager_6" to "AM_DescriptionHere". Don't forget to use CamelCase.

- Add annotations!

  Use annotations broadly to describe functionality where it happens or attach it to a transformer to describe in detail what is happening. A best practice is having one annotation for every 3-5 transformers.

  Appropriately color code your annotation (see the color coding sub-section below).

  Ensure that annotations are attached to either a transformer, bookmark, or connector. By attaching the annotation, it will ensure that collapsed bookmarks will function properly when the workbench is viewed in FME Server. To attach an annotation, right-click it and select which item it should be connected to.

- Keep connections organized using junctions, tunnels, hidden connections, or vertexes.

  Use junctions, tunnels, and hidden connections to help keep your workspace clean and understandable. More information on junctions, tunnels, and hidden connections can be found here. Best practices in using these can be found here. If it makes it more complicated to understand, don't use them! Make sure to label appropriately when using!

  Avoid connectors crossing over each other. This can be achieved by moving ports up/down by right clicking the port and clicking "Move Up" or "Move Down".

Insert vertexes to redraw connectors to avoid overlaps.

- Ensure that ALL metadata in the workspace parameters completed!

  See the sub-section below on required metadata. See the workspaces parameters documentation for additional guidance.

- Use consistent object layout

  Object layout is a personal preference but keep it consistent! More information on object layout can be found here.

  Keep enough space between transformers to see data counts on the connection lines.

  Utilize the Align tools to create a consistent look/feel.

Bookmarks/Annotations Color Coding

- Doesn't work/Needs improvement/To-Do (Color: #ff0000, Opacity: 5)
- In progress (Color: #ffaa00, Opacity: 5)
- Completed & Working (Color: #007400, Opacity: 5)
- Notes (Color: #939393)
- Summary Annotation (Color: #bcebff)
- Group of Multiple Bookmarks (Color: #ffffcd, Opacity 5)

Workspace Parameters

REQUIRED Metadata

This is **VERY** important!!!

- Workspace Name: (NOTE this is NOT the file name, but the name in the workspace parameters.) Use the standard naming conventions.
    - WorkspaceName_Dev(01-99)(A-Z) when developing or testing workspaces. For example, if you were updating the production workspace DBtoAGOL_Prod04 and this was your first update it would be named DBtoAGOL_Dev04A.
    - WorkspaceName_Prod(01-99) for workspaces that have been finalized and meet or exceed all documented requirements and best practices.
- Overview: Describe what the workspace does. Where does the data come from and where does it go to? What is it for/why is it needed? How often is the source data updated? How often does it need to run? Make a note if the python version requires a specific version to run.
- Help: Include a point of contact for someone that supports the workspace and tips to help with common issues (if any occur).
- History: When finalizing a workspace for production use by others, you MUST include your name, date, and a brief but detailed description of what you changed since the last time the workspace was production quality. You can also This is also useful to keep track of minor changes throughout the development lifecycle so you know what all you've done!
    - Date Format: YYYYMMDD

Scripting

Use **Python 3.7+** by default for future support. If you are writing to ArcGIS Online via the python script, use the **ArcGIS Pro 2.x 3.6.** Some transformers may still require 2.7, so you may have to downgrade python. Make sure to note this in your workspace.

Readers

General Tips

- Use the newest readers

  Ensure that your readers match the latest version of FME (or match the version that is on FME server if uploading there). The great developers over at SAFE constantly refine and update readers and using the latest ones ensures that you get the best performance out of your workspace.

- Use as few readers as possible

  If you have multiple tables that are coming from the same database with the same user, use 1 reader with multiple feature types. If you are optimizing existing workspaces, pick a reader and import the tables to that reader and remove the other ones. This helps make updating workspaces easier and makes it cleaner! Note that you will still need multiple readers if you are using transformers such as the FeatureMerger where you set suppliers first.

- Disable feature types that aren't being used

  When feature types are left enabled on workspaces and not required, it creates clutter on your canvas and reduces performance because the feature type data is still read in.

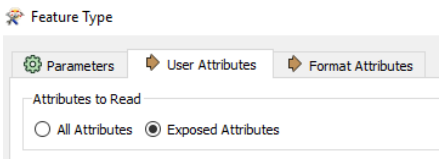
- Use Database/Web Connections where possible

  When you use a connection instead of embedding the connection parameters, you can easily create and update a single connection in the future for all your workspaces instead of having to embed the parameters to each location that requires it. See the section below on Web/Database Connections for best practices.

  If reading from a database, and you use OS authentication, ensure that yourfmeserveruser users have access to the database prior to uploading your workspace to FME server.

- Expose only attributes that you need!

  If looking to speed up existing workspaces, the easiest and quickest thing to do is to define attributes on your readers. This article gets into detail and compares methods of filtering data, but an easy way to think of it is that each of your attributes is a weight and your feature is a horse. The more weight (attributes) you put on your horse (feature) the slower it (workspace) will run.



- Expose only features that you need!

Similar to the previous point, the less records you read in, the quicker your workspace will run! When possible, use the WHERE or SELECT parameters to define features prior to reading them in. Also use the bounding box search if you only need a specific geographic area.

- Leverage databases to do the hard work!

  Use complex, yet simple sql statements (i.e. select, where, and group by). See this link for tips.

- When developing, set the "Max Features to Read" parameter for each of your readers.

  When developing and testing, a good idea is to set the Max Features to Read parameter. This is super helpful when using large datasets, but you only want to verify your process is working with a smaller subset before running them all through. **Ensure that you REMOVE this prior to publishing to FME server!**

- Enable Feature Caching on FME Desktop

  When developing and testing, a good idea is to enable feature caching. It helps speed up your reading and writing by writing each transformers output to a table and allows you to easily visualize the data in inspector. It also speeds up the workspace by only running from the point in your workspace where changes have been made, so you don't have to constantly re-run the first half of your workspace that you didn't change.

- Attempt to have all value-added data (i.e. route ID, reference post, etc) in the source dataset.

  Having value-added data such as route ID or reference post in the source dataset saves a LOT of time and improves accuracy of the data. A great example of this is a workspace that spent 20+ minutes getting route ID and reference posts, when it could be added to the original dataset, and only had to be ran once per feature. Think smarter, not harder.

ArcGIS Server Feature Service

When trying to connect to authenticated REST services, you will need to set the authentication to "NTLM" and then for username enter yourorganizationaddomain\USERNAME

ArcGIS Portal

FME 2019 does not support SAML authentication. Only built-in user accounts are supported.

Attribute names in portal feature services are ALL lowercase!

Oracle Spatial

Pre-filter SDO Geometry

When using an oracle spatial reader for reading geometry, ensure that you include the following in the **SELECT** statement area (which can be found by going to the *Navigator > Feature Type > Parameters > Format* OR opening up a feature type by clicking the gear), which will ensure that your job will always run successful, even if there is no geometry or if there is bad geometry in the tables. Replace the highlighted portion with your schema and table name.

Lines

SELECT * FROM SCHEMA.TABLENAME LINETABLE WHERE SDO_GEOM.VALIDATE_GEOMETRY_WITH_CONTEXT (geometry, 0.005) = 'TRUE' AND GEOMETRY IS NOT NULL AND (LINETABLE.GEOMETRY.SDO_GTYPE = 2002 OR LINETABLE.GEOMETRY.SDO_GTYPE = 2006)

Polygons

SELECT * FROM SCHEMA.TABLENAME POLYGONTABLE WHERE SDO_GEOM.VALIDATE_GEOMETRY_WITH_CONTEXT (geometry, 0.005) = 'TRUE' AND GEOMETRY IS NOT NULL AND (POLYGONTABLE.GEOMETRY.SDO_GTYPE = 2003 OR POLYGONTABLE.GEOMETRY.SDO_GTYPE = 2007)

Points

SELECT * FROM SCHEMA.TABLENAME POINTTABLE WHERE SDO_GEOM.VALIDATE_GEOMETRY_WITH_CONTEXT (geometry, 0.005) = 'TRUE' AND GEOMETRY IS NOT NULL AND POINTTABLE.GEOMETRY.SDO_GTYPE = 2001

Geodatabase SDE

Filter feature classes

Set the WHERE Clause (which can be found by going to the *Navigator > Feature Type > Parameters > Format* OR opening a feature type by clicking the gear OR for an entire reader can be set *Navigator > Parameters*)

- Filter out RAMS geometry

  Use the current RAMS network geometry with the following WHERE statement:

  EFFECTIVE_END_DATE IS NULL

- Filter out invalid geometry

  Line geometry (where SHAPE.LEN is the name of the shape length field):

  SHAPE.LEN IS NOT NULL

  Other geometry (where SHAPE is the name of the shape field):

  SHAPE IS NOT NULL

Writers

General Tips

- Use Database/Web Connections where possible

  When you use a connection instead of embedding the connection parameters, you can easily create and update a single connection in the future for all your workspaces instead of having to embed the parameters to each location that requires it. See the section below on Web/Database Connections for best practices.

  If reading from a database, and you use OS authentication, ensure that yourfmeserveruser users have access to the database prior to uploading your workspace to FME server.

- Keep test writers in production workspaces

  When publishing workspaces to production, simply disable the test writer in the navigator panel. This helps make future testing easier.

- Set the write order so the one getting the most records is set first.

  Modify the write order when the Workbench is writing to more than one location. You can set this parameter in *Navigator > Workspace Parameters > Translation > Order Writers By*.

- If appropriate, update instead of insert where possible

  When writing datasets, it is much faster and more efficient to update than it is to truncate and rewrite an entire dataset.

  When updating data in a destination dataset, writers have the option to update based on any unique field in the dataset.

  When updating a Portal or ArcGIS Online feature service, the writer can only update based on ObjectIDs or GlobalIDs. By default, ObjectIDs are used. If GlobalIDs are used, ensure that this option is enabled in the service parameters. FME reads GlobalIDs differently depending on their type. FME includes {} bookends for GlobalID types, while it doesn't include them for GUID types. This could be the culprit if errors occur when using GlobalIDs.

- Determine the method for truncating destination dataset

  Truncate with Writer: Leveraging the "Truncate" functionality within a writer takes a bit longer, but it allows a rollback option in case the workbench cannot finish. This method is ideal when the destination dataset needs to have data in them, even if it isn't complete or the most accurate.

  Truncate with Script: Leveraging a SQL or Python script to truncate the destination dataset prior to writing data is much faster than the truncate within the writer, but comes at the cost of not being able to rollback the truncate. This means that the destination dataset will remain empty until the next time the workbench runs. This method is ideal when the destination dataset must only show the latest data or when old data is considered invalid with the risk that when the workspace fails, there will be no data in the destination dataset. It is strongly

suggested that this option is used only after trying the Truncate with Writer and experiencing slow truncate times. The following statement should be used in the **SQL To Run Before Write**:

```
FME_SQL_DELIMITER ;
TRUNCATE TABLE SCHEMA.TABLENAME;
```

Geodatabase SDE

Transaction Type should always be set to "None" when writing to PGIS or TGIS.

Truncate with SQL Script for ETL

The statement below should be put in the **SQL To Run Before Write** (which can be found in: *Navigator > Geodatabase_SDE Writer > Parameters > Advanced*). Replace the highlighted portion with your schema and table name.

```
FME_SQL_DELIMITER ;
TRUNCATE TABLE SCHEMA.TABLENAME;
```

ArcGIS Online Feature Service

If writing to ArcGIS Online more than once (scheduled job) or a lot of data, consider writing to a file geodatabase and using this python script to append the data (much faster than the applyEdits, which the writer uses).

ArcGIS Portal

FME 2019 does not support SAML authentication for Portal. Only built-in user accounts are supported.

Attribute names in portal feature services are ALL lowercase!

Transformers

General Tips

- Reduce duplicated and similar transformer types to improve performance and make your workspace layout cleaner.

  Attempt to combine attribute transformers like AttributeCreator, AttributeRemover, AttributeRenamer, etc into a single AttributeManager transformer.

  Duplicating transformers is a bad practice as it requires multiple transformers to be edited individually, which is harder to maintain.

  If you find yourself using an AttributeFilter with duplicated AttributeManager transformers, Try using the AttributeValueMapper or SchemaMapper transformer to allow scaling with a minimum number of edits.

- Use the least amount of custom python scripting as possible.

  This not only reduces dependencies and other issues when switching between python versions, but also makes it easier for others to understand your workspace. If you do have to use python scripting, use comments in the code!

- Use as few transformers as possible.

  This makes workspaces more efficient, easier to understand, easier to update, and your canvas less cluttered.

- When creating attributes in your workspace, prefix them with _

  This helps the attribute move up to the top in inspector. It also helps it to stand out, so you know that it wasn't part of any of the original datasets.

- Download/Print the FME Transformer Reference Guide!

  This guide provides tips and tricks like keyboard shortcuts, along with a list and description of each FME transformer.

Attribute Manager

- Try to condense multiple attribute transformers into a single attribute manager. This is especially important when using @Expression() in multiple transformers, to cut down on TCL calls and compiling.
- View additional tips and tricks for the attribute manager in this Safe Article.

Date/Time

To parse out the hour to add:
@Substring(@DateTimeNow(local),24,1)

To add time (Replaces DateTimeCalculator):
@DateTimeAdd(@Left(@Value(timestamp),14),PT@Substring(@DateTimeNow(local),24,1)H)

To calculate difference (Replaces DateDifferenceCalculator):

- **Both times are required to be in same format**

@DateTimeDiff(@Value(CURRENT_DT),@Value(PHOTO_FILEDATE),Minutes)

Regular Expression

(.[^<]*)

This will find anything between the characters to the left and right of it. Use with a string searcher.

Web/Database Connections

Using database and web connections where possible make it simpler for managing parameters when reading/writing to a database or an online service. Following the naming conventions below is strongly suggested when publishing to FME server. Note that if you use OS authentication for database access, yourfmeserveruser must be granted access to the database when publishing to FME server.

ArcGIS Online

UserName@AGOL (ie: IowaDOT_GIS@AGOL)

FME Server Publishing Standards Checklist

FME Workspace General Requirements

- User's FME version must match that of the FME server to which they are publishing to. Temporary exceptions may be made to accommodate bug workarounds.
- FME Workspace Parameters are filled in and complete.
    - Name
    - Category – Choose one from the FME default dropdown.
    - Overview – Define information about what the workspace does, along with other requirements.
    - Help – Who should be contacted if questions arise regarding this workspace.
    - History – Ensure you keep track of changes made and when they were made, especially when making changes to workspaces in production.
- Workspace is properly documented with appropriate number of annotations and bookmarks that logically group parts of your process together.
- **Ensure that debugging is DISABLED when uploading workspaces to FME server (unless absolutely necessary!**
- Ensure that the python interpreter is not set to 2.7, as that is now a deprecated version. 2.7 should only be set if is required by custom transformers.

Readers/Writers

- There are no references to files on local drives. All files and connections must be referenced via UNC path to the specified shared directory on FME Server that matches the user's repository folder.
- Use Web and Database Connections where needed and use the standard naming conventions to reduce duplicate connections. All connections will be managed by the GIS Team. If you require one that isn't already created, contact the GIS Team to help get one created.
- When uploading workspaces to FME server, ensure that the test writer is disabled in the navigator panel. (*Tools > FME Options > Translation > Log Message Filter*)

Transformers

- If custom transformers are required, work with the GIS Team to get them uploaded to the shared transformers folder so others can use them. Never upload any versions of SkyFire to FME server, as that is already managed by the GIS Team.
- There should be no inspectors or loggers enabled when uploading to server. This slows down the workspace when it runs.

Production Publishing Process

DESKTOP

a. Create workspace in FME Desktop and run to ensure that it works correctly.
b. Ensure debugging is disabled.

FME TEST Server

a. Verify publishing standards have been met and files are referenced using \\yourtestfmeserver\ UNC paths.
b. Publish to FME Test
c. Run workspace on FME Test. Verify output is correct.
d. Once you have a working version, commit the version. **If you are updating a workspace, don't commit unless it actually works!** In the commit notes, document what you changed.
e. Continue to the process for publishing to FME production server.

FME PRODUCTION Server

a. Verify files are referenced using \\yourprodfmeserver\ UNC paths.
b. Have a FME-using colleague review your workspace.
c. Publish to FME Production.
    a. Ensure that you commit your FME Job to version control and add a date and other descriptive text about what you modified (or "Initial Commit" if new). This helps provide a history of changes that have been made to roll back to older versions if necessary.
d. Run workspace on FME Production. Verify output is correct.
e. Once you have a working version, commit the version. **If you are updating a workspace, don't commit unless it actually works!** In the commit notes, document what you changed.
f. Contact the GIS Team to schedule your job (if needed). If your job takes longer than 10 minutes to run or is to be scheduled to run more than once a week, contact the GIS Team to help optimize your workspace to conserve server resources.

Notes on Upgrading Workspaces

General Tips

- Ensure that the best practices highlighted above are included in your workspace
- Only update workspaces up to the current version of FME server
- Create a copy of your workspace, upgrade/test the copy before pushing it to production

Readers

All readers should be replaced when upgrading a workspace. Before replacing a reader, inspect the reader to ensure that any select/where statements are retrieved first so that they can be copied to the new reader.

Tip: When testing new readers, disable the existing one until testing is done, then remove it.

Quick Tip: You can also update a reader by right clicking on it in the Navigator window and select "update reader". Some options that were configured may be reset to default values, so you need to make sure that they get updated.

Writers

All writers should be replaced when upgrading a workspace. Before replacing a writer, confirm that the settings such as update/insert/delete/truncate are properly copied.

Tip: When testing new writers, disable the existing one until testing is done, then remove it.

Quick Tip: You can also update a writer by right clicking on it in the Navigator window and select "update writer". Some options that were configured may be reset to default values, so you need to make sure that they get updated.

Transformers

Upgrade transformers when documented performance enhancements exist (or when possible in other circumstances). Update and test one transformer at a time. This will allow you to identify issues during upgrading. When older workbenches are opened in newer versions of FME Desktop, transformers that need to be upgraded are identified in the Navigator > Upgradable Transformers menu.

Appendix A: Key Terms

ETL: The abbreviation for the Extract, Translate, Load process. This term is generally used to describe the process used in reading, transforming, and writing data.

FME: Feature Manipulation Engine

FME Desktop: This is the desktop-based version of FME. This software is used to create and test workbenches before they are published to FME Server.

FME Server: This the server-based version of FME. This software provides a web-based interface to automate and schedule when workbenches will be run.

Objects: This refers to transformers, readers, and writers that are on the canvas.

Reader/Source Dataset: Reads data from a source and brings into a workbench so it can be processed.

Safe Software: Name of company that created FME Desktop and FME Server.

Transformer: A tool used inside a workbench to manipulate data.

Workspace: The actual data ETL model that could be comprised of Readers, Writers, Transformers or Scripts.

Writer/Destination Dataset: Writes data out to a set location to complete the ETL process.

Appendix B: FME Server

Security

\\yourfmeserver\[REPOSITORYFOLDER]\ - All FME users will have read/write access to repository folders that they have access to. This folder will allow users to keep data on the server and access it via unc path.

\\yourfmeserver\[REPOSITORYFOLDER]\SDE - All FME users will have read access to repository folders that they have access to. This folder will consist of SDE connection files.

Permissions will be granted via the “Advanced Sharing” functionality of Windows. Security will be locked down via the security tab.

Version Control

Version control is set up on FME server and should be leveraged whenever updates are made to jobs, so they can be rolled back if needed. Versions will automatically be committed once per day on Production.

Appendix C: Upgrading FME Server

The following procedures should be followed when upgrading FME Server. This process will be used to replicate the production environment into a test environment and then replicate the test environment back to production to ensure proper testing is completed.

Upgrading Test

Notes:

- If any issues are encountered through this process, ensure that they are DOCUMENTED and contact SAFE Software for mitigation or to report a bug.

Prior to Upgrading Test (a week before upgrade):

a) Notify all users with an account on FME server and others who utilize FME server jobs, that an upgrade is planned. Include outage date/time.

Prior to Upgrading Test (on day of upgrade):

a) Ensure that a copy is made of the repositories folder (C:\ProgramData\Safe Software\FME Server\repositories).
b) Run the Bulk Commit for Version Control workspace and ensure that the local history is pushed remotely to GitHub.
c) Create a backup of Test. Note that this backup only includes the components listed here.

Following Upgrading Test (on day of upgrade):

a) Restore the Test Backup. (This will restore all the jobs that are on test)
b) Configure License
c) Replicate Production
    a. Create a backup from Production. Note that this backup only includes the components listed here.
    b. Restore the Production Backup to Test. Overwrite existing jobs (because production is more accurate than test)
    c. Disable all schedules
    d. Update jobs that need to have references updated
d) Configure Version Control
    a. Create a new personal access token on GitHub
    b. Enable and configure Version Control.
    c. Fetch from Remote
e) Configure the Config files appropriately
    a. Configure FME Download Service to use https: in the fmeEngineConfig.txt file, add the ResultPrefix parameter under the section SUB_SECTION FILE_DOWNLOAD_SERVICE, on the line that starts SUCCESS_RESPONSE, and before the ResultRootDir parameter: ...|ResultPrefix=https://yourtestfmeserver.gov|ResultRootDir=...
f) Configure System Email
    a. SMTP Server: yoursmtpserver
    b. Server Port: 25

    c. Connection Security: None
    d. Email From: fmetestserver@yourdomain.gov
g) Upload Files
    a. Custom Coordinates System Definition File
    b. Custom Transformers
h) Testing
    a. Request a Test License for # of engines that match Production
    b. Enable schedules for jobs that run more than once an hour on test and disable them on production.
    c. Run jobs manually that run less than once an hour.
    d. Test to ensure that jobs run properly, produce expected results, and engines restart correctly.
    e. Document any issues experienced during testing and report to FME.
    f. Once completed with testing, disable schedules on test, and re-enable schedules on production.

Upgrading Production

Prior to Upgrading Production (a week before upgrade):

a) Notify all users with an account on FME server and others who utilize FME server jobs, that an upgrade is planned. Include outage date/time.

Prior to Upgrading Production (on day of upgrade):

a) Ensure that a copy is made of the repositories folder (C:\ProgramData\Safe Software\FME Server\repositories).
b) Run the Bulk Commit for Version Control workspace and ensure that the local history is pushed remotely to GitHub.
c) Create a backup from Production. Note that this backup only includes the components listed here.

Following Upgrading Production (on day of upgrade):

a) Restore the Production Backup
b) Configure License
c) Configure Version Control
    a) Create a new personal access token on GitHub
    b) Enable and configure Version Control.
    c) Fetch from Remote
d) Configure the Config files appropriately
    a) Configure FME Download Service to use https. In the fmeEngineConfig.txt file, add the ResultPrefix parameter under the section SUB_SECTION FILE_DOWNLOAD_SERVICE, on the line that starts SUCCESS_RESPONSE, and before the ResultRootDir parameter: ...|ResultPrefix=https://yourprodfmeserver.gov|ResultRootDir=...
e) Configure System Email
    a) SMTP Server: yoursmtpserver
    b) Server Port: 25

c) Connection Security: None
d) Email From: fmeserver@yourdomain.gov

f) Upload Files
a) Custom Coordinates System Definition File
b) Custom Transformers