



GE Digital

ADMS

3.15

Model Manager User Guide

Copyright

Copyright 2011-2023 General Electric Company and/or its affiliates ("GE"). All Rights Reserved. This document is the confidential and proprietary information of GE and may not be reproduced, transmitted, stored, or copied in whole or in part, or used to furnish information to others, without the prior written permission of GE.

Change History

Date	Change Description
Jan 26, 2023	ADMS 3.15: <ul style="list-style-type: none">• Editorial and formatting updates.

Contents

About This Document	vii
Purpose of This Document	vii
Who Should Use This Document	vii
For More Information.....	vii
1. Introduction	1
2. Functional Overview	3
2.1 Model Manager Work Flow	4
2.2 Model Manager Input Files.....	4
2.3 Model Manager Output Files	5
2.4 Model Manager Directory Structure	6
2.5 Processing Station Files	7
2.5.1 Processing Two Adjoining Stations	8
2.5.2 (I)SSG Status List.....	9
2.5.3 Blocked Stations List	10
2.6 Modeling Request Processing.....	11
2.7 Disaster Recovery	11
2.8 Substation Group File Design	12
2.8.1 Substation Group File Example	12
2.8.2 SsgFile XSD Schema for SSG File Validation.....	14
2.9 Substation Group History File	15
2.10 Script Objects.....	15
2.11 Substation Incremental Files.....	15
3. Model Manager Displays Overview	17
3.1 Model Manager Controls.....	17
3.1.1 Model Manager Toolbar	18
3.1.2 Substation Group Files Summary Tab.....	19
3.1.2.1 Main Substation Group Files Summary Pane	19
3.1.2.2 Contextual Menu	20
3.1.2.3 Substation Status Pane	21
3.1.2.4 Substation Items Tab.....	22
3.1.2.5 Converter Log Tab	23
3.1.2.6 Model Review Log Tab.....	24
3.1.3 Detailed Substation Group Files Summary Tab	24
3.1.4 History by Substation Tab	26
3.1.4.1 Contextual Menu	27
3.1.4.2 Substation Devices Tab	28
3.1.4.3 Converter Log History Tab	28
3.1.4.4 Model Review Log Tab	29
3.1.5 Upload History Tab	30
3.1.6 SOAP History Tab	31
3.1.7 Scada Online Model Update	33
3.1.8 Log Tab.....	34
3.1.9 Color Coding.....	34
3.1.10 Modeling Status Indicator.....	35
3.1.11 Service Connection Indicator	35

4. Configuring Model Manager	36
4.1 Folders Tab	36
4.2 Workflow Options Tab.....	38
4.2.1 SSG Processing Options	39
4.2.2 Workflow Triggers.....	40
4.2.3 General	41
4.2.4 User Interface Items	42
4.3 Input Files Tab	42
4.4 Log Tab	44
4.5 Converter Tab.....	45
4.6 Navigation Overview Builder Tab	46
4.7 Scripts Tab	47
4.7.1 Edit/New Script Object Dialog Box	47
4.8 PreInput Tab.....	49
4.8.1 SSG Pre-Processing Options	50
4.8.2 PreInput Folder Monitoring.....	50
4.8.3 Archiving	51
4.8.4 MMA Web Service Details	51
4.8.5 MMA Client Details.....	51
4.9 Model Review Tab.....	52
4.10 RESTful Interface Tab.....	55
5. Using the Model Manager	57
5.1 Starting the Model Manager.....	57
5.2 Stopping the Model Manager.....	59
5.3 Automatically Processing Substation Files.....	59
5.4 Reviewing Substation Files	60
5.4.1 Approving Substation Files	61
5.4.2 Rejecting Substation Files	62
5.4.3 Setting Status to Under Review	63
5.4.4 Reviewing the Substation File Comments.....	65
5.5 Unblocking Stations	66
5.6 Processing Files with Converter Fatal Errors	66
5.7 Partial Model Upload for ISSG Files	67
5.8 Deleting Records	68
5.9 Performing a Manual Rollback.....	69
5.10 Using Script Objects	70
5.11 Monitoring the Input Folder for Unprocessed Files	71
5.12 Reprocessing Substation Files.....	72
5.13 Incrementally Updating Station Externals Files	72
5.14 Automatically Creating SSG Files	75
5.15 Customizing Model Manager.....	76
5.15.1 Configuring Column Width and Order	76
5.15.2 Sorting Data	77
5.15.3 Saving User Preferences.....	77
6. Running the Stand-Alone Model Manager	78
6.1 Starting the Stand-Alone Client.....	78
6.2 Controls and Menus.....	79
6.3 Configuring the Stand-Alone Client	79

Appendix A. (I)SSG Validation Process	81
Appendix B. Model Manager SOAP Web Service	82
B.1 Service Messages.....	82
B.2 Debug Service Messages	87
B.2.1 Consumer Soap Message Request Fields.....	87
B.2.2 ConsumerResponse Soap Message Request Fields.....	90
B.2.3 GetJobStatus Soap Message Request Fields.....	90
B.2.4 GetJobStatusResponse Soap Message Request Fields.....	91
B.2.5 JobStatus Soap Message Fields.....	92
B.2.6 JobStatusType Enumerated Values.....	93
B.2.7 GetNetworkDataSetResync Soap Message Fields	93
B.2.8 GetNetworkDataSetResyncResponse Soap Message Fields	94
B.3 Message Schemas	95
Appendix C. Model Manager REST Web Service.....	119
C.1 Service Messages.....	119
C.2 Message Schemas for Scada Online Model Update.....	123
C.3 Model Manager REST Interface Adapter	127
C.3.1 Model Manager and TIBCO Interface.....	127

Figures

Figure 1. The Model Manager Application	1
Figure 2. Model Management Process Overview	3
Figure 3. Toolbar Button Availability	18
Figure 4. Substation Group Files Summary Tab of the Model Manager Tool	19
Figure 5. Substation Group Files Summary Contextual Menu.....	20
Figure 6. Substation Status Pane	22
Figure 7. Substation Status Pane Contextual Menu	22
Figure 8. Substation Items	23
Figure 9. Converter Log	23
Figure 10. Model Review Log.....	24
Figure 11. Detailed Substation Group Files Summary Tab	25
Figure 12. Detailed Substation Group Files Summary Tab (with Model Review Status Column)	25
Figure 13. History by Substation Tab of the Model Manager Tool	26
Figure 14. History by Substation Tab Contextual Menu.....	27
Figure 15. Model Review Log.....	29
Figure 16. Upload History Tab of the Model Manager Tool	30
Figure 17. SOAP History Tab of the Model Manager Tool.....	31
Figure 18. Scada Online Model Update Tab of the Model Manager Tool.....	33
Figure 19. Log Tab of the Model Manager Tool	34
Figure 20. Model Manager Configuration Editor Folders Tab.....	37
Figure 21. Model Manager Configuration Editor Workflow Options Tab	38
Figure 22. Model Manager Configuration Editor Input Files Tab	43
Figure 23. Model Manager Configuration Editor Log Tab.....	44
Figure 24. Model Manager Configuration Editor Converter Tab	45
Figure 25. Model Manager Configuration Editor Navigation Overview Builder Tab	46
Figure 26. Model Manager Configuration Editor Scripts Tab.....	47
Figure 27. Edit/New Script Object Dialog Box	48

Figure 28. Model Manager Configuration Editor PreInput Tab	49
Figure 29. Model Manager Configuration Editor Model Review Tab	52
Figure 30. Model Manager Configuration Editor RESTful Interface Tab	55
Figure 31. Model Manager Service Client Properties.....	57
Figure 32. Input Folder Monitoring.....	71
Figure 33. Sorting Model Manager Data.....	77
Figure 34. Model Manager Properties	78
Figure 35. The Model Manager Application	79
Figure 36. Model Manager Configuration Pop-Up.....	80
Figure 37. Model Manager - GIS Adapter Data Flow	82
Figure 38. Model Manager - External Systems Data Flow.....	119
Figure 39. Model Manager - TIBCO System Interface.....	127

Tables

Table 1. Validation Rules for Single and Multiple Station Groups Processing	81
---	----

About This Document

This document is supplied as a part of GE's ADMS product.

Purpose of This Document

This document describes the user interface and functionality of the ADMS Model Manager tool.

This document provides a functional description of the Model Manager tool and instructions for how to use the tool. The Model Manager is responsible for converting station data files and transferring the resulting model files to the appropriate folder for further review or transfer into the online ADMS servers.

Who Should Use This Document

This document is intended for engineers and modelers who use the ADMS Model Manager tool.

For More Information

For more information, refer to the following:

- *Model Manager Software Installation and Maintenance Guide*
- *Distribution Management System User Guide*
- *Substation Editor User Guide*
- *Customer List Compiler User Guide*
- *Geographic Background Compiler User Guide*
- *ADMS Modeling Overview and Converter User Guide*

1. Introduction

The Model Manager (MMA) tool performs “Model Management” for the ADMS system. The Model Manager creates station model files from collections of input files by invoking model “compilers” (Converter and Navigation Overview Builder) to create binary model files that define the distribution system base model. The resulting station model files are then stored in the appropriate folder for further review or transfer into the online ADMS servers.

The Model Manager tool can run in Automatic mode, when it constantly scans for new files, or in Manual mode.

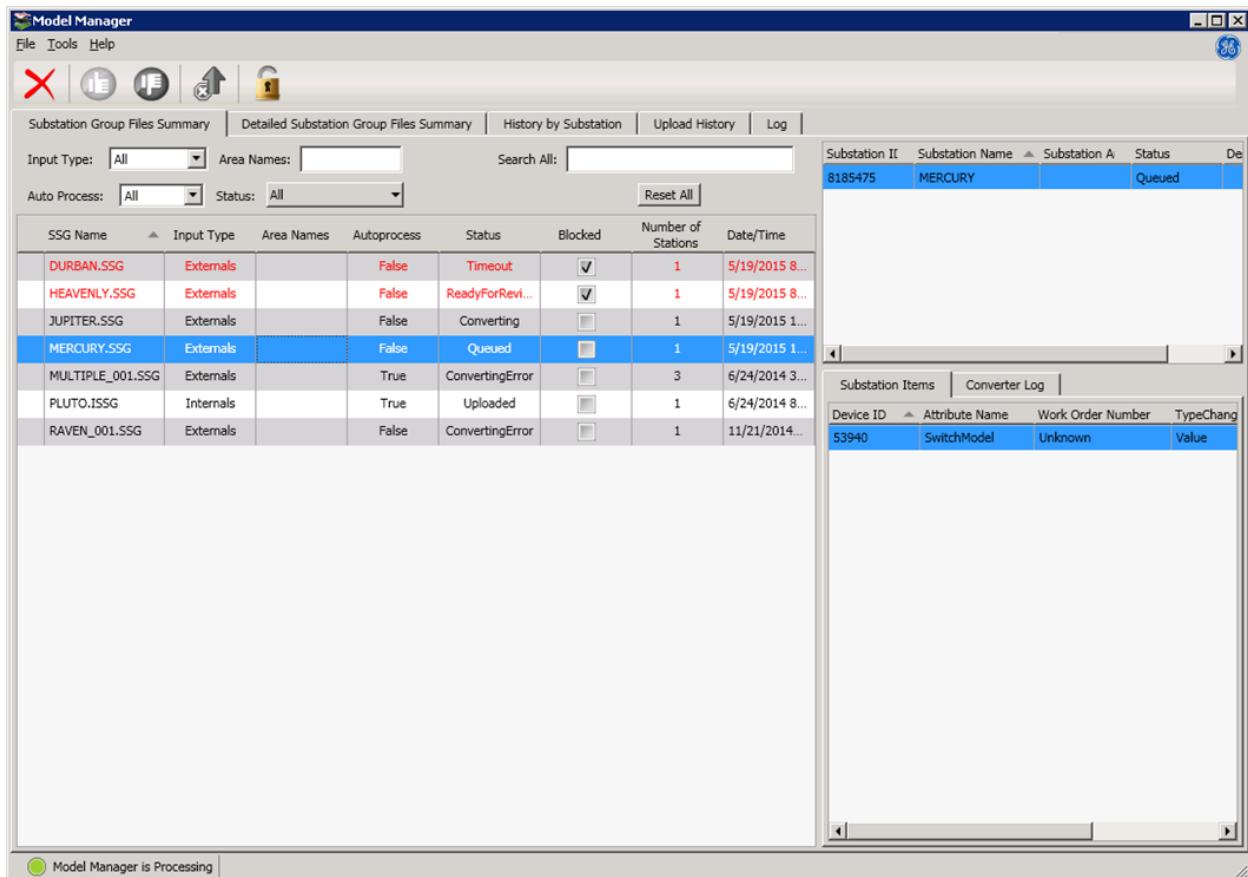


Figure 1. The Model Manager Application

The Model Manager tools consist of the following:

- **Model Manager Service:** A Windows service that runs the business logic of the Model Manager application separately from the client and connects to the Model Manager client.
- **Model Manager Service Client:** Connects to the model manager service and provides the user interface to process and review files manually.
- **Model Manager Configuration Editor:** Provides options to configure the Model Manager client.
- **Model Manager Application:** Provides a user interface similar to the Model Manager Service client user interface and allows the user to manage all of the processes manually. Unlike the

Service Client, the stand-alone client does not run automatically or automate any processes. The Configuration Editor is available to the stand-alone client via the corresponding menu option.

- **Model Manager - GISA Interface Server:** Provides a Windows Communication Foundation (WCF) based web service hosted within the Model Manager application and is responsible for interacting with the GE GIS Adapter (GISA) system.

The interface server is a component of the Model Manager's Windows service, ADMS Model Management Service, and is not hosted as a separate service. This interface is responsible for receiving Simple Object Access Protocol (SOAP) messages sent by GISA and delegating the SOAP message requests to the Model Manager application. The Model Manager interface service host can be enabled by selecting the Enable SOAP Message Handling option in the Model Manager configuration. When the Model Manager application/service is not running, the Model Manager SOAP interface also stops and any SOAP messages sent during this period are not tracked by the Model Manager. It is the responsibility of the GISA client to resend messages when the Model Manager server is back up and running. SOAP messages are securely exchanged between GISA and the Model Manager using HTTPS.

- **Model Manager - GISA Interface Adapter:** Provides an interface for sending SOAP messages from the Model Manager to GISA. This is used to send model exchange status information (SSG state) in the form of SOAP messages to the GIS interface. All asynchronous responses from the Model Manager to GIS are sent using this interface adapter.

2. Functional Overview

The Model Manager processes data from various sources, as shown in Figure 2.

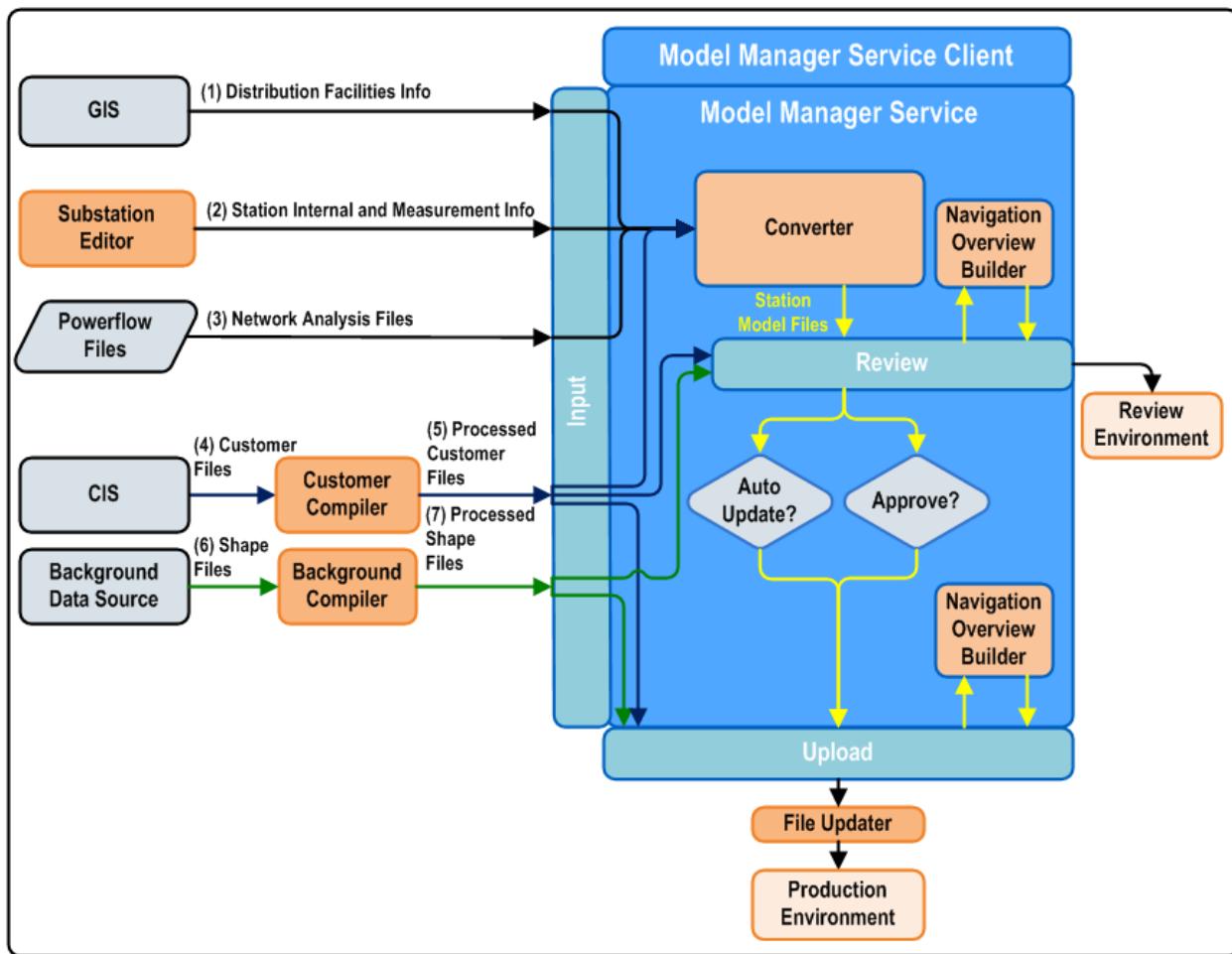


Figure 2. Model Management Process Overview

The Model Manager directly calls the Converter and the Navigation Overview Builder through the command-line interface. Although the Model Manager wraps the Converter and the Navigation Overview Builder, they can still function as standalone applications.

Note: The Navigation Overview Builder application is not part of the substation group file workflow. It runs once the resulting .mod files are copied to the Upload folder. The application builds the actual navigation overview and copies the resulting navoverview.nvw file to the Upload folder so that the file can be deployed to the ADMS client consoles.

Note: The Model Manager application does not automate the Geographic Background Compiler, the Customer Compiler, or the Substation Editor. These are the modeling engineer's responsibility.

Note: The Model Manager application also does not automate the process for reviewing the model files, the navigational overview files, the background data, or the customer file. This is the modeling engineer's responsibility.

2.1 Model Manager Work Flow

The Model Manager tool performs the following functions:

- Reads the substation group files (SSG and ISSG) and determines the stations to be converted.
- Passes the necessary attributes to the Converter in a command-line call, including station name, input file paths, and output file paths.
- Receives a completion and validation status from the Converter through the Converter's logs. If the Converter fails to complete or crashes, the Model Manager application detects it and displays this information and the validation status in the Model Manager's display tabs, including the Converter's log tab, and also logs it in the audit log.
- Copies the station model files and resulting Converter log/analyst files to the appropriate folders (see section 2.4 Model Manager Directory Structure) based on the individual station validation status, the AutoProcess flag [from the (I)SSG file], and the list of stations that must be processed together.
- If a new station file has been detected (based on a master station list stored in a file), converts the new station. After processing a pre-configured number of stations, calls the Navigation Overview Builder application. Copies the resulting navoverview.nvw file to the File Updater folder for deployment to the ADMS client consoles.
- Displays the station files that are currently being processed and their conversion status. Displays the change summary provided in the (I)SSG file.
- Receives SOAP messages from GISA, automatically creates SSG files for the substations to be exported, and processes the SSG files.
- Sends SOAP messages to GISA to update the SSG state and requests resynchronizing or re-exporting a substation from GISA.

2.2 Model Manager Input Files

The Model Manager's input data includes the following files:

- **Geographic Information System (GIS):** Provides the following topology and facility data for all feeder devices:
 - Station .xml or .csv files in the format <SubstationName>_<ID number>.xml and <SubstationName>_sub_<ID> number.csv. The extract number matches the name of a substation group file described below. The Model Manager also supports the naming pattern without the suppressed ID part. For example, <Substation Name>.xml, which matches the substation names in the SSG file or <SSG Name>_sub.csv.
 - Substation group (SSG) file in the format <SubstationName>_<ExtractID>.ssg. This file contains stations involved in the extract that should be processed together and the AutoProcess flag. The Model Manager also supports the naming pattern with the suppressed ID part. For example, <SubstationName>.ssg.

For example, if there is a load transfer between two stations, or if the opening of the point between stations is moved and/or the tie between stations is modified, the changes affect both stations. Therefore, the stations should be processed together.

Note: The same naming pattern must be used for both GIS and SSE files.

- **Substation Editor (SSE):** Provides topology and facility data for substation internal devices and other devices not modeled in the GIS.
 - Internal SSG files (ISSG). ISSG files are automatically created by the Substation Editor. When an ISSG file is moved into the Input folder, the Model Manager starts processing without the corresponding external SSG files. The analyst can choose to move the ISSG file and trigger processing without the external file or wait until the external file is available. To have a complete model, add both ISSG and SSG files into the Input folder.
 - XML internals files. The station internals files start with the station name followed by an underscore and the term “INTERNAL” (for example, “SUBSTATIONNAME_INTERNALS.xml”).
 - XML placements files. The placements files start with the station name followed by an underscore and the term “PLACEMENTS” (for example, “SUBSTATIONNAME_PLACEMENTS.xml”).
- **Customer Information System (CIS):** Provides .bpcl files preprocessed by the Customer Compiler.
- **Background File Data Source:** Provides the input binary background files (*.geo) and binary search file (*.gsch).

The Background Compiler is similar to the Converter tool. It takes a number of input files and generates binary files that can either be reviewed in the review environment or automatically pushed out to the File Updater. The background data source serves to refer to all possible sources of shape files that are input files for the Background Compiler.

- **Power Flow Files:** Required when the Converter solves the power flow as part of its validation.

The power flow files include: alg.dat, msgdef.dat, optalg.dat, phid.dat, and tftype.dat.

These files are stored in the Processing folder.

2.3 Model Manager Output Files

The output files of the Model Manager tool include the following:

- **Binary Station Model Files (*.mod):** The station model files are created by the Converter and moved to the appropriate output folder based on the validation status and the substation group file AutoProcess flag.

Model files are deployed to the ADMS clients and servers via the File Updater subsystem.
- **Converter Validation Log File:** Lists all informative, warning, and fatal error messages generated for each station processed by the Converter. The log can be viewed in the Model Manager tool.
- **Navigation Overview File (navoverview.nvw):** Provides the feeder boundary polygons. This file is re-created by the Navigation Overview Builder every time a pre-configured number of station files is added to ADMS. The navigation overview file is deployed to ADMS clients via the File Updater subsystem.

2.4 Model Manager Directory Structure

The Model Manager directory consists of the following folders:

- **Input:** The modeling engineer places new (I)SSG and station internal and placement files (XML or CSV) extracted from the GIS to this folder.

When the Model Manager tool identifies a new (I)SSG file in the Input folder, it moves this file and all relevant station files to the Processing folder before it starts the next (I)SSG file processing.

The current system navigation overview file must be placed in the Input folder by the modeling engineer.

- **Processing:** Contains files to be converted. When the Model Manager application finds new station files in the Input folder, it copies the files into the Processing folder for conversion and analysis. It also stores the power flow files.

Global.csv, DMS_Defaults.csv, and global_internals.xml must be in the Processing folder to prepare the modeling environment for work. You can also place the AutomationSchemes.csv, ProtectionSettings.csv, RegulationControl.csv, and CapacitorControlsData.csv files in the Processing folder.

If the "Auto Create SSG File for Substations if AutomationSchemes.csv File has Changed", "Auto Create SSG File for Substations if ProtectionSettings.csv File has Changed" and "Auto Create SSG File for Substations if RegulationControl.csv File has Changed" options are selected, the Model Manager creates the DevicesStationDetails.xml file with station breakers, reclosers, and regulators in the Processing folder for detecting changes in the AutomationSchemes.csv, ProtectionSettings.csv, and RegulationControl.csv files.

Note: When a CSV configuration file changes (for example, global.csv or ProtectionSettings.csv), the affected station models must be rebuilt to apply the changes.

- **Review:** Contains the current model files that have not been auto-processed, model files that did not pass validation (had one or more fatal errors), or a copy of the current auto-processed files. The modeling engineer archives files from this folder manually. The Model Manager tool keeps the latest version.

If the Model Review option is enabled, the <Station>_StationDetails.dat file with information required for the automated model review (for example, the backbone feeder length and distribution transformer count) is created in the Review folder when converting a model, and then copied to the Previous folder.

- **Upload:** Contains station model files for uploading to all ADMS clients and servers. The Model Manager tool uses model files from the Upload folder together with the just-converted .mod files to build the actual navigation overview. The Upload folder also contains the land base file, which the analyst must manually place into the folder.
- **Previous:** Each time a new station model file is successfully loaded into the real-time environment, the model file and XML files are copied from the Processing folder to the Previous folder. Therefore, the Previous folder always contains the station internals and externals files that are currently running on the system.

If the "Auto Create SSG File for Substations if AutomationSchemes.csv File has Changed", "Auto Create SSG File for Substations if ProtectionSettings.csv File has Changed", "Auto Create SSG File for Substations if RegulationControl.csv File has Changed", and "Auto Create SSG File for Substations if CapacitorControlsData.csv File has Changed" options are selected, the Model

Manager copies the AutomationSchemes.csv, ProtectionSettings.csv, RegulationControl.csv, and CapacitorControlsData files from the Processing folder to the Previous folder each time these files are updated in the Processing folder.

- **History:** Stores history about the modeling process. The History folder contains zip files exchanged from GISA or External Systems, rejected SSG files, SOAP message history information, substation to GISA instance mapping information, and Scada Online Model Update information.

If the "Enable Model Timeline Log Creation" option is selected, the model conversion, automated review, and upload summary information (including timestamps, file names, total feeder length, and device counts) is recorded in the <Station>_ModelTimeLine.csv file. The <Station>_ModelTimeLine.csv file is created in the "Model_TimeLine" folder under the History folder for every successful model upload per substation.

- **Backup:** Contains separate archives with input files for each uploaded model. Also contains a folder with automatically rejected SSG files, and another folder with zip archives received from the GIS Adapter.
- **PreInput:** The PreInput folder contains zip files that exchanged between the GISA system. A zip file contains a set of Substation Externals (*.xml) files.

When the Model Manager receives a SOAP message from GISA, the Model Manager can either receive the zip files directly in the PreInput folder or initiate an SFTP file transfer of the zip files from the GISA Model Exchange folder.

The Model Manager can automatically create SSG file(s) for the substations exported from GISA and initiate the SSG workflow process. The PreInput folder is mandatory for automatically creating SSG files by the Model Manager.

The Model Manager environment uses FTP or SFTP to transfer station model files, land base files, and navigation overview files (for the client) to a folder for the File Updater application.

2.5 Processing Station Files

The Model Manager processes the station files in the following order:

1. Checks the Input folder for an (I)SSG file.

The Model Manager starts processing when new (I)SSG files are added to the Input folder (for descriptions of the Model Manager triggering options, see section 4.2 Workflow Options Tab).

2. Reads the (I)SSG file and validates that the directory contains all of the station files listed in the (I)SSG. If station files are missing, the processing status is changed to Timeout. If all files are provided, the Model Manager moves the (I)SSG and the station files to the Processing folder.

Note: The Model Manager overwrites any files with the same name in the Processing folder.

Tip: The processing status, including error details, can be reviewed on the Log tab.

3. Launches the Converter tool for the station files residing in the Processing folder.

If any of the required files (global.csv, DMS_Defaults.csv, and global_internals.xml) is missing in the Processing folder, the Converter fails to produce the station model, and the status is changed to Converting Error. You can add missing files and restart processing using the Reprocess Substation Group(s) from Processing option.

Note: If the new (I)SSG file has a name that already exists in the history, the Model Manager tool overwrites the history and starts processing for the new (I)SSG file.

Note: If you use the ampersand (&) in a substation name, the (I)SSG file validation fails. The ampersand is a reserved character in XML. To resolve the issue, replace the ampersand character in a substation name (for example, <Name>BAPTIST HOSP T&D</Name>) with the escape sequence "&" (for example, <Name>BAPTIST HOSP T&D</Name>).

If the Use Extract ID Scheme for the SSG File Name configuration option is checked and the XML or CSV file ends with _<ExtractId>, the process removes the _<ExtractId> portion of any XML or CSV file. The _<ExtractId> portion of the SSG file is not removed in either case. For instructions on how to set this configuration option, see section 4.2 Workflow Options Tab.

The Model Manager checks for the (I)SSG Automated Model Review status to determine whether a station model should be manually reviewed. For more details about the automated review option and the review criteria, see section 4.2 Workflow Options Tab.

The Model Manager checks for the AutoProcess flag in the (I)SSG file to determine if the station model should be automatically uploaded to the online ADMS server.

If the AutoProcess flag is set to True and the Converter validation passes, the Model Manager copies the output files to the Review folder and moves the output files to the Upload folder. This keeps the processed files in the Processing folder until the Model Manager starts processing the next files.

If the AutoProcess flag is set to False, the Model Manager copies the output files to the Review folder. If the files are approved, the Model Manager copies the output files to the Upload folder and keeps the processed files in the Review folder until the Model Manager starts processing the next files. Then, the Model Manager copies the successfully uploaded files from the Processing folder to the Previous folder.

Note: If a fatal converting error occurs, the Model Manager tool aborts the converting process, ignores the AutoProcess flag, and copies the (I)SSG file to the Review folder.

Note: After the (I)SSG file is moved to the Review folder or the verification timeout is exceeded, all stations in this file have the "Blocked" status. No (I)SSG files containing any "Blocked" stations can be processed. For more information about the "Blocked" status and how to unblock stations, see section 2.5.3 Blocked Stations List.

4. Creates an archive with input files of uploaded models in the Backup folder.
5. After processing a pre-configured number of stations, calls the Navigation Overview Builder application to create a navigation overview file of stations stored in the Upload folder.

2.5.1 Processing Two Adjoining Stations

If the model change affects two adjoining stations A and B (for example, the change of the tie switch between two stations), both stations must be processed together.

If a fatal error occurs during the model processing of station A, and the processing of station B is successful, neither the A nor the B station is placed in the Upload folder automatically.

If the station A fatal errors are fixed, both A and B stations must be processed together again, regardless of the previous successful model processing of station B.

2.5.2 (I)SSG Status List

The following (I)SSG file processing statuses are available:

- **Missed:** Station files not found while verifying input data.
- **Queued:** The file is in the queue for processing.
- **VerifyingInputData:** Input data validation is in progress.
- **InputValid:** The input file's data is successfully validated for completeness.
- **Converting:** The file is being processed by the Converter tool.
- **Converted:** The file is successfully converted by the Converter tool.
- **ReadyForReview:** The file is ready for review.
- **Review:** The substation file is in review (substation validation status).
- **UnderReview:** The file is set to Under Review.
- **ModelNeedsReview:** This is a blocking state that indicates the station model needs manual review. The Model Manager triggers the ModelNeedsReview (Manual Model Review) state by comparing the station model with the previous model based on the Model Review configuration parameters.
- **Uploaded:** The file has been manually or automatically approved and uploaded.
- **ModelingSettingsChanged:** This is a blocking state that indicates the Model Manager has automatically created SSG files for substations that have settings changed in the ProtectionSettings.csv or RegulationSettings.csv files. The Model Manager triggers the ModelingSettingsChanged state by comparing the ProtectionSettings.csv and RegulationSettings.csv files in the Processing folder against the files in the Previous folder.
- **UploadedWithErrors:** The file has been uploaded with Converter fatal errors in the log.
- **Timeout:** All the input files were not copied to the Input folder during the pre-defined 30-second time span.
- **Rejected:** The Ready for Review file was rejected by the user.
- **Publishing:** Model files are produced, and the converter log is validated. The .mod file is moving to the Upload folder.
- **MovingToProcessing:** The station files and the (I)SSG file are moving to the Processing folder.
- **ConverterFatals:** Indicates that fatal errors occurred in the Converter log.
- **ErrorCopyInputToProcessing:** Indicates that a problem occurred when files moved into the Processing folder. This type of error requires actions from the analyst.
- **PublishingError:** Indicates that an error occurred when files moved into the Upload folder.
- **UnableToRestore:** Workflow cannot be recovered from the file.
- **SchemaValidationFailed:** The (I)SSG file failed to pass the schema validation.
- **SSGError:** The (I)SSG file did not pass the validation procedure.
- **ConvertingError:** The file did not pass validation by the Converter tool.

2.5.3 Blocked Stations List

When the processed (I)SSG files are moved to the review environment or time out, the Model Manager records the substation from these files into the Blocked Station List. No other (I)SSG files containing one or more substations from the Blocked Station List can be processed by the Model Manager.

When processing new station files, the Blocked Stations List prevents parallel processing on a single substation detected from two different (I)SSG files.

The (I)SSG records with substations in the Blocked Stations List are shown in the Model Manager client highlighted in red. Also, the History by Substation tab shows the blocked substations in red text with a beige background and with an extra check box in the Blocked column.

The screenshot displays two windows of the Model Manager application. The top window shows the 'Substation Group Files Summary' tab, where a substation named 'HEAVENLY.ISSG' is listed as 'Blocked'. The bottom window shows the 'History by Substation' tab, which lists several substation entries, some of which are marked as 'Blocked'.

Substation Name	Substation ID	Substation Area	Blocked	Current Status
HEAVENLY	HEAVENLY		<input checked="" type="checkbox"/>	Publishing
OLYMPIC	OLYMPIC		<input checked="" type="checkbox"/>	Review
BACHELOR	BACHELOR		<input type="checkbox"/>	Uploaded
COHO	COHO		<input type="checkbox"/>	ConverterFatal
MARS	MARS		<input type="checkbox"/>	Publishing
VENUS	VENUS		<input type="checkbox"/>	Uploaded

When	SSG file name	SSG Status	Substation Status	Description
10/7/2014 5:01:50 AM	HEAVENLY.ISSG	Publishing	Review	Station approved
10/7/2014 4:59:38 AM	HEAVENLY.ISSG	ReadyForReview	Review	Ready for review
10/7/2014 4:59:38 AM	HEAVENLY.ISSG	Converting	Converted	Converted
10/7/2014 4:59:10 AM	HEAVENLY.ISSG	Converting	Converting	Converting...
10/7/2014 4:59:09 AM	HEAVENLY.ISSG	MovingToProcessing	InputValid	Files moved into processing
10/7/2014 4:59:08 AM	HEAVENLY.ISSG	VerifyingInputData	InputValid	Station internals verified
10/7/2014 4:59:07 AM	HEAVENLY.ISSG	Queued	Queued	Queued for Conversion

To unblock a station:

1. Review the (I)SSG file that the station belongs to.
2. Navigate to the History by Substation tab.
3. Right-click the blocked station and select Unblock Substation. This removes the station from the Blocked Station List.

Or select the station file and click the Unblock toolbar button.

Or deselect the Blocked checkbox.

To block a station:

1. Review the (I)SSG file that the station belongs to.
2. Navigate to the History by Substation tab.
3. Right-click the station and select Block Substation. This moves the station to the Blocked Station List.

Or select the station file and click the Block toolbar button.

Or select the Blocked checkbox.

2.6 Modeling Request Processing

In the Model Manager application, all command processing is synchronized. The Model Manager processes commands only when it is in a state to receive user input. Any consecutive right-click commands are queued until the backend is in an idle state. The idle state occurs after all current states of the (I)SSG file processing have finished one step in their workflow.

For example, if you set the number of Converter threads to four in the Configuration pop-up (for a computer with a four-core CPU) and copied 100 (I)SSG files and externals into the input directory:

- The Model Manager begins to process four (I)SSG workflows.
- The workflows go into the VerifyingInputData state.
- As the workflows are processing (each workflow can take different amounts of time to finish), their statuses appear on the Substation Group Files Summary display.
- If you right-click to delete an (I)SSG record before all four workflows are processed, the (I)SSG record background is highlighted blue and “Modeling Request Pending” appears at the bottom of the window.
- After all four workflows have processed through one state, the Model Manager starts processing the next user requests.

Note: The Model Manager processes tasks in the order they are entered. If the Model Manager processes duplicate requests (for example, two identical Delete requests), it issues an error message for the failed request.

2.7 Disaster Recovery

The Model Manager stores the last successfully uploaded substation in the Previous folder. In addition, the application archives the successfully processed input files in the Backup folder and stores the zip packages for a configurable number of days to allow retrieving “last known good” model files if there are issues with the newly created substation models. Still, the Model Manager does not provide a backup of its directory structure.

When the Model Manager is in the Idle state, you can safely copy or move all files. The Model Manager’s state is determined by its input files.

The workflow information, including the workflow state, is stored in the Substation Group History files (*.ssgh) in the History folder.

When a state of the workflow is running (for example, the Model Manager is in the Converting state), the application’s file persistence is highly assured. However, if the Model Manager stops working while a file is processing, there is a small chance that it might lose track of files needed to process a given workflow state. Restarting the Model Manager could result in the UnableToRestore state. To ensure disaster recovery, it is essential to provide a backup mechanism for the Model Manager’s file system.

2.8 Substation Group File Design

The substation group (SSG) file is an XML file created by the GIS system.

The internal substation group (ISSG) file is an XML file created by the Substation Editor.

The SSG and ISSG files include the following key data required by the Model Manager:

- **AutoProcess Flag:** If the flag is set to True, this indicates that the successfully generated station model file should be copied to the Processing folder and automatically moved to the online ADMS server.

The modeling engineer sets the AutoProcess flag in the GIS. From the Model Manager perspective, it is the read-only information that cannot be modified. The AutoProcess flag and the Converter validation results determine where the generated station model file is placed: either in the Processing only or copied to the Processing and to a folder to deploy to the ADMS clients and servers via the File Updater subsystem.

- **ChangeItems:** The history changed since the last time the station was extracted from the GIS can be optionally included in the SSG file. The Model Manager displays the changed history in a tabular display.
- **Files:** This property is required for ISSG files. The Model Manager will process station internals, externals, and placement files listed in this field.

Note: If you use the ampersand (&) in a substation name, the SSG file validation fails. The ampersand is a reserved character in XML. To resolve the issue, replace the ampersand character in a substation name (for example, <Name>BAPTIST HOSP T&D</Name>) with the escape sequence "&#amp;" (for example, <Name>BAPTIST HOSP T&#amp;D</Name>).

2.8.1 Substation Group File Example

The following is an example of an external SSG file with three substations. It shows the AutoProcess flag set to True. In addition, it includes a changed item.

```
<?xml version="1.0" encoding="utf-8"?>
<SsgFile xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <AutoProcess>true</AutoProcess>
  <Substations>
    <Substation>
      <Name>3LIONS</Name>
      <ID>3LIONS</ID>
      <ChangedItems>
        <SubstationItem>
          <DeviceID>53940</DeviceID>
          <AttributeName>SwitchModel</AttributeName>
          <WorkOrderNumber>Unknown</WorkOrderNumber>
          <TypeChange>Value</TypeChange>
          <Description>SwitchModel Change</Description>
        </SubstationItem>
      </ChangedItems>
    </Substation>
    <Substation>
      <Name>BAKER</Name>
```

```

<ID>BAKER</ID>
<ChangedItems>
  <SubstationItem>
    <DeviceID>53940</DeviceID>
    <AttributeName>SwitchModel</AttributeName>
    <WorkOrderNumber>Unknown</WorkOrderNumber>
    <TypeChange>Value</TypeChange>
    <Description>SwitchModel Change</Description>
  </SubstationItem>
</ChangedItems>
</Substation>
<Substation>
  <Name>COHO</Name>
  <ID>COHO</ID>
  <ChangedItems>
    <SubstationItem>
      <DeviceID>53940</DeviceID>
      <AttributeName>SwitchModel</AttributeName>
      <WorkOrderNumber>Unknown</WorkOrderNumber>
      <TypeChange>Value</TypeChange>
      <Description>SwitchModel Change</Description>
    </SubstationItem>
  </ChangedItems>
</Substation>
</Substations>
</SsgFile>

```

The following is an example of an internal ISSG file. It shows the AutoProcess flag set to False. In addition, it includes the list of files that are required for the processing.

```

<?xml version="1.0"?>
<SsgFile xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <ID>9aea75fa-6bb1-4fca-8cf5-2e6f3b586452</ID>
  <AreaNames />
  <AutoProcess>false</AutoProcess>
  <DateTime>2014-10-07T04:53:53.5685752-07:00</DateTime>
  <TypeOfInput>Internals</TypeOfInput>
  <Substations>
    <Substation>
      <Name>RAVEN</Name>
      <ID>RAVEN</ID>
      <AreaName />
      <Description />
      <ChangedItems />
    </Substation>
  </Substations>
  <Files>
    <string>RAVEN_INTERNALS.XML</string>
    <string>RAVEN_PLACEMENTS_RAVEN.XML</string>
    <string>RAVEN_PLACEMENTS_RAVEN_AUTOMATED.XML</string>
    <string>RAVEN_PLACEMENTS_RAVEN_CLOSELOOP.XML</string>
  </Files>
</SsgFile>

```

2.8.2 SsgFile XSD Schema for SSG File Validation

The XSD schema checks the validity of the SSG file build.

```
<?xml version="1.0" encoding="utf-8"?>
<xss:schema xmlns:xss="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
  <xss:element name="SsgFile" type="SsgFile" nillable="false"/>
  <xss:complexType name="SsgFile">
    <xss:all>
      <xss:element minOccurs="0" maxOccurs="1" name="ID" type="xs:string" />
      <xss:element minOccurs="0" maxOccurs="1" name="AreaNames" type="xs:string" />
      <xss:element minOccurs="0" maxOccurs="1" name="AutoProcess">
        <xss:simpleType>
          <xss:restriction base="xs:string">
            <xss:pattern value="^ *(true|True|TRUE|false|False|FALSE|1|0) *$"/>
          </xss:restriction>
        </xss:simpleType>
      </xss:element>
      <xss:element minOccurs="0" maxOccurs="1" name="DateTime" type="xs:dateTime" />
      <xss:element minOccurs="0" maxOccurs="1" name="TypeOfInput" type="TypeOfInput" />
    </xss:all>
    <xss:element minOccurs="1" maxOccurs="1" name="Substations" type="ArrayOfSubstation" />
    <xss:element minOccurs="0" maxOccurs="1" name="Files" type="ArrayOfString" />
  </xss:complexType>
  <xss:complexType name="ArrayOfString">
    <xss:sequence>
      <xss:element minOccurs="0" maxOccurs="unbounded" name="string" type="xs:string" />
    </xss:sequence>
  </xss:complexType>
  <xss:simpleType name="TypeOfInput">
    <xss:restriction base="xs:string">
      <xss:enumeration value="Externals"/>
      <xss:enumeration value="Internals"/>
    </xss:restriction>
  </xss:simpleType>
  <xss:complexType name="ArrayOfSubstation">
    <xss:sequence>
      <xss:element minOccurs="1" maxOccurs="unbounded" name="Substation" type="Substation" nillable="true" />
    </xss:sequence>
  </xss:complexType>
  <xss:complexType name="Substation">
    <xss:all>
      <xss:element minOccurs="1" maxOccurs="1" name="Name" >
        <xss:simpleType>
          <xss:restriction base="xs:string">
            <xss:pattern value="[\w&amp;.!@%\(\)\-_ ]+"/> />
          </xss:restriction>
        </xss:simpleType>
      </xss:element>
      <xss:element minOccurs="0" maxOccurs="1" name="ID" type="xs:string" />
```

```

<xs:element minOccurs="0" maxOccurs="1" name="ImpactedSubstations"
type="ArrayOfSubstationItem" />
    <xs:element minOccurs="0" maxOccurs="1" name="Description" type="xs:string" />
    <xs:element minOccurs="0" maxOccurs="1" name="AreaName" type="xs:string" />
    <xs:element minOccurs="0" maxOccurs="1" name="ChangedItems" type="ArrayOfSubstationItem" />
</xs:all>
</xs:complexType>
<xs:complexType name="ArrayOfSubstationItem">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="SubstationItem" type="SubstationItem" nillable="true" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="SubstationItem">
    <xs:all>
        <xs:element minOccurs="0" maxOccurs="1" name="DeviceID" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="AttributeName" type="xs:string" />
    </xs:all>
    <xs:element minOccurs="0" maxOccurs="1" name="WorkOrderNumber" type="xs:string" />
    <xs:element minOccurs="0" maxOccurs="1" name="TypeChange" type="xs:string" />
    <xs:element minOccurs="0" maxOccurs="1" name="Description" type="xs:string" />
    <xs:element minOccurs="0" maxOccurs="1" name="DateTime" type="xs:dateTime" />
    </xs:all>
</xs:complexType>
</xs:schema>

```

2.9 Substation Group History File

The workflow for an SSG is saved in the Substation Group History file (*.ssgh). This file stores both the state of the current workflow for the corresponding SSG and the history for any SSG files with the same name. The Substation Group History file data allows constructing the SSG file views on both the Substation Group Files Summary tab and the History by Substation tab. The Substation Group History files are stored in the History folder.

2.10 Script Objects

Script objects make PowerShell scripts that can be triggered at various points in the model manager workflow. For more details, see sections 4.7 Scripts Tab and 5.10 Using Script Objects.

To enhance the interaction between PowerShell scripts and Model Manager, a customized host is implemented in Model Manager to execute the PowerShell script. All PowerShell script exit statements should be “\$host.SetShouldExit(XXX)”, where XXX is an integer that represents the exit code.

2.11 Substation Incremental Files

GIS Adapter (GISA) can publish full or incremental CIMDNOM Substation XML (station externals) files. An incremental file contains information about the updates to a substation model. When the Model Manager processes an incremental file, the Model Manager synchronizes the file against the master

substation file that is stored in the Model Manager Processing or Input folder and creates full CIMD NOM Substation XML files.

The Converter tool can only process full substation XML files.

For more details about processing incremental files, see section 5.13 Incrementally Updating Station Externals Files.

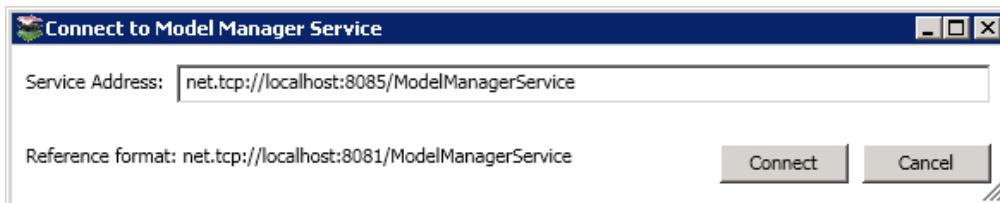
3. Model Manager Displays Overview

The Model Manager supports multiple tabbed views. This chapter describes the views, and their fields and controls.

3.1 Model Manager Controls

The Model Manager Service Client includes the following menu commands:

- **Service > Connect:** Opens the Connect to Model Manager Service dialog box, which allows the modeling engineer to manually edit the service address and connect to the service.



- **Service > Build Nav Overview:** Opens a dialog box to build the overview.nvw file in the Upload folder.
- **Service > Start Backend:** Starts the Model Manager Backend. The Model Manager Backend is the engine for processing the model manager workflows. The Model Manager Backend starts by default when the Model Manager Service starts.
- **Service > Stop Backend:** Stops the Model Manager backend. Stopping the Model Manager backend does not stop the Model Manager Service and does not break the connection to the Model Manager Service. However, if the Model Manager backend stops, the Model Manager Service will not process the model manager workflows.
- **Help > About the Model Manager:** Opens the About Model Manager dialog box with the copyright and version information.
- **Help > User's Guide:** Opens the Model Manager help.

The Model Manager application includes the following menu commands:

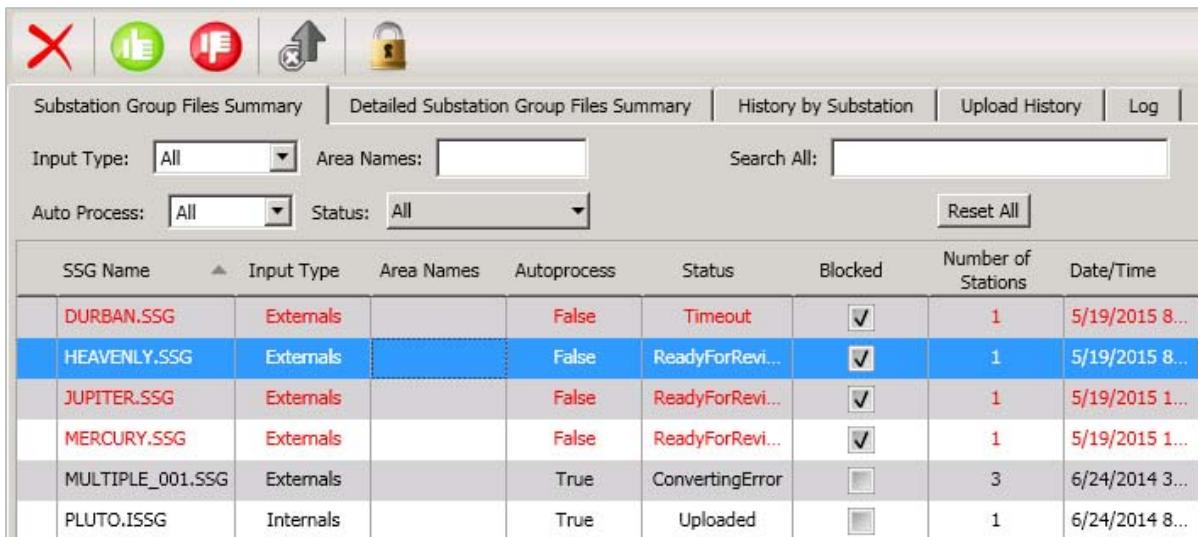
- **File > Pause:** Stops the Model Manager application. When stopped, the Model Manager will not process the model manager workflows. Pause can be used when configuring the application.
- **File > Resume:** Starts the Model Manager application.
- **File > Exit:** Closes the Model Manager application.
- **Tools > Configuration:** Opens the Model Manager Configuration Editor. For more information, refer to chapter 4 Configuring Model Manager.
- **Tools > Build Nav Overview:** Opens a dialog box to build the overview.nvw file in the Upload folder.
- **Help > About the Model Manager:** Opens the About Model Manager dialog box with the copyright and version information.
- **Help > User's Guide:** Opens the Model Manager help.

3.1.1 Model Manager Toolbar

The Model Manager toolbar is only available when the Substation Group Files Summary tab is opened. The toolbar options include:

-  Delete the selected substation group file.
-  Approve the selected substation group file.
-  Reject the selected substation group file.
-  Upload the selected substation group file with errors.
-  Unblock the selected station file and move out of the blocked station list.
-  Block the selected station file and move into the blocked station list.

Toolbar buttons are unavailable when functions are not applicable to the selected files and records. In Figure 3, the selected substation group file is ready for review. The available toolbar options for this file are Delete, Approve, Reject, and Block.



SSG Name	Input Type	Area Names	Autoprocess	Status	Blocked	Number of Stations	Date/Time
DURBAN.SSG	Externals		False	Timeout	<input checked="" type="checkbox"/>	1	5/19/2015 8...
HEAVENLY.SSG	Externals		False	ReadyForRevi...	<input checked="" type="checkbox"/>	1	5/19/2015 8...
JUPITER.SSG	Externals		False	ReadyForRevi...	<input checked="" type="checkbox"/>	1	5/19/2015 1...
MERCURY.SSG	Externals		False	ReadyForRevi...	<input checked="" type="checkbox"/>	1	5/19/2015 1...
MULTIPLE_001.SSG	Externals		True	ConvertingError	<input type="checkbox"/>	3	6/24/2014 3...
PLUTO.ISSG	Internals		True	Uploaded	<input type="checkbox"/>	1	6/24/2014 8...

Figure 3. Toolbar Button Availability

3.1.2 Substation Group Files Summary Tab

The Substation Group Files Summary tab shows the current status of the (I)SSG files processing.

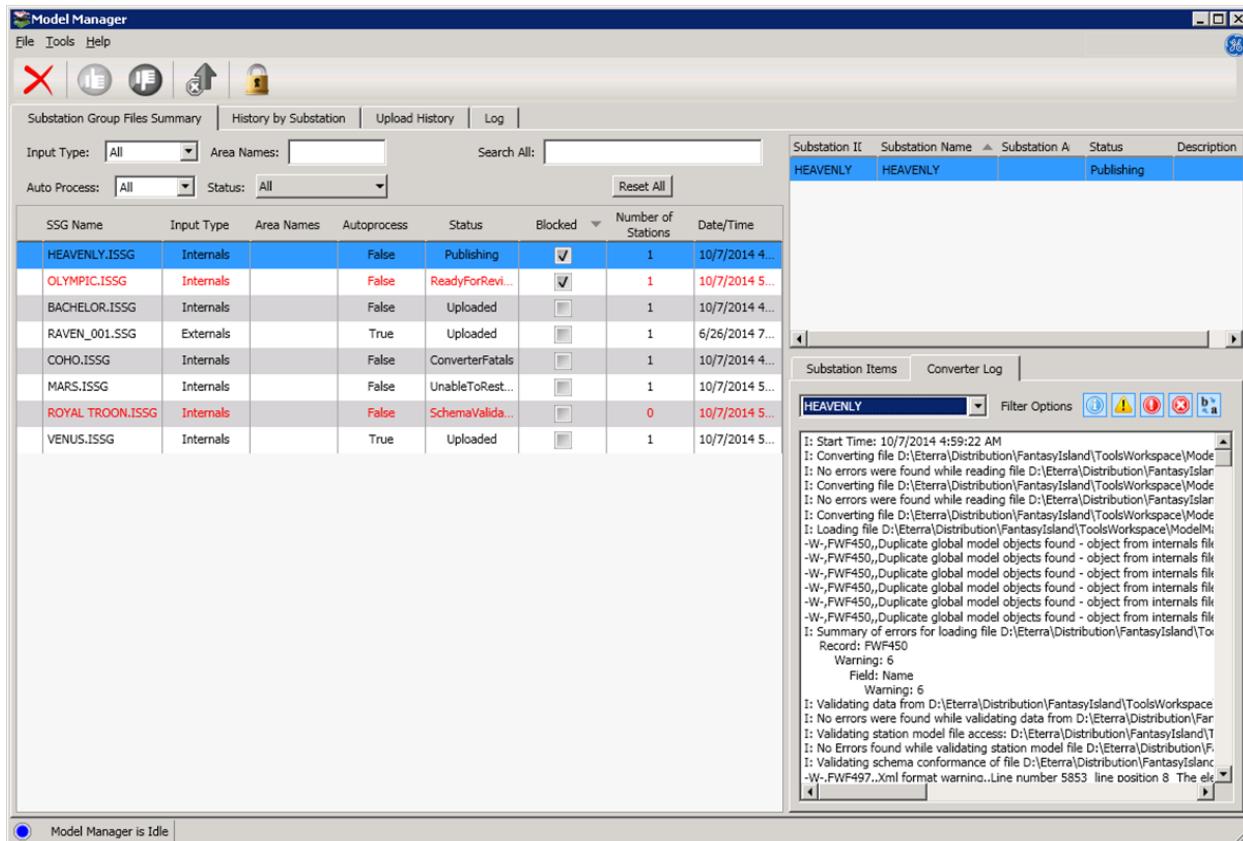


Figure 4. Substation Group Files Summary Tab of the Model Manager Tool

The Filtering pane allows you to filter data on the main Substation Group Files Summary pane:

- **Input Type:** Filter by the input type (All, Internals, or Externals).
- **Area Names:** Filter by area names.
- **AutoProcess:** Filter by the AutoProcess flag value (All, True, or False).
- **Status:** Filter by all available (I)SSG file statuses. For more information about statuses, see section 2.5.2 (I)SSG Status List.
- **Search All:** This text box provides general wild card filtering, which is common to all of the data columns.
- **Reset All:** Clicking this button clears all of the filters.

3.1.2.1 Main Substation Group Files Summary Pane

The main Substation Group Files Summary pane displays the file's processing status and allows you to delete, approve, or reject the selected records (depending on their status) via the right-click menu.

- **Rotating** (if available at row header): Shows a request is pending.
- **SSG Name:** Displays the substation group file names.
- **Input Time:** The input type of the (I)SSG file (Internal/External).
- **Area Names:** Area name.
- **AutoProcess:** Displays the AutoProcess flag state (True/False).
- **Status:** Current processing status of the (I)SSG file.
- **Blocked:** If selected, indicates that the station is in the Blocked Station List.
- **Number of Stations:** Displays the number of substations in the (I)SSG file.
- **Date/Time:** Date and time when the substation group file was processed.

3.1.2.2 Contextual Menu

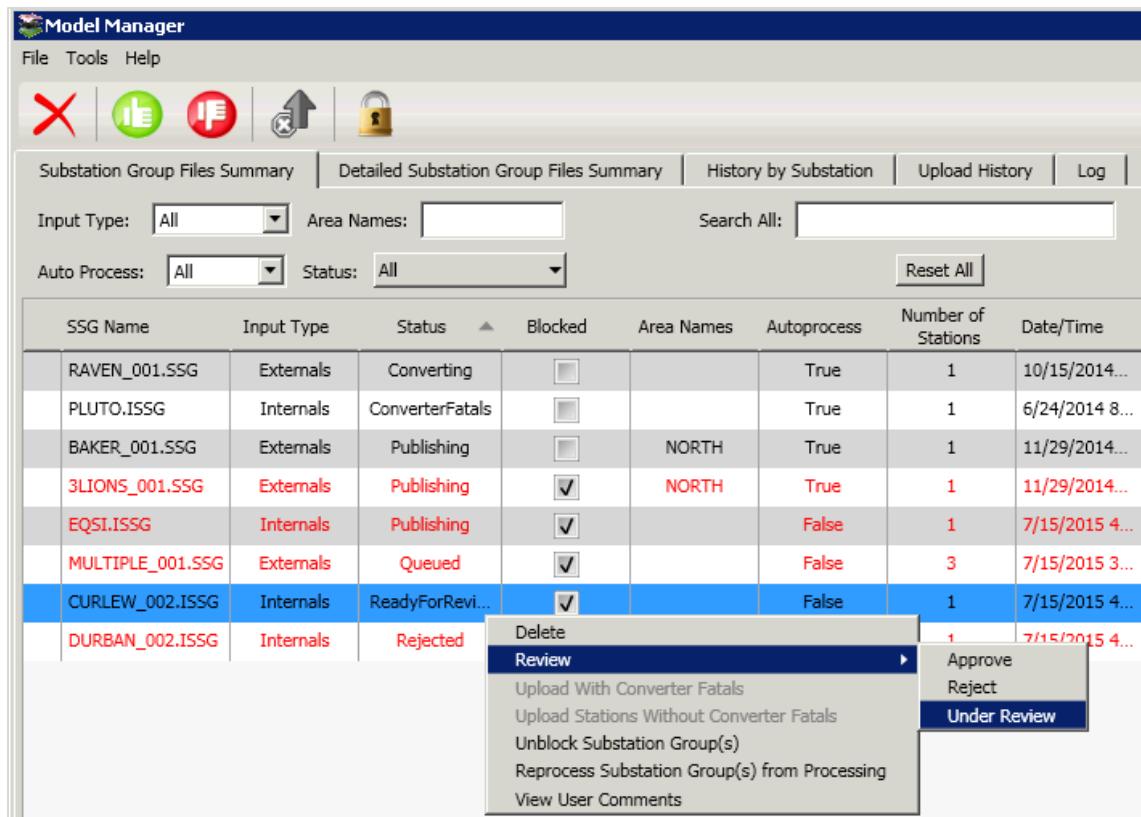


Figure 5. Substation Group Files Summary Contextual Menu

To view the contextual menu, right-click a file record on the main Substation Group Files Summary pane. The menu options include:

- **Delete:** Deletes the selected substation group file.

- **Review**
 - **Approve:** Approves the upload of the selected substation group file.
 - **Reject:** Rejects the upload of the selected substation group file.
 - **Under Review:** Indicates that the (I)SSG is being reviewed, and allows adding and reviewing comments.
- **Upload With Converter Fatal:** Uploads the output file even though it contains errors.
- **Upload Without Converter Fatal:** Uploads stations without converter fatal and blocks stations with converter fatal. This option is available only when:
 - The Partial Model Upload for ISSG option on the Workflow Options Tab of the Model Manager Configuration Editor is selected
 - Internal SSG file (ISSG) is loaded
 - The ISSG file contains more than one substation
 - The ISSG file contains both stations with converter fatal and stations that convert successfully (without converter fatal)
- **Block/Unblock Substation Group(s):** Blocks/Unblocks the selected substation group(s).
- **Reprocess Substation Group(s) from Processing:** Reprocesses the substation group using files from the Processing folder.
- **View User Comments:** Opens the SSG User Comments dialog box to review the user comments.
- **ModelingSettings:**
 - **Approve:** Approves uploading the selected substation group file.
 - **Reject:** Rejects uploading the selected substation group file.

3.1.2.3 Substation Status Pane

The Substation Status pane displays the processing status and details for substations included in the (I)SSG file selected on the Substation Group Files Summary pane.

- **Substation ID:** The substation's identification number.
- **Substation Name:** The name of the substation (for example, Baker).
- **Substation Area:** The area of the substation.
- **Status:** The status of the substation file (for example, Converted, Review, Uploaded).
- **Description:** A free-form comment from the modeling engineer who created the (I)SSG file.

Substation ID	Substation Name	Substation Area	Status	Description
BAKER	BAKER		ConverterF...	
CURLEW	CURLEW		ConverterF...	
PLUTO	PLUTO		ConverterF...	

Figure 6. Substation Status Pane

Substation ID	Substation Name	Substation Area	Status	Description
BAKER	BAKER		ConverterF...	
CURLEW	CURLEW		ConverterF...	
PLUTO	PLUTO		ConverterF...	

Figure 7. Substation Status Pane Contextual Menu

The contextual menu allows you to send a substation resynchronization (re-export) request to the GIS Adapter via the Model Manager web service. This option is only enabled when the Enable SOAP Message Handling option on the PreInput Tab of the Model Manager Configuration Editor is selected.

3.1.2.4 Substation Items Tab

The Substation Items tab displays details for the devices that were modified since the last time the stations were extracted from the GIS or since the last processing. This is optional information that may be included in an (I)SSG file.

- **Device ID:** The device's identification number.
- **Attribute Name:** Name of the device's attribute that has been changed (for example, switch model, phase).
- **Work Order Number:** Number of the work order.
- **TypeChange:** The type of change that has been made (for example, value).
- **Date/Time:** Date and time when the (I)SSG file was last modified.
- **Description:** A free-form comment from the modeling engineer who created the (I)SSG file.

Substation Items					
Device ID	Attribute Name	Work Order Number	TypeChange	Date/Time	Description
53940	SwitchModel	Unknown	Value	11/4/2016 3:17:10 AM	SwitchModel Change

Figure 8. Substation Items

3.1.2.5 Converter Log Tab

The Converter Log tab displays all the Converter processing actions.

Unlike the Converter Log tab on the History by Substation display, the Converter Log on the main display monitors the workflow process for the currently processing (I)SSG files. To display the converter log for a specific substation, select the substation from the drop-down list.

Substation Items		Converter Log	Model Review Log
3LIONS			Filter Options
I: Start Time: 1/31/2018 5:52:19 AM			

```

I: Converting file D:\Eterra\Distribution\FantasyIsland\ToolsWorkspace\ModelManagerFolders\Processing\DMS_Defaults.csv
I: No errors were found while reading file D:\Eterra\Distribution\FantasyIsland\ToolsWorkspace\ModelManagerFolders\Processing\dms_defaults.csv
I: Converting file D:\Eterra\Distribution\FantasyIsland\ToolsWorkspace\ModelManagerFolders\Processing\global.csv
-W-,FWF210,647,Data record field is not unique. This record will not be loaded.,Record name: CONDUCTOR_MODEL; Field name: NAME
I: Summary of errors for reading file D:\Eterra\Distribution\FantasyIsland\ToolsWorkspace\ModelManagerFolders\Processing\global.csv
Record: FWF210
    Warning: 1
        Field: NAME_CONDMOD
    Warning: 1
I: Converting file D:\Eterra\Distribution\FantasyIsland\ToolsWorkspace\ModelManagerFolders\Processing\global_internals.xml
I: Loading file D:\Eterra\Distribution\FantasyIsland\ToolsWorkspace\ModelManagerFolders\Processing\global_internals.xml
I: No errors were found while loading file D:\Eterra\Distribution\FantasyIsland\ToolsWorkspace\ModelManagerFolders\Processing\global_internals.xml
I: Validating data from D:\Eterra\Distribution\FantasyIsland\ToolsWorkspace\ModelManagerFolders\Processing\global.csv
-E-,FWF526,749,Capability Curve max value is less than the Generator/Energy Resource Real Power max value,Record name: EnergyResource
-E-,FWF527,750,Capability Curve max value is greater than the Generator/Energy Resource Real Power max value,Record name: EnergyResource
-E-,FWF527,751,Capability Curve max value is greater than the Generator/Energy Resource Real Power max value,Record name: EnergyResource
-E-,FWF526,752,Capability Curve max value is less than the Generator/Energy Resource Real Power max value,Record name: EnergyResource
-E-,FWF527,753,Capability Curve max value is greater than the Generator/Energy Resource Real Power max value,Record name: EnergyResource
-E-,FWF527,755,Capability Curve max value is greater than the Generator/Energy Resource Real Power max value,Record name: EnergyResource

```

Figure 9. Converter Log

The Converter Log's Filter Options toolbar allows filtering messages by type, and includes the following options:

- Show Info:** Select to show informational messages.
- Show Warnings:** Select to show warning messages.
- Show Errors:** Select to show error messages.
- Show Fatal:** Select to show fatal error messages.
- Show Replacements:** Select to show replacement value logs.

Converter log messages can be filtered using the right-click menu with the same filtering options. To view the menu, right-click anywhere on the Converter Log tab.

3.1.2.6 Model Review Log Tab

The Model Review Log tab shows details about the automated model review process. This tab only appears if the Enable Model Review option on the Workflow Options tab is enabled.

- **Station:** The station name.
- **Trigger Type:** The trigger type.
- **ID:** The feeder ID.
- **Name:** The feeder name.
- **Previous Value:** Value in the previously generated station model.
- **New Value:** Value in the currently generated station model.

Substation Items		Converter Log	Model Review Log		
Station	TriggerType	ID	Name	PreviousValue	NewValue
BAKER	Change in Feeder...	8161075	F5463	31123.4	31353.5
BAKER	Change in Custom...	8161075	F5463	1543	1548
BAKER	Change in Downst...	19725761	51189	181	187

Figure 10. Model Review Log

3.1.3 Detailed Substation Group Files Summary Tab

The Detailed Substation Group Files Summary tab appears only if the View Detailed SSG Files Summary Tab option on the Workflow Options Tab of the Model Manager Configuration Editor is selected.

This tab is similar to the Substation Group Files Summary Tab, but shows processing details for all substations in the input (I)SSG files. For example, if an (I)SSG file contains three substations, this tab will provide processing details for all of them.

In addition, the Substation Status pane contains the following columns:

- **Review Completed:** This column keeps track of substations whose review is complete.
- **Model Review Status:** This column shows whether the automated model review was performed or skipped for the substation. For the Skipped status, the reason can be found on the Log Tab of the Model Manager. The “Not Applicable” status is used in cases where the last upload was done with the disabled Model Review option, and then the feature was enabled. This column appears only if the “Enable Model Review Feature” option on the Workflow Options Tab is enabled.

The screenshot shows the 'Detailed Substation Group Files Summary' tab. At the top, there are dropdown menus for 'Input Type' (All), 'Area Names' (empty), 'Search All' (empty), and 'Auto Process' (All). Below this is a table with columns: Substation Name, SSG Name, Input Type, Area Names, Autoprocess, Status, Blocked, and Date/Time. The table contains rows for multiple substation files like 3LIONS, BAKER, COHO, DURBAN, HEAVENLY, JUPITER, MERCURY, PLUTO, and RAVEN. To the right of the main table is a smaller table titled 'Substation Items' with columns: Substation ID, Substation Name, Substation Ar, Status, Review Comp, Model Review, and Description. It lists entries for 3LIONS, BAKER, and COHO. A separate window titled 'Substation Items' is also visible.

Figure 11. Detailed Substation Group Files Summary Tab

This screenshot shows the same interface as Figure 11, but with an additional column 'Model Review' added to the main table's header. The table now includes columns: Substation Name, SSG Name, Input Type, Area Names, Autoprocess, Status, Blocked, Date/Time, and Model Review. The data remains the same as in Figure 11, with the new column providing a status for each entry.

Figure 12. Detailed Substation Group Files Summary Tab (with Model Review Status Column)

3.1.4 History by Substation Tab

The History by Substation tab presents the processing status and conversion summary of (I)SSG files from the substation perspective.

The screenshot shows the Model Manager Tool interface with the 'History by Substation' tab selected. The main pane displays a table of substation processing history. The columns are: When, SSG file name, SSG Status, Substation Status, and Description. Below this is a filtering pane with fields for Substation Name and Substation Area, and a 'Reset All' button. To the right is a smaller pane titled 'Substation Devices' showing a single entry for a device ID and attribute name. At the bottom left is a status indicator: 'Model Manager is Idle' with a blue circular icon.

When	SSG file name	SSG Status	Substation Status	Description
4/10/2015 6:24:07 AM	MULTIPLE_001.SSG	Converting	Converted	Converted
4/10/2015 6:22:53 AM	MULTIPLE_001.SSG	Converting	Converting	Converting...
4/10/2015 6:22:50 AM	MULTIPLE_001.SSG	VerifyingInputData	InputValid	No station files found - Reprocessed
4/10/2015 6:22:49 AM	MULTIPLE_001.SSG	Queued	Queued	Queued for Conversion
4/10/2015 6:19:55 AM	MULTIPLE_001.SSG	VerifyingInputData	InputValid	Station externals found
4/10/2015 6:19:54 AM	MULTIPLE_001.SSG	Queued	Queued	Queued for Conversion

Device ID	Attribute Name	Work Order Number	TypeChange	Date/Time	Description
53940	SwitchModel	Unknown	Value	6/24/2014 3:20:50 AM	SwitchModel

Figure 13. History by Substation Tab of the Model Manager Tool

The Filtering pane allows you to filter data on the main History by Substation pane:

- **Substation Name:** Filter by substation names.
- **Substation Area:** Filter by substation area names.
- **Reset All:** Click this button to clear all of the filters.

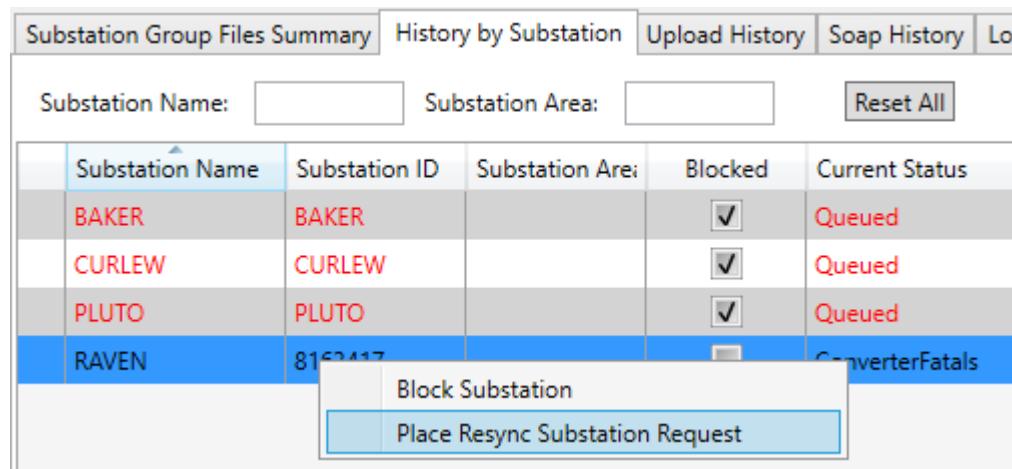
The History by Substation tab includes:

- **Rotating** (if available at row header): Shows a request is pending.
- **Substation Name:** The name of the substation (for example, Baker).
- **Substation ID:** The substation's identification number.
- **Substation Area:** The name of the area that the substation belongs to.
- **Blocked:** If selected, indicates that the station is in the Blocked Station List. A substation file can be blocked or unblocked using either this option, the corresponding toolbar button, or the right-click menu option.
- **Current Status:** Current processing status for the substation.
- **When:** Date and time when the substation group file was processed for the selected substation record.

- **SSG File Name:** Displays the substation group file name for the selected substation record.
- **SSG Status:** Current processing status of the (I)SSG file for the selected substation record.
- **Substation Status:** The status of the substation file's validation (for example, Missed, Exists) for the selected substation record.
- **Description:** A free-form comment from the modeling engineer who created the (I)SSG file.

3.1.4.1 Contextual Menu

To view the contextual menu, right-click a station file on the History by Substation tab.



The screenshot shows the 'History by Substation' tab of the Model Manager interface. At the top, there are tabs for 'Substation Group Files Summary', 'History by Substation' (which is active), 'Upload History', 'Soap History', and 'Log'. Below the tabs are search fields for 'Substation Name' and 'Substation Area', and a 'Reset All' button. The main area is a table with columns: Substation Name, Substation ID, Substation Area, Blocked, and Current Status. The table contains four rows for BAKER, CURLEW, PLUTO, and RAVEN. The RAVEN row is highlighted with a blue background. A contextual menu is open over the RAVEN row, listing 'Block Substation' and 'Place Resync Substation Request'.

	Substation Name	Substation ID	Substation Area	Blocked	Current Status
	BAKER	BAKER		<input checked="" type="checkbox"/>	Queued
	CURLEW	CURLEW		<input checked="" type="checkbox"/>	Queued
	PLUTO	PLUTO		<input checked="" type="checkbox"/>	Queued
	RAVEN	8162417		<input type="checkbox"/>	ConverterFatal

Figure 14. History by Substation Tab Contextual Menu

The menu options include:

- **Unblock Substation:** Unblocks the station. This option only appears for blocked stations.
- **Block Substation:** Blocks the station and moves the file to the Blocked Stations list. This option only appears for unblocked stations.
- **Place Resync Substation Request:** Sends a substation resynchronization (re-export) request to the GIS Adapter via the Model Manager web service. This option is only enabled when the Enable SOAP Message Handling option on the PreInput Tab of the Model Manager Configuration Editor is selected.

3.1.4.2 Substation Devices Tab

The Substation Devices tab displays details for the devices that were modified since the last time the station was extracted from the GIS or since the last processing. This is optional information that may be included in an (I)SSG file.

The Substation Devices tab includes:

- **Device ID:** The device's identification number.
- **Attribute Name:** Name of the device's attribute that has been changed (for example, switch model, phase).
- **Work Order Number:** Number of the work order.
- **TypeChange:** The type of change that has been made (for example, value).
- **Date/Time:** Date and time when the (I)SSG file was last modified.
- **Description:** A free-form comment from the modeling engineer who created the (I)SSG file.

Substation Devices					
Device ID	Attribute Name	Work Order Number	TypeChange	Date/Time	Description
53940	SwitchModel	Unknown	Value	6/26/2014 7:27:08 AM	SwitchModel Change

3.1.4.3 Converter Log History Tab

Unlike the Converter Log tab on the main display of the Model Manager application, the Converter Log History tab on the History by Substation display shows the archived log files for the selected station. The archived files have an appended timestamp to their names and are stored in the Previous folder. To display the converter log for a specific substation, select the substation from the combo box below the converter log.

The Converter Log's Filter Options toolbar allows filtering messages by type and includes the following options:

- Show Info:** Select to show informational messages.
- Show Warnings:** Select to show warning messages.
- Show Errors:** Select to show error messages.
- Show Fatal:** Select to show fatal error messages.
- Show Replacements:** Select to show replacement value logs.

Converter log messages can be filtered using the right-click menu with the same filtering options. To view the menu, right-click anywhere on the Converter Log tab.

The screenshot shows the 'Converter Log History' tab selected. The log window displays a series of informational (I), warning (W), and error (E) messages. Key entries include:

- I: Start Time: 9/10/2019 9:14:57 AM
- I: No errors were found while loading file D:\Eterra\Distribution\FantasyIsland\ToolsWorkspace\ModelManagerFolders\Processing\Prot...
- I: No errors were found while loading file D:\Eterra\Distribution\FantasyIsland\ToolsWorkspace\ModelManagerFolders\Processing\Regu...
- I: Converting file D:\Eterra\Distribution\FantasyIsland\ToolsWorkspace\ModelManagerFolders\Processing\DMS_Defaults.csv
- I: No errors were found while reading file D:\Eterra\Distribution\FantasyIsland\ToolsWorkspace\ModelManagerFolders\Processing\DMS...
- I: Converting file D:\Eterra\Distribution\FantasyIsland\ToolsWorkspace\ModelManagerFolders\Processing\global.csv
- W-,FWF210,649,Data record field is not unique. This record will not be loaded.,Record name: CONDUCTOR_MODEL; Field name: NAME
- I: Summary of errors for reading file D:\Eterra\Distribution\FantasyIsland\ToolsWorkspace\ModelManagerFolders\Processing\global.csv
- Record: FWF210
- Warning: 1
- Field: NAME_CONDMOD
- Warning: 1
- I: Converting file D:\Eterra\Distribution\FantasyIsland\ToolsWorkspace\ModelManagerFolders\Processing\global_internals.xml
- I: Loading file D:\Eterra\Distribution\FantasyIsland\ToolsWorkspace\ModelManagerFolders\Processing\global_internals.xml
- I: No errors were found while loading file D:\Eterra\Distribution\FantasyIsland\ToolsWorkspace\ModelManagerFolders\Processing\glob...
- I: Validating data from D:\Eterra\Distribution\FantasyIsland\ToolsWorkspace\ModelManagerFolders\Processing\global.csv
- E-,FWF526,751,Capability Curve max value is less than the Generator/Energy Resource Real Power max value,Record name: EnergyReso...
- E-,FWF527,752,Capability Curve max value is greater than the Generator/Energy Resource Real Power max value,Record name: EnergyR...
- E-,FWF527,753,Capability Curve max value is greater than the Generator/Energy Resource Real Power max value,Record name: EnergyR...
- E-,FWF526,754,Capability Curve max value is less than the Generator/Energy Resource Real Power max value,Record name: EnergyReso...
- E-,FWF527,755,Capability Curve max value is greater than the Generator/Energy Resource Real Power max value,Record name: EnergyR...
- E-,FWF527,757,Capability Curve max value is greater than the Generator/Energy Resource Real Power max value,Record name: EnergyR...
- E-,FWF526,759,Capability Curve max value is less than the Generator/Energy Resource Real Power max value,Record name: EnergyR...

3.1.4.4 Model Review Log Tab

The Model Review Log tab shows details about the automated model review process. This tab only appears if the Enable Model Review option on the Workflow Options tab is enabled.

- **Station:** The station name.
- **Trigger Type:** The trigger type.
- **ID:** The feeder ID.
- **Name:** The feeder name.
- **Previous Value:** Value in the previously generated station model.
- **New Value:** Value in the currently generated station model.

Model Review Log					
Station	TriggerType	ID	Name	PreviousValue	NewValue
BAKER	Change in Feede...	8161075	F5463	31123.4	31353.5
BAKER	Change in Custo...	8161075	F5463	1543	1548
BAKER	Change in Down...	19725761	51189	181	187

Figure 15. Model Review Log

3.1.5 Upload History Tab

The Upload History tab shows summary information about uploaded substation model files.

Upload Time	SSG Name	Area Names
10/7/2014 5:54:09 AM	RAVEN_001.SSG	
10/7/2014 5:13:28 AM	VENUS.ISSG	
10/7/2014 5:04:48 AM	BACHELOR.ISSG	

Station Name	Files	TimeStamp
RAVEN	RAVEN_001.xml	10/7/2014 5:54:09 AM
	RAVEN_INTERNALS.XML	10/7/2014 5:54:09 AM
	RAVEN_PLACEMENTS_RAVEN.XML	10/7/2014 5:54:09 AM
	RAVEN_PLACEMENTS_RAVEN_AUTOMATED.XML	10/7/2014 5:54:09 AM
	RAVEN_PLACEMENTS_RAVEN_CLOSELOOP.XML	10/7/2014 5:54:09 AM

Model Manager is Idle

Figure 16. Upload History Tab of the Model Manager Tool

The Upload History tab's options include:

- **Show Archives for the Last Days:** Shows the upload history for the specified number of days.
- **Area Names:** Filters the upload history by the specified areas.
- **Reprocess:** Click this button to reprocess the selected substation. The arrow at the right of the button opens the Backup folder, where the archived station input files are stored. For more information about the Backup folder, see section 4.1 Folders Tab.

The Upload History tab includes:

- **Upload Time:** Date and time when the substation model file was uploaded.
- **SSG Name:** Name of the (I)SSG file.
- **Area Names:** Substation area name.
- **Station Name:** Substation name. This field is only populated for SSG files.
- **Files:** Input files used to build the substation model file.
- **TimeStamp:** Upload date and time.

Note: The Upload History tab is populated based on the contents of the backup folder. If you remove a backup of a station model from the backup folder, this station will not be shown on the Upload History tab.

3.1.6 SOAP History Tab

The SOAP History tab appears only if the Enable SOAP Message Handling option on the PreInput Tab of the Model Manager Configuration Editor is selected.

The SOAP History tab shows summary information about processing different types of SOAP messages (Consumer, GetJobStatus, JobStatus, and GetNetworkDataSetResync) exchanged among the GIS Adapter and the Model Manager web service.

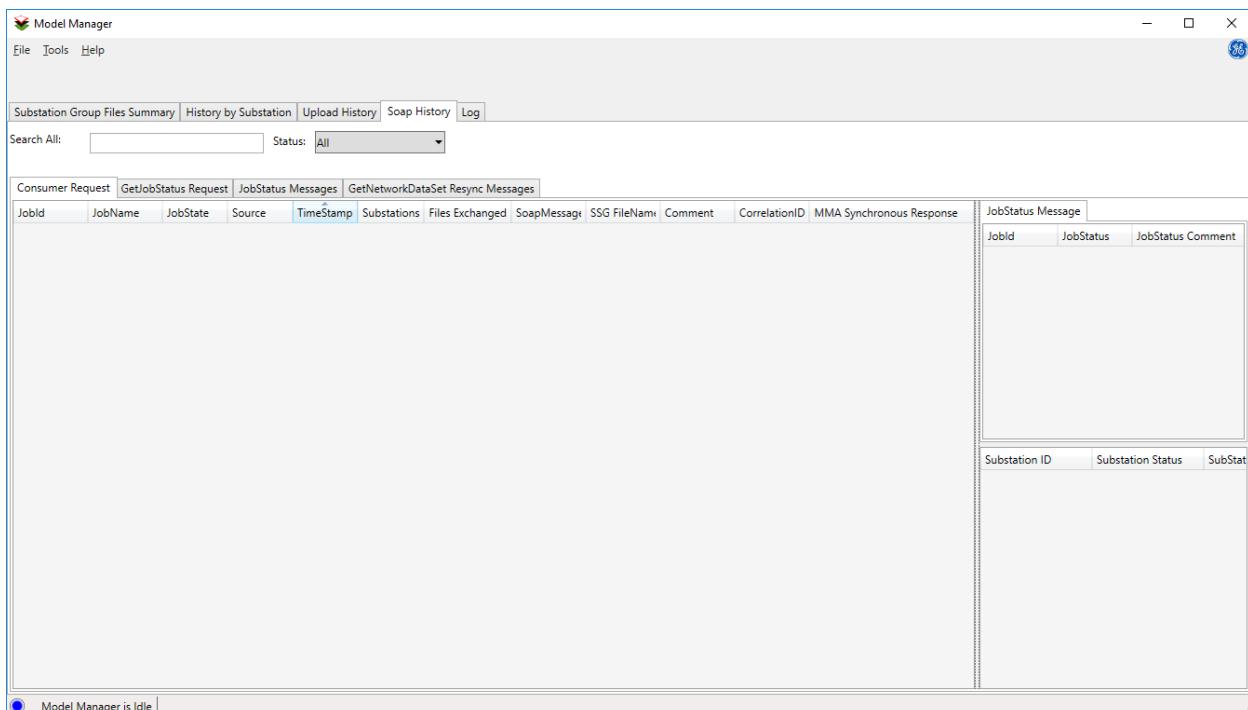


Figure 17. SOAP History Tab of the Model Manager Tool

The SOAP History tab's options include:

- **Search All:** This text box provides general wild card filtering, which is common to all data columns.
- **Status:** Filters the table by processing status (All, None, Queued, Processed, or Rejected).

The Consumer Request tab includes:

- **Job ID:** The unique identifier to reference a message exchange. This ID is derived from the message ID in the CIM metadata. The ConsumerResponse message contains this identifier.
- **Job Name:** For the Initial Load and Resynch job states, this is the station name. For the AsBuilt job state, this is the Work Order name.

- **Job State:** The job state, which can be one of the following:
 - **InitialLoad:** Station initial load, for full model data.
 - **ReExport:** Station resynchronization, for full model data.
 - **AsBuilt:** Incremental update data, for model difference data.
 - **Proposed:** Incremental update data, for model difference data in future networks.
- **Source:** Source of the message.
- **Time Stamp:** Data and time when the message exchange was performed.
- **Substations:** Substations in the message.
- **Files Exchanged:** Files exchanged.
- **Soap Message State:** Message state.
- **SSG File Name:** SSG file name.
- **Comment:** Additional information.
- **Correlation ID:** Correlation ID.
- **MMA Synchronous Response:** Specifies the synchronous response status (good, bad, none, or no response).

The GetJobStatus Request tab includes:

- **Job ID:** The unique identifier to reference a message exchange. This ID is derived from the message ID in the CIM metadata. The ConsumerResponse message contains this identifier.
- **Time Stamp:** Data and time when the message exchange was performed.
- **Data Version:** Data version of the message.
- **Source:** Source of the message.
- **Async Reply Flag:** If True, the GIS Adapter expects an asynchronous JobStatus service response message with the current job status. If False, the GIS Adapter expects the GetJobStatusResponse message.
- **Soap Message State:** Message state.
- **MMA Synchronous Response:** Specifies the synchronous response status (good, bad, none, or no response).

The JobStatus Messages tab includes:

- **Job ID:** The unique identifier to reference a message exchange. This ID is derived from the message ID in the CIM metadata. The ConsumerResponse message contains this identifier.
- **Job Status** The job status (Queued, Creating, Created, Introducing, Ready, Rejected, or Accepted).
- **Job Status Comment:** Additional information, such as a Model Manager job rejection reason.
- **Message ID:** ID of the message.
- **GISA Synchronous Response:** Specifies the synchronous response status (good, bad, none, or no response).

- **SOAP Request Message Type:** Specifies the message type (None, ConsumerRequest, or GetJobStatusRequest).
- **Comment:** Additional information.

The GetNetworkDataSet Resync Messages tab includes:

- **Message ID:** ID of the message.
- **Time Stamp:** Data and time when the message exchange was performed.
- **Source:** Source of the message.
- **Async Reply Flag:** If True, the GIS Adapter expects an asynchronous JobStatus service response message with the current job status. If False, GIS Adapter expects the GetJobStatusResponse message.
- **User ID:** ID of the user who requested data resynchronization.
- **GISA Synchronous Response:** Specifies the synchronous response status (good, bad, none, or no response).

3.1.7 Scada Online Model Update

The Scada Online Model Update tab appears only if the Enable RESTful Interface Server option on the RESTful Interface Tab of the Model Manager Configuration Editor is selected.

The Scada Online Model Update tab shows summary information about Scada model updates exchanged among the External Systems and the Model Manager over HTTP.

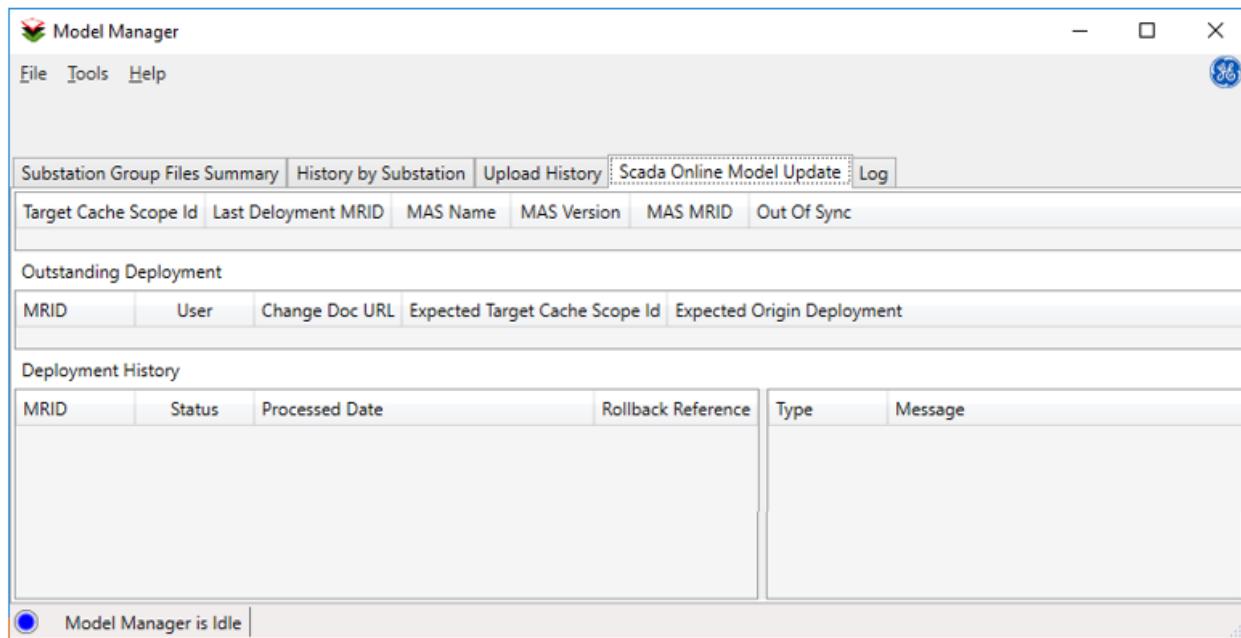


Figure 18. Scada Online Model Update Tab of the Model Manager Tool

The Scada Online Model Update tab's fields include:

- Details for the Scada Online Model Update model authority set.

- Details for Scada Online Model Update pending deployments.
- Details for Scada Online Model Update past deployments.

3.1.8 Log Tab

The Log tab displays all the informational, warning, and error messages related to the file processing along with the message timestamp and the user credentials.

Substation Group Files Summary				Detailed Substation Group Files Summary	History by Substation	Upload History	Log
Date and Time	Type	User	Message				
1/31/2018 5:51:43 AM	Info	SYSTEM	Backed up file "D:\Eterra\Distribution\FantasyIsland\ToolsWorkspace\ModelManagerFolders\Review\\3LIONS.log" to previous folder.				
1/31/2018 5:51:43 AM	Info	SYSTEM	Backed up file "D:\Eterra\Distribution\FantasyIsland\ToolsWorkspace\ModelManagerFolders\Review\\3LIONS.mod" to previous folder.				
1/31/2018 5:51:43 AM	Info	SYSTEM	Modeling State Changed: Modeling				
1/31/2018 5:51:43 AM	Info	SYSTEM	Modeling workflow queued: MULTIPLE_001.SSG State: Publishing				
1/31/2018 5:51:43 AM	Info	Hab06	Substation group unblocked: MULTIPLE_001.SSG Station 3LIONS is unblocked Station BAKER is unblocked Station COHO is unblocked				
1/31/2018 5:34:19 AM	Info	SYSTEM	Ssg file detected: D:\Eterra\Distribution\FantasyIsland\ToolsWorkspace\ModelManagerFolders\Input\\Multiple_002.ssg File Owner: BUILTIN\Administrators Last Date modified: 11/4/2016 3:17:10 AM				
1/31/2018 5:31:08 AM	Info	Hab06	Ssg file approved for publishing: MULTIPLE_001.SSG				
1/31/2018 5:28:58 AM	Info	Hab06	Overview created in Review folder				
1/31/2018 5:28:57 AM	Info	SYSTEM	Modeling State Changed: Modeling				
1/31/2018 5:28:55 AM	Warning	SYSTEM	Blocking station detected: COHO				
1/31/2018 5:28:55 AM	Warning	SYSTEM	Blocking station detected: BAKER				
1/31/2018 5:28:55 AM	Warning	SYSTEM	Blocking station detected: 3LIONS				
1/31/2018 5:28:55 AM	Error	SYSTEM	Unable to perform Model Review since StationDetails.dat file not found in Previous folder for station: COHO				
1/31/2018 5:28:55 AM	Error	SYSTEM	Unable to perform Model Review since StationDetails.dat file not found in Previous folder for station: BAKER				
1/31/2018 5:28:55 AM	Error	SYSTEM	Unable to perform Model Review since StationDetails.dat file not found in Previous folder for station: 3LIONS				
1/31/2018 5:27:48 AM	Info	SYSTEM	Converting COHO				
1/31/2018 5:26:26 AM	Info	SYSTEM	Converting BAKER				

Figure 19. Log Tab of the Model Manager Tool

3.1.9 Color Coding

The Model Manager's color-coding controls include the following:

WINGED FOOT SSG Red text on beige background indicates blocked stations. For more information about the Blocked status, see section 2.5.3 Blocked Stations List.

ROYAL TROON SSG A blue background indicates pending modeling requests.

3.1.10 Modeling Status Indicator

The modeling status indicator is located at the bottom left of the Model Manager window. It displays the current state of the Model Manager application.

- Indicates that the modeling service is idle.
- Indicates that the current modeling process is in progress.
- **Invalid Configuration:** Indicates that a configuration error has occurred. You should review the log file for error messages and fix the error using the Configuration Editor.
- **Processing Error:** Indicates that a processing error caused the Model Manager to stop processing. The processing errors may occur because of an input file with the read-only flag, or a read-only batch file. You should review the log file for error messages.

3.1.11 Service Connection Indicator

The service connection indicator is located at the bottom left of the Model Manager window. It displays the connection status of the Model Manager Service Client to the Model Manager Service.

- **Connected to Model Manager Service** Indicates the Model Manager Service Client is connected to the Model Manager Service.
- **Cannot connect to Model Manager Service** Indicates the connection with the Model Manager Service failed. You can reconnect to the Service using the Service > Connect menu option.

4. Configuring Model Manager

The Model Manager Configuration Editor allows you to set the processing options and modify the locations of the Model Manager's folders, components, and log files.

You can access the Model Manager Configuration Editor from the Windows Start menu using the following path: Eterra > Distribution > Tools > Configuration Tools > Model Manager Configuration Editor.

The changes are saved in the Model Manager configuration file, modelmanager.xml. The default location of the configuration file is indicated in the “Config. File” field at the bottom of each tab of the Model Manager Configuration Editor.

4.1 Folders Tab

The Folders tab includes paths to Model Manager folders. To specify a path to a folder, enter the path or navigate to the required folder using the Browse button . If a specified folder does not exist, the folder is created automatically when the Model Manager starts.

- **Input Folder:** The path to a folder that the Model Manager monitors for new (I)SSG files.
- **Processing Folder:** The path to a folder that the Model Manager uses for conversion and analysis.
- **Upload Folder:** The path to a folder that holds station models for deploying to all ADMS clients and services via the File Updater software.
- **Review Folder:** The path to a folder that stores the current model files that have not been auto-processed or model files that did not pass validation.
- **History Folder:** The path to a folder that stores the history of the modeling process.
- **Previous Folder:** The path to a folder that stores the last version of a station model file and its XML files from SSE and GIS. Each time a new station model file is loaded into the real-time environment, the old files are moved to the Previous folder.
- **Backup Folder:** The path to a folder that stores the backups of an (I)SSG, related station model and XML files from SSE and GIS. Each time a new (I)SSG file gets into the Uploaded state, its files are moved to a unique archive in the Backup folder.
- **PreInput Folder:** The path to a folder that stores substation external files that are published by the GIS Adapter (GISA) via the Model Manager web service. When an external file appears in the PreInput folder, the Model Manager creates an SSG file for that external file and moves the SSG file and the external XML file to the Input folder for further processing.
- **Change Doc Folder:** The path to a folder that stores Scada Online Model Update Change Document files. Each time a Change Document file appears in this folder, the Model Manager detects the file and publishes the file to Scada for deployment or further processing.

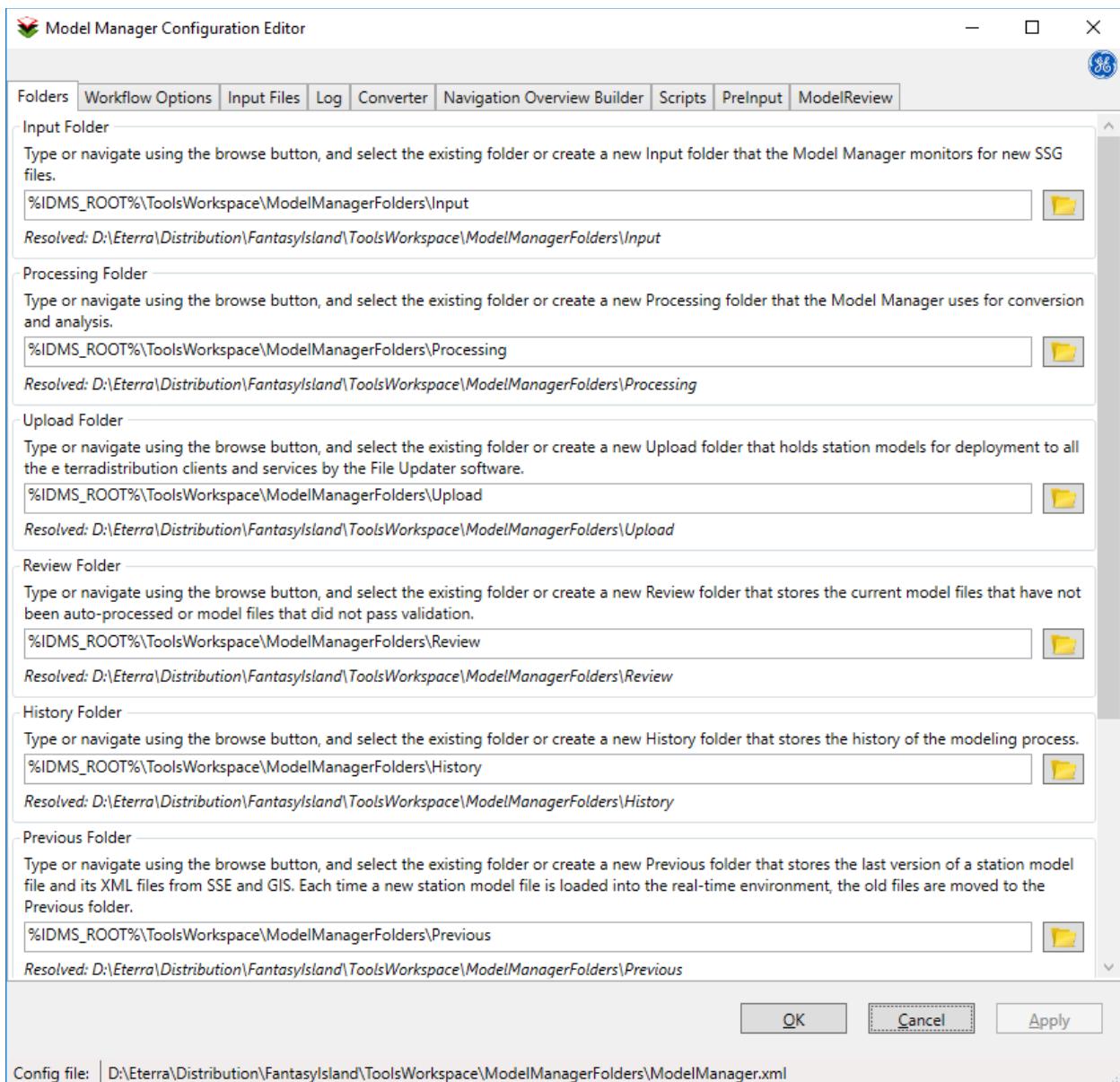


Figure 20. Model Manager Configuration Editor Folders Tab

4.2 Workflow Options Tab

The Workflow Options tab allows you to configure the Model Manager operation options.

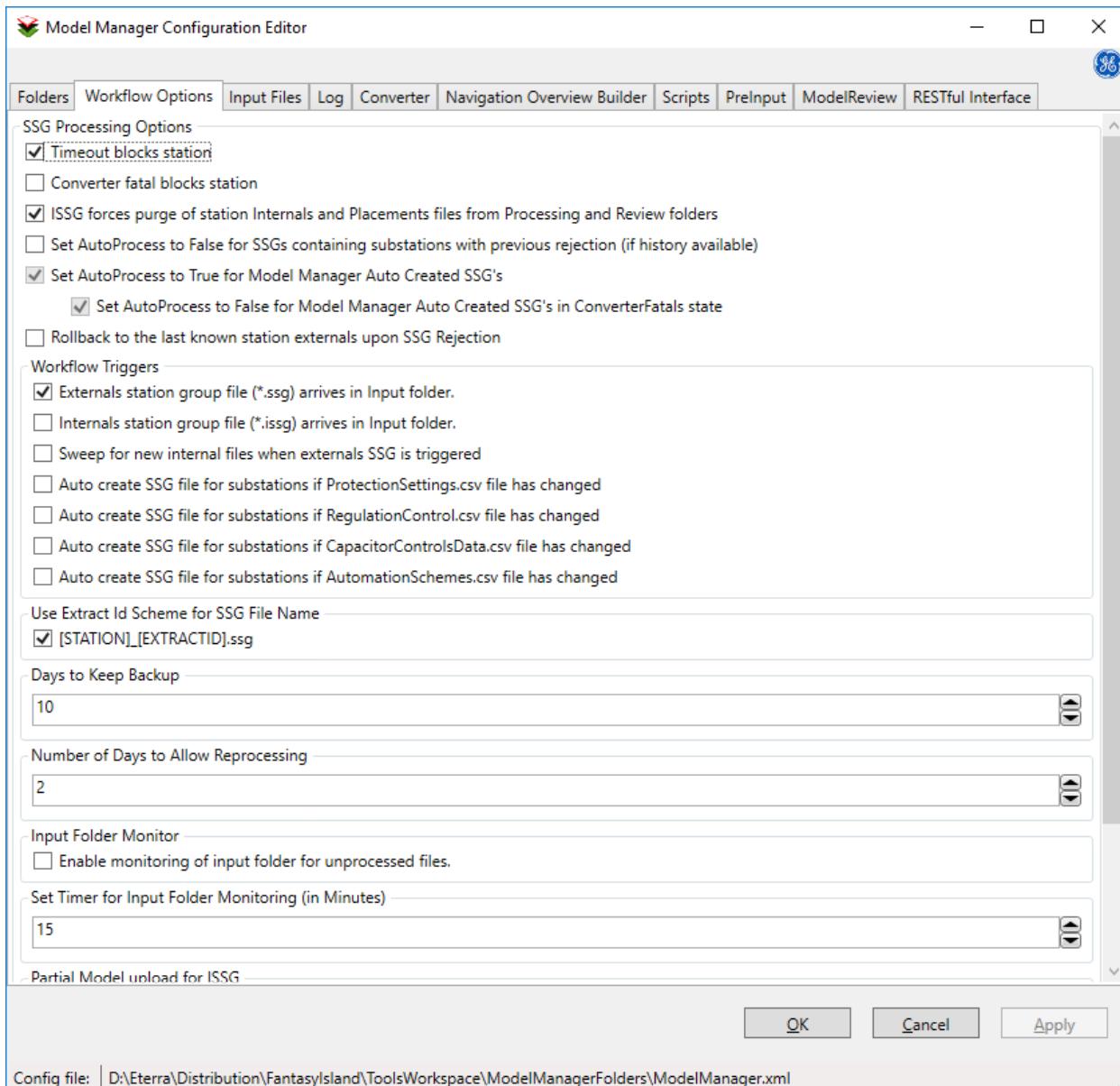


Figure 21. Model Manager Configuration Editor Workflow Options Tab

4.2.1 SSG Processing Options

SSG Processing Options	
<input checked="" type="checkbox"/>	Timeout blocks station
<input type="checkbox"/>	Converter fatal blocks station
<input checked="" type="checkbox"/>	ISSG forces purge of station Internals and Placements files from Processing and Review folders
<input type="checkbox"/>	Set AutoProcess to False for SSGs containing substations with previous rejection (if history available)
<input checked="" type="checkbox"/>	Set AutoProcess to True for Model Manager Auto Created SSG's
<input checked="" type="checkbox"/>	Set AutoProcess to False for Model Manager Auto Created SSG's in ConverterFatal state
<input type="checkbox"/>	Rollback to the last known station externals upon SSG Rejection

This section describes (I)SSG file processing options.

- **Timeout Blocks Station:** Select this check box to enable or disable blocking stations from the (I)SSG file when processing results in a timeout (by default, 30 seconds).
For more information about the Blocked status, see section 2.5.3 Blocked Stations List.
- **Converter Fatal Blocks Station:** Select this check box to block stations of the (I)SSG files that were processed by the Converter with a fatal error.
- **ISSG Forces Purge of Station Internals and Placements Files from Processing and Review Folders:** Select this check box to remove substation internals and placements files from processing and review folders before processing a substation ISSG file.
Use this option to avoid obsolete files from remaining in the work folders. This also prevents possible problems when ISSG provides only a partial set of internals and placements files.
- **Set AutoProcess to False for SSGs Containing Substations with Previous Rejection (If History Available):** If this option is selected, the Model Manager automatically changes the AutoProcess flag to False for SSG files that have previously been rejected.

Note: This option is not available for ISSG files.

- **Set AutoProcess to True for Model Manager Auto Created SSG's:** If this option is selected, the AutoProcess flag is set to True in SSG files that are automatically created by the Model Manager.
For more information about automatically creating SSG files, see section 5.14 Automatically Creating SSG Files.
- **Set AutoProcess to False for Model Manager Auto Created SSG's in ConverterFatal State:** If this option is selected, the AutoProcess flag is set to False for SSG files that were automatically created by the Model Manager when their status changes to ConverterFatal.
- **Rollback to the Last Known Station Externals upon SSG Rejection:** If this option is selected, when an SSG file in the Ready for Review or Model Needs Review state is rejected, then the associated station externals files are overwritten by the relevant station externals files from the Previous folder.

4.2.2 Workflow Triggers

Workflow Triggers

- Externals station group file (*.ssg) arrives in Input folder.
- Internals station group file (*.issg) arrives in Input folder.
- Sweep for new internal files when externals SSG is triggered
- Auto create SSG file for substations if ProtectionSettings.csv file has changed
- Auto create SSG file for substations if RegulationControl.csv file has changed
- Auto create SSG file for substations if CapacitorControlsData.csv file has changed
- Auto create SSG file for substations if AutomationSchemes.csv file has changed

This section describes Model Manager workflow trigger options.

- **Externals (*.ssg) File Arrives in Input Folder:** Select this option to run the Model Manager when SSG files appear in the Input folder.
- **Internals (*.issg) File Arrives in Input Folder:** Select this option to run the Model Manager when ISSG files appear in the Input folder.
If both options are selected, the application starts processing the first file that arrives in the Input folder.
- **Sweep for New Internal Files when Externals SSG is Triggered:** If selected, when triggered by the SSG file, the Model Manager will process station internals and placement files for substations listed in this SSG file.
- **Auto Create SSG File for Substations if ProtectionSettings.csv File has Changed:** If selected, when the ProtectionSettings.csv file changes, the Model Manager creates SSG files for substations that have protection settings changed.
- **Auto Create SSG File for Substations if RegulationControl.csv File has Changed:** If selected, when the RegulationControl.csv file changes, the Model Manager creates SSG files for substations that have regulation control settings changed.
- **Auto create SSG file for substations if CapacitorControlsData.csv file has changed:** If selected, when the CapacitorControlsData.csv file changes, the Model Manager creates SSG files for substations that have capacitor control settings changed.
- **Auto create SSG file for substations if AutomationSchemes.csv file has changed:** If selected, when the AutomationSchemes.csv file changes, the Model Manager creates SSG files for substations that have automated scheme settings changed.

The Model Manager compares the AutomationSchemes.csv, ProtectionSettings.csv, RegulationControl.csv, and CapacitorControlsData.csv files in the Processing folder against the files in the Previous folder and triggers creating SSGs for substations that are impacted by the changes. For AutomationSchemes.csv, ProtectionSettings.csv, and RegulationControl.csv comparison, a DevicesStationDetails.xml file with station breakers, reclosers, and regulators is created in the Processing folder.

The created SSG files are placed in the Input folder with the SSG state set to ModelingSettingsChanged. The model manager analyst can then approve or reject the SSG files. If approved, the substations move to the Queued state for further processing. If rejected, the SSG files move to the Rejected state.

Substation Group Files Summary							
Input Type:		Area Names:		Search All:			
Auto Process:		Status:		Reset All			
SSG Name	Input Type	Area Names	Autoprocess	Status	Blocked	Number of Stations	Date/Time
BETHPAGE.SSG	Externals		True	Uploaded	<input checked="" type="checkbox"/>	1	3/26/2019 3:14:29 AM
AUTO_BETHPAGE.SSG	Externals	NEAC	True	ModelingSettingsChanged	<input checked="" type="checkbox"/>	1	3/26/2019 3:30:20 AM

4.2.3 General

Use Extract Id Scheme for SSG File Name
 [STATION]_[EXTRACTID].ssg

Days to Keep Backup

Number of Days to Allow Reprocessing

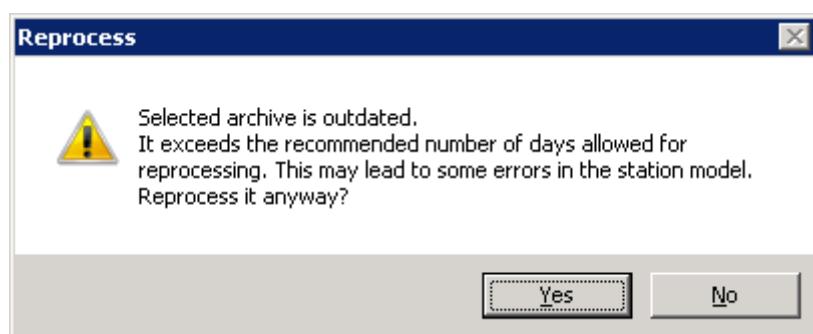
Input Folder Monitor
 Enable monitoring of input folder for unprocessed files.

Set Timer for Input Folder Monitoring (in Minutes)

Partial Model upload for ISSG
 Enable partial model upload for stations without fatal errors in ISSG.

This section describes general workflow options of the Model Manager.

- **Use Extract ID Scheme for SSG File Name:** Select this check box to process only the substation group (SSG) files in format <SubstationName>_<ExtractID>.ssg. If this check box is not selected, the Model Manager allows for the naming pattern with the suppressed ID part (for example, <SubstationName>.ssg).
- **Days to Keep Backup:** Specifies the number of days for station backup files to be kept in the backup folder.
- **Number of Days to Allow Reprocessing:** Specifies the number of days to allow station reprocessing using the backup files.



- **Input Folder Monitor:** Select this check box to enable the monitoring of the model manager input folder for station XML files that are unprocessed or waiting for an (I)SSG file. For more information, see section 5.11 Monitoring the Input Folder for Unprocessed Files.
- **Set Timer for Input Folder Monitoring (in Minutes):** Specifies the time period (in minutes) for the input folder to be monitored. The minimum value is 5 minutes and the maximum value is 120 minutes.
- **Partial Model Upload for ISSG:** Select this check box to enable partial model upload for ISSG files. If this option is selected, stations without Converter fats are uploaded, and stations with Converter fats are blocked. For more information, see section 5.7 Partial Model Upload for ISSG Files).

4.2.4 User Interface Items

User Interface Items

<input type="checkbox"/> Allow Popup for viewing and entering User Comments for SSG files under review
<input checked="" type="checkbox"/> View Detailed SSG Files Summary Tab
<input type="checkbox"/> Check Review Completed for all stations before allowing SSG to be approved

This section describes User Interface options of the Model Manager.

- **Allow Pop-Up for Viewing and Entering User Comments for SSG Files under Review:** If this option is selected, the SSG Under Review dialog box appears when you set the (I)SSG state to Under Review.
On the SSG Under Review dialog box, you can view previously added review comments and add new ones.
- **View Detailed SSG Files Summary Tab:** This option configures displaying of the Detailed Substation Group Files Summary Tab.
- **Check Review Completed for All Stations before Allowing SSG to be Approved:** Select this option to allow approving only (I)SSGs with all substation successfully reviewed (the Review Completed flag must be set).

4.3 Input Files Tab

The Input Files tab allows you to add or remove input files and choose which folders to move the files into during processing.

- **Add:** Adds a new blank record in the Input Files panel. You can then enter a new file extension and set moving actions.
- **Remove:** Removes selected items from the Input Files panel.

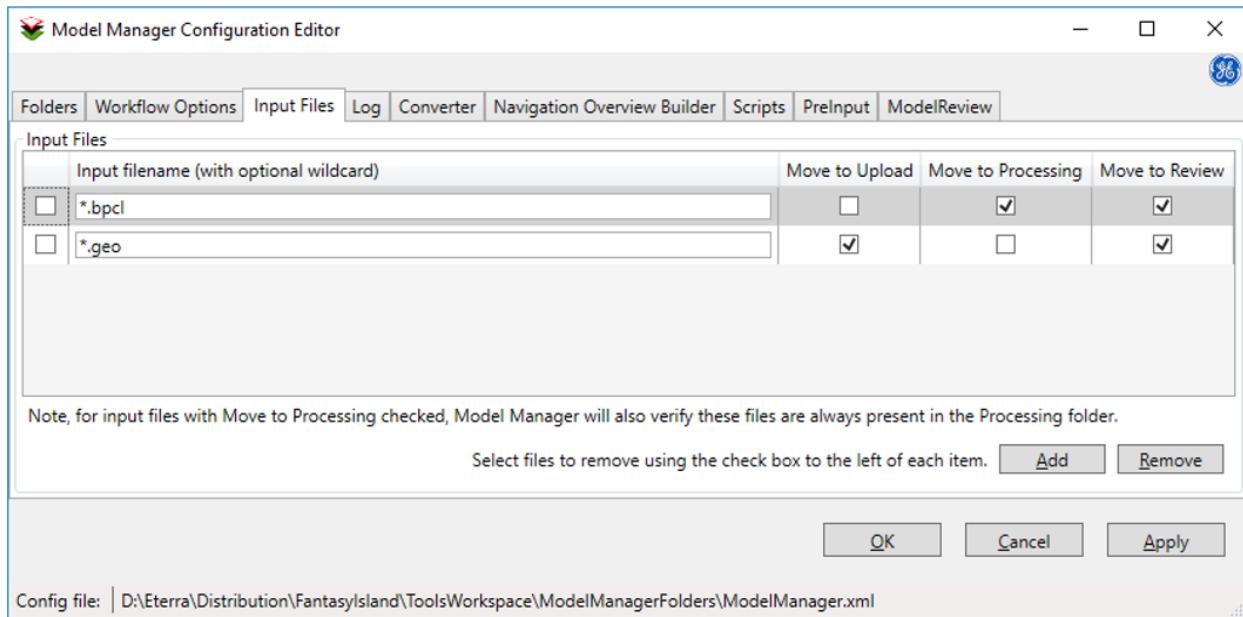


Figure 22. Model Manager Configuration Editor Input Files Tab

4.4 Log Tab

The Log tab sets the location of the log file and manages log message storage.

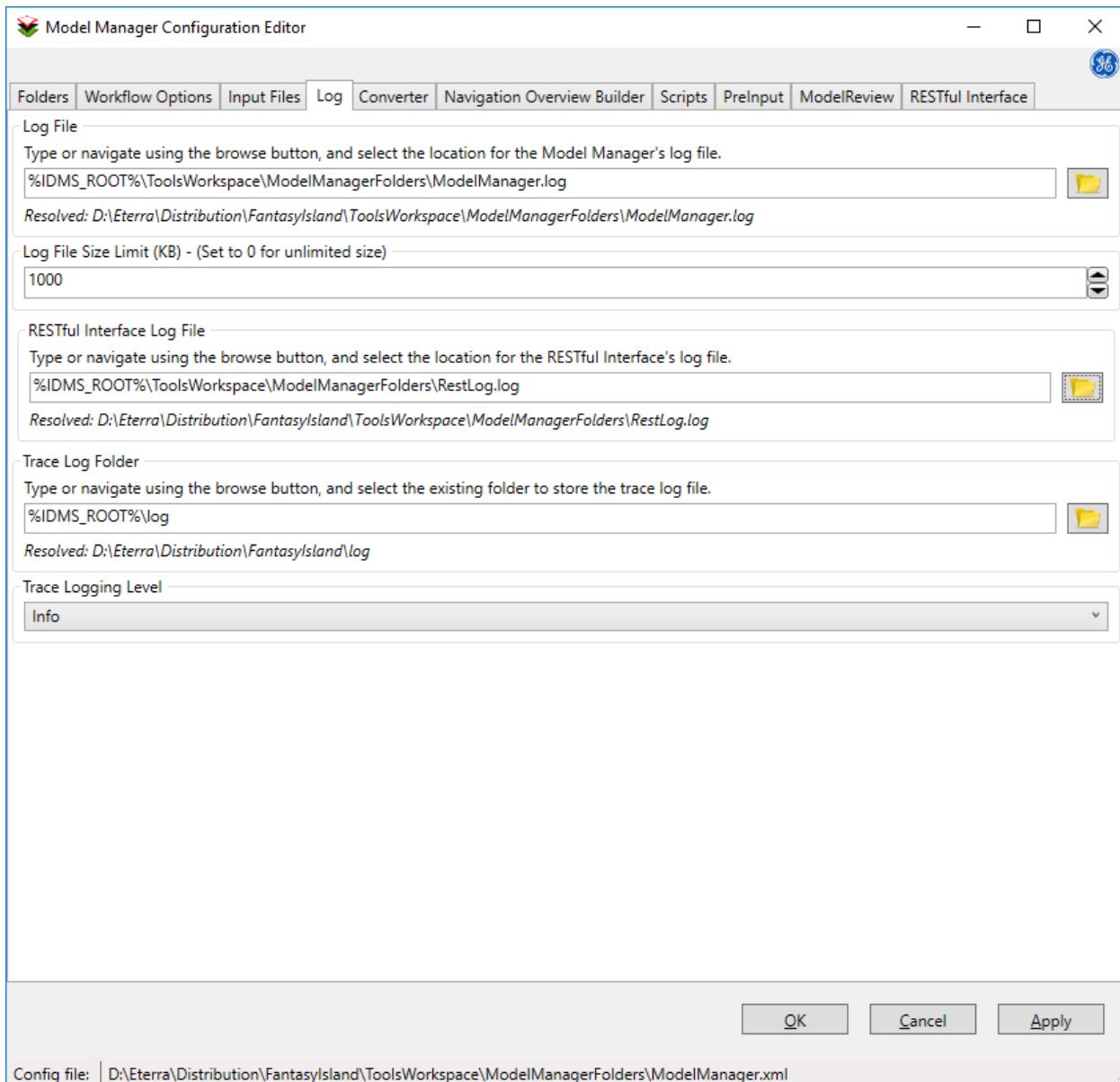


Figure 23. Model Manager Configuration Editor Log Tab

The Log tab includes:

- **Log File:** Enter the path to a folder (or navigate to a folder using the Browse button) for the Model Manager's log file.
- **Log File Size Limit:** Enter the maximum log file size in KB.

When the log file size reaches the set limit, the ModelManager.log file is saved using the following naming convention:

ModelManager_YYYYMMDDhhmmss.log

The “YYYYMMDDhhmmss” extension is a timestamp of when the file is closed. The closed log file is ready to be archived.

- **RESTful Interface Log File:** Enter the path to a folder (or navigate to a folder using the Browse button ) for the Model Manager’s REST Interface log file.
- **Trace Log Folder:** Enter the path to a folder (or navigate to a folder using the Browse button ) for the Model Manager’s trace log file.
- **Trace Logging Level:** Allows you to specify the tracing messages logging level (Fatal, Error, Warning, Info, Debug1, Debug2, Debug3, etc.). Tracing messages are used for debug purposes.

4.5 Converter Tab

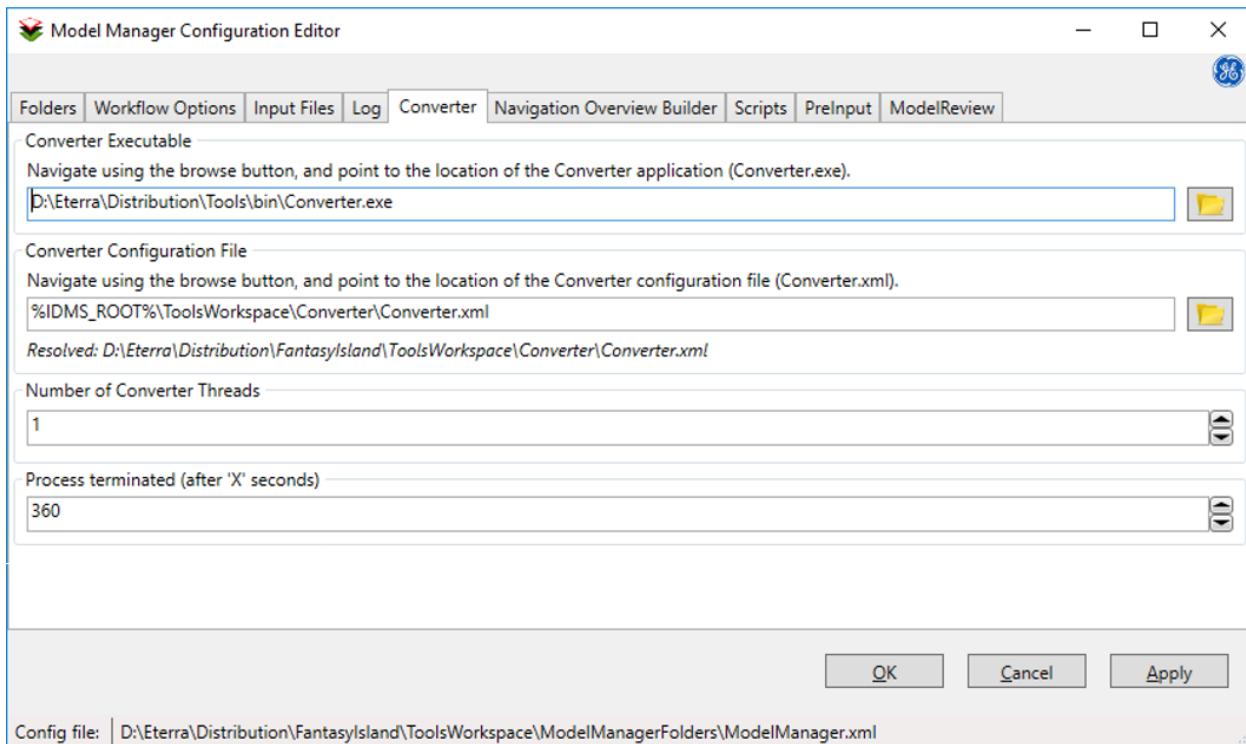


Figure 24. Model Manager Configuration Editor Converter Tab

The Converter tab includes:

- **Converter Executable:** Path to the location of the Converter application (Converter.exe).
- **Converter Configuration File:** Path to the location of the Converter application’s configuration file (Converter.xml).
- **Number of Converter Threads:** The number of workflows that the Converter can process simultaneously. For more information, see section 2.6 Modeling Request Processing.

- **Process Terminated (After X Seconds):** Sets the time (in seconds) after which the Converter processing terminates.

4.6 Navigation Overview Builder Tab

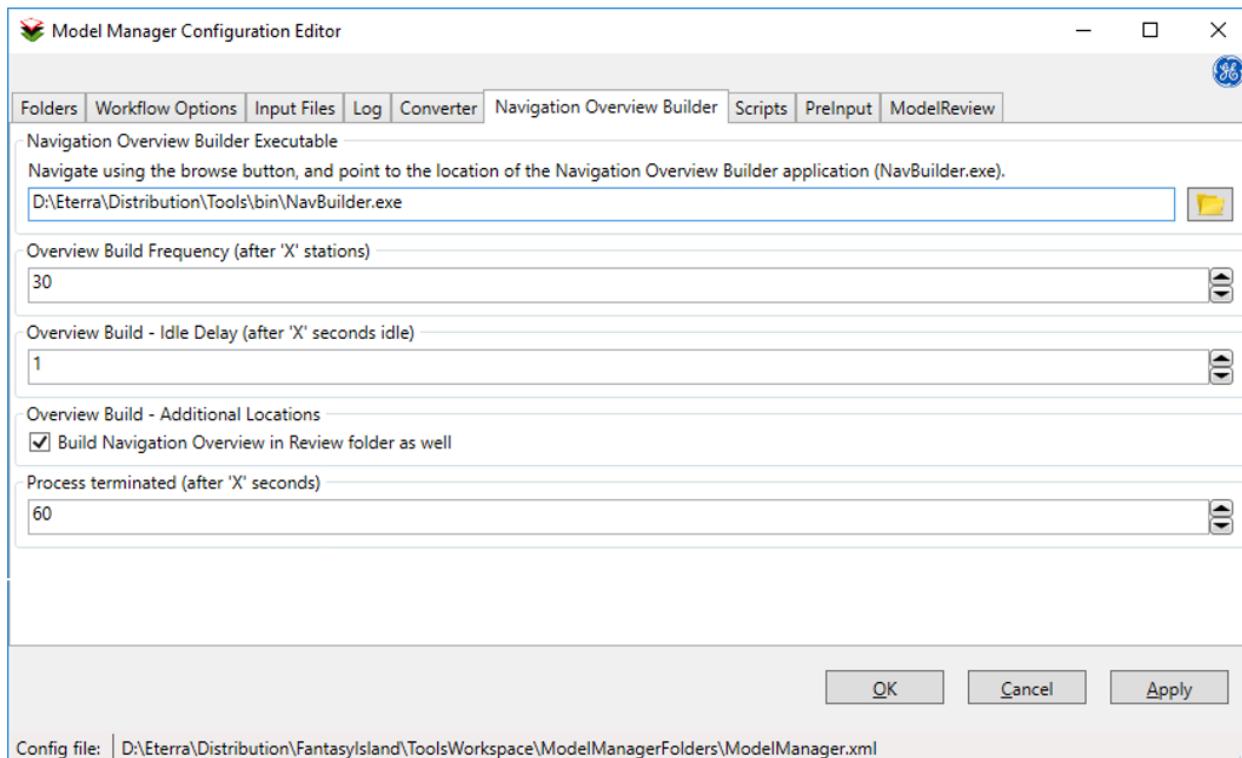


Figure 25. Model Manager Configuration Editor Navigation Overview Builder Tab

The Navigation Overview Builder tab includes:

- **Navigation Overview Builder Executable:** Path to the location of the Navigation Overview Builder application (NavBuilder.exe).
- **Overview Build Frequency (After X Stations):** The number of stations that need to be built before the Navigation Overview Builder application runs. The Model Manager interrupts (I)SSG processing to build an overview file in the Upload folder.
- **Overview Build Idle Delay (After X Seconds Idle):** When (I)SSG processing is finished, an overview file is scheduled to build after a specified time has elapsed. This allows the user to approve other (I)SSG files and complete other tasks in the Model Manager before the Navigation Overview Builder runs. After the (I)SSG files are reviewed, the timer resets to the number of seconds defined in this field.
- **Overview Build – Additional Locations:** Provides an option to build the Navigation Overview file in both the Upload and the Review folders. This option is selected by default.
- **Process Terminated (After X Seconds):** Sets the time (in seconds) after which the Navigation Overview build terminates.

4.7 Scripts Tab

The Scripts tab allows you to manage script objects, which can be used to backup files, update a customer file with an incremental file, perform command-line operations, etc. A script object monitors system events and other events, to which a PowerShell script can be attached. The attached PowerShell script executes when the listened event executes.

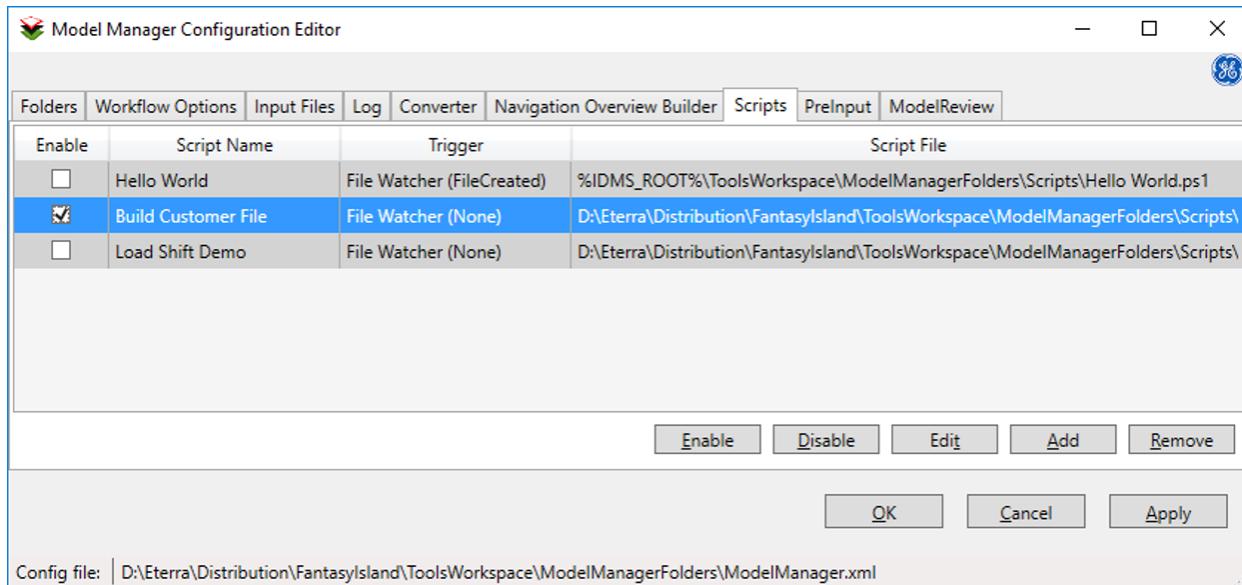


Figure 26. Model Manager Configuration Editor Scripts Tab

The Scripts tab includes the following buttons:

- **Enable:** Enables the selected script object.
- **Disable:** Disables the selected script object.
- **Edit:** Edit a selected script object. An Edit Script Object dialog box appears.
- **Add:** Add a new script object. A New Script Object dialog box appears.
- **Remove:** Remove the selected script object.

4.7.1 Edit/New Script Object Dialog Box

The Edit/New Script Object dialog box includes the following fields and buttons:

- **Name:** The name of the script object.
- **Hint:** Shows a tooltip while the cursor hovers over a script object on the Scripts tab.
- **Enable:** Activates the script object so that the script object can be triggered when a monitored event executes. If this is not selected, the script object does not listen to the trigger.
- **Synchronous:** If this option is selected, the Model Manager workflow is blocked at the time of the PowerShell script execution. If this option is not selected, the Model Manager can process files simultaneously while executing the PowerShell script.

In synchronous mode, the Model Manager waits until the script returns an exit code before executing other pending tasks.

- **Trigger:** An event can be monitored by a script object. Two types of triggers are implemented: “File Watcher” and “Conversion Status”.
- **Enable Triggering Scripts on MMA Idle State:** If this option is selected, the script is executed only after the Model Manager reaches the idle state.
 - **Script Trigger – Idle Delay:** Specifies the time period in seconds for the Model Manager to be in the idle state before executing the script. The minimum value is 1 minute, and the maximum value is 600 minutes.
- This option helps controlling the script invocation when multiple batches of (I)SSG files are arriving in short periods of time.
- **PowerShell Script:** Path to an executable PowerShell script.

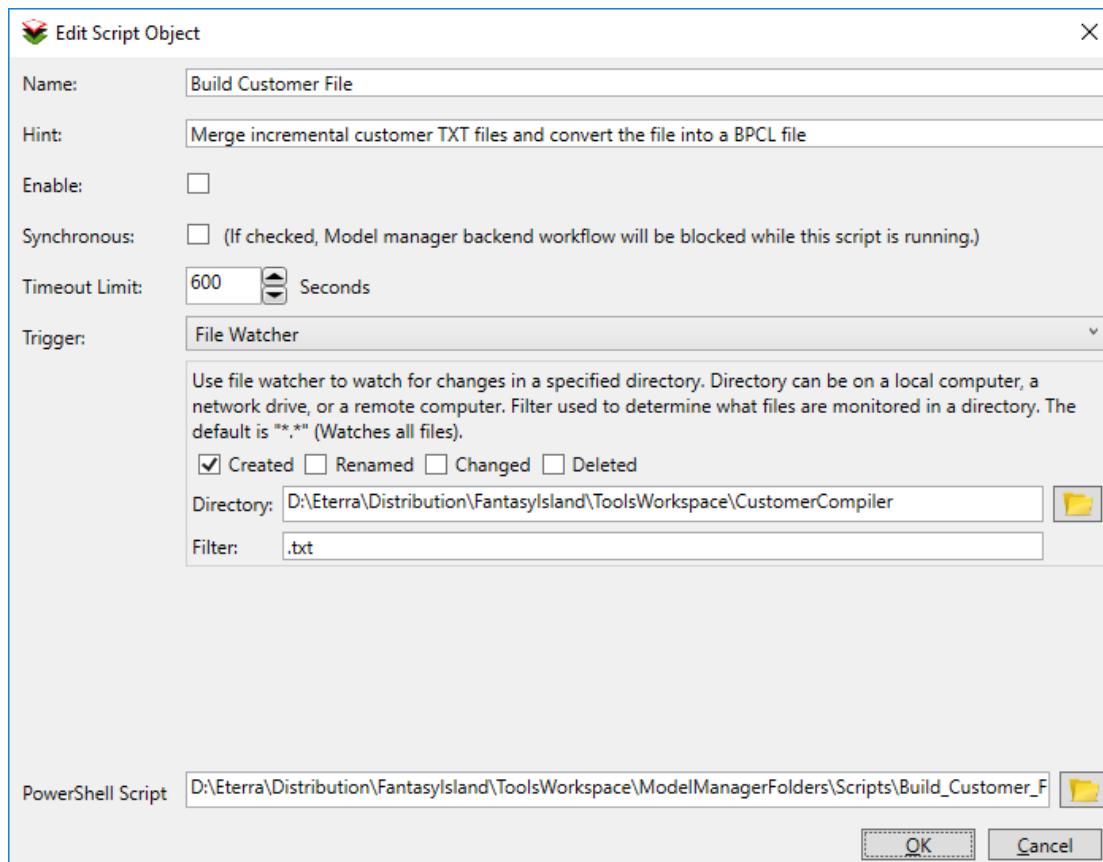


Figure 27. Edit/New Script Object Dialog Box

For more information, see section 5.10 Using Script Objects.

4.8 PreInput Tab

The PreInput tab allows you to configure the PreInput folder of the Model Manager, which is used for processing CIMDNON files that are supplied by the GIS Adapter for full or incremental updates of substation externals XML files.

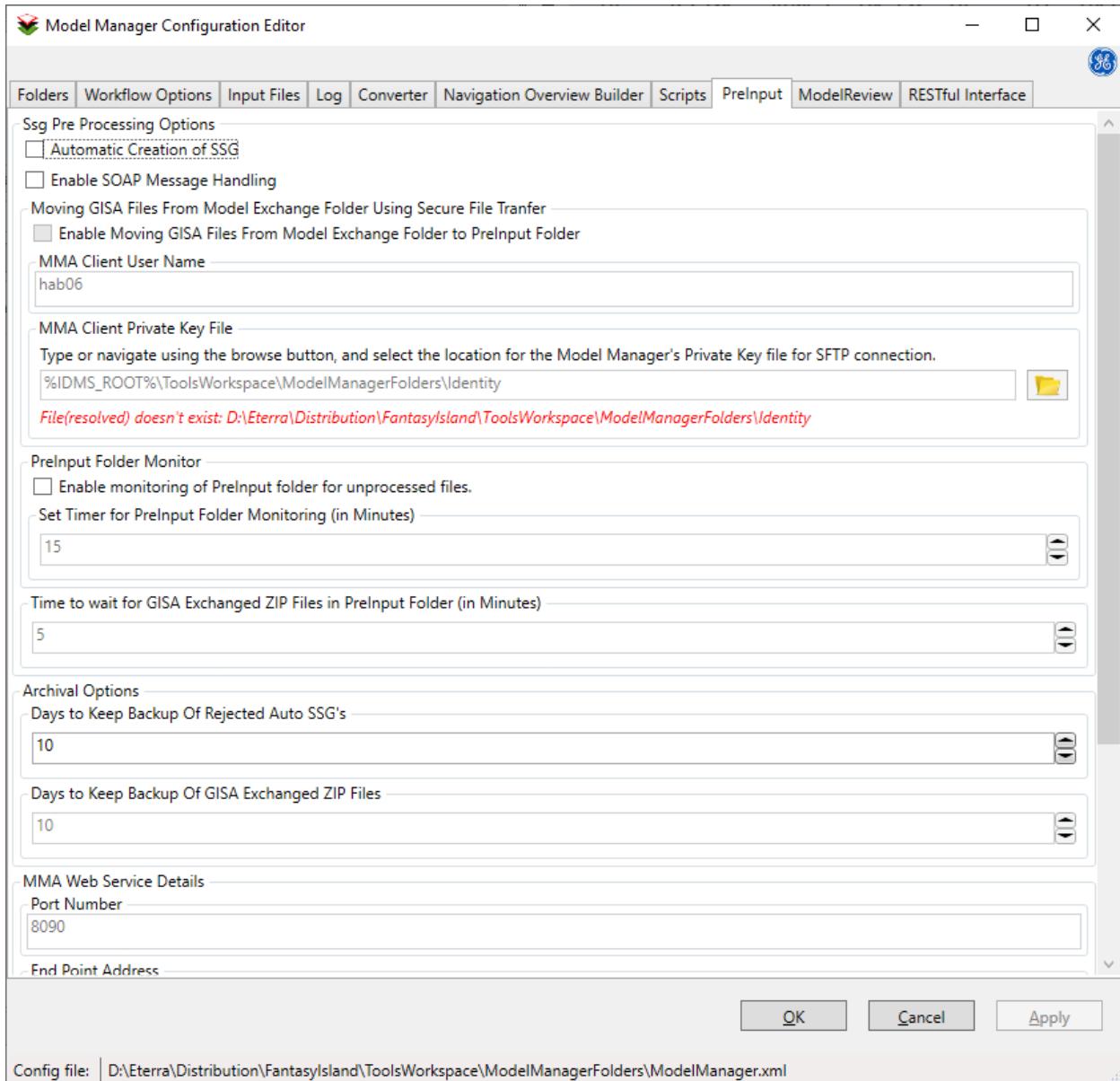


Figure 28. Model Manager Configuration Editor PreInput Tab

4.8.1 SSG Pre-Processing Options

The screenshot shows the 'Ssg Pre Processing Options' configuration page. It includes sections for 'Automatic Creation of SSG', 'Enable SOAP Message Handling', 'Moving GISA Files From Model Exchange Folder Using Secure File Transfer', 'MMA Client User Name' (set to 'hab06'), 'MMA Client Private Key File' (with a browse button and path '%IDMS_ROOT%\Tools\Workspace\ModelManagerFolders\Identity'), and a note about the file path being resolved.

Ssg Pre Processing Options

Automatic Creation of SSG

Enable SOAP Message Handling

Moving GISA Files From Model Exchange Folder Using Secure File Transfer

Enable Moving GISA Files From Model Exchange Folder to PreInput Folder

MMA Client User Name
hab06

MMA Client Private Key File

Type or navigate using the browse button, and select the location for the Model Manager's Private Key file for SFTP connection.
%IDMS_ROOT%\Tools\Workspace\ModelManagerFolders\Identity

File(resolved) doesn't exist: D:\Eterra\Distribution\Fantasy\Island\Tools\Workspace\ModelManagerFolders\Identity

This section describes SSG file pre-processing options.

- **Automatic Creation of SSG:** Automatically creates an SSG file when a full or incremental substation externals or internals file appears in the PreInput folder. The SSG file is created according to the schema provided in section 2.8.2 SsgFile XSD Schema for SSG File Validation. For more information, see section 5.14 Automatically Creating SSG Files.
- **Enable SOAP Message Handling:** Enables the Model Manager web service, which is used to receive substation externals and internals file updates from the GIS Adapter. This option requires that the Automatic Creation of SSG option is enabled.
- **Enable Moving GISA Files from Model Exchange Folder to PreInput Folder Using Secure File Transfer:** Enables moving GISA files from the Model Exchange folder to the PreInput folder using SFTP.
 - **MMA Client User Name:** The user name that is used for initiating SFTP connections.
 - **MMA Client Private Key File for SFTP:** The path to the Model Manager's private key file for SFTP.

For more details about setting up an SFTP connection for the Model Manager and defining the Model Exchange folder, refer to the *Model Manager Software Installation and Maintenance Guide*.

4.8.2 PreInput Folder Monitoring

The screenshot shows the 'PreInput Folder Monitor' configuration page. It includes sections for 'Enable monitoring of PreInput folder for unprocessed files' (checked), 'Set Timer for PreInput Folder Monitoring (in Minutes)' (set to 15), and 'Time to wait for GISA Exchanged ZIP Files in PreInput Folder (in Minutes)' (set to 5).

PreInput Folder Monitor

Enable monitoring of PreInput folder for unprocessed files.

Set Timer for PreInput Folder Monitoring (in Minutes)
15

Time to wait for GISA Exchanged ZIP Files in PreInput Folder (in Minutes)
5

This section describes options for monitoring the PreInput folder.

- **Enable Monitoring of PreInput Folder for Unprocessed Files:** Select this option to enable monitoring of the model manager PreInput folder for files that are unprocessed, such as station XML files. This option is similar to the Input Folder Monitor option on the Workflow Options tab. For more information, see section 5.11 Monitoring the Input Folder for Unprocessed Files.

- **Set Timer for PreInput Folder Monitoring (in Minutes):** Specifies the time period (in minutes) for the PreInput folder to be monitored. The minimum value is 5 minutes and the maximum value is 120 minutes.
- **Time to wait for GISA Exchanged ZIP Files in PreInput Folder (in Minutes):** Specifies the time period (in minutes) to wait for more zip files before starting the processing. The minimum value is 5 minutes and the maximum value is 15 minutes.

4.8.3 Archiving

Archival Options

Days to Keep Backup Of Rejected Auto SSG's	<input type="text" value="10"/>
Days to Keep Backup Of GISA Exchanged ZIP Files	<input type="text" value="10"/>

This section describes the PreInput folder archiving options.

- **Days to Keep Backup of Rejected Auto SSG's:** Specifies the number of days for SSGs that were automatically created by the Model Manager and rejected due to conflicting incremental and full files to be kept in the Backup\RejectedAutoSSGS folder.
- **Days to Keep Backup of GISA Exchanged ZIP Files:** Specifies the number of days for zip files from the GIS Adapter to be kept in the Backup\GISAEExchangedZipFiles folder.

4.8.4 MMA Web Service Details

MMA Web Service Details

Port Number	<input type="text" value="8090"/>
End Point Address	<input type="text" value="http://localhost:8090/MmaWebService.svc"/>

This section describes the Model Manager web service options.

- **MMA Web Service Port Number:** The Model Manager web service port.
- **MMA Web Service End Point Address:** The Model Manager web service endpoint URL.

4.8.5 MMA Client Details

MMA Client Details

Certificate File	Type or navigate using the browse button, and select the location of certificate file used for Client Authentication. <input type="text" value="%IDMS_ROOT%\MMASoapInterface.cer"/> <input type="button" value="Browse"/>				
GISA Instance Details	<table border="1"> <tr> <td>GISA Instance Name</td> <td>GISA Service End Point Address</td> </tr> <tr> <td colspan="2">Select GISA instance to remove using the check box to the left of each item. <input type="button" value="Add"/> <input type="button" value="Remove"/></td> </tr> </table>	GISA Instance Name	GISA Service End Point Address	Select GISA instance to remove using the check box to the left of each item. <input type="button" value="Add"/> <input type="button" value="Remove"/>	
GISA Instance Name	GISA Service End Point Address				
Select GISA instance to remove using the check box to the left of each item. <input type="button" value="Add"/> <input type="button" value="Remove"/>					

This section describes the Model Manager web service client options.

- **Certificate File:** The path to the certificate file for client authentication.
- **GISA Instance Details:**
 - **GISA Instance Name:** The GIS Adapter instance name.
 - **GISA Service End Point Address:** The GIS Adapter web service endpoint URL.

4.9 Model Review Tab

The Model Review tab allows you to configure the Automated Model Review functionality of the Model Manager.

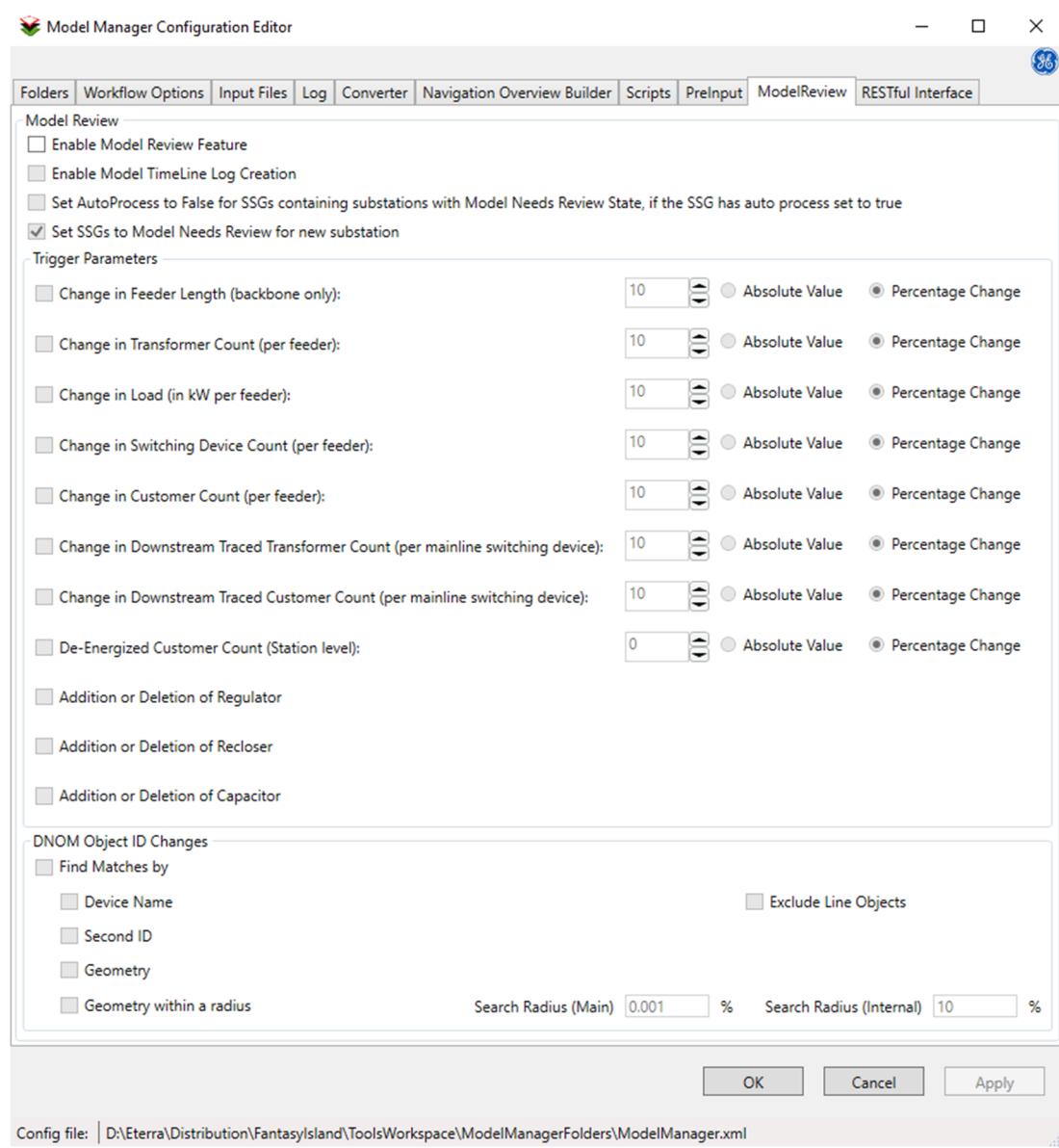


Figure 29. Model Manager Configuration Editor Model Review Tab

The Model Review tab includes the following fields and controls:

- **Enable Model Review:** If this option is selected, the Automated Model Review functionality is enabled.

The generated substation model is compared against the last generated model. If a change in any of the trigger parameters is more than the configured threshold, the model state is set to the ModelNeedsReview (Manual Model Review) state. The ModelNeedsReview state indicates that the (I)SSG(s) and/or the generated model needs to be manually reviewed. Details about the affected trigger(s) can be found on the Log Tab.

Note: If the Model Review option is enabled, the <Station>_StationDetails.dat file with information required for the automated model review (for example, the backbone feeder length and distribution transformer count) is created in the Review folder when converting a model, and then copied to the Previous folder. This file must exist in the Previous folder to enable model review. Otherwise, the review is skipped.

- **Enable Model Timeline Log Creation:** If this option is selected, the model conversion, automated review, and upload summary information (including timestamps, file names, total feeder length, and device counts) is recorded in the <Station>_ModelTimeLine.csv file. The <Station>_ModelTimeLine.csv file is located in the “Model_TimeLine” folder under the History folder for every successful model upload per substation.
- **Set AutoProcess to False for SSGs containing substations with Model Needs Review state, if the SSG has AutoProcess set to True:** If this option is selected, the AutoProcess flag is set to False for (I)SSGs with stations where changes to the Model Review trigger parameters exceed the configured threshold values. The station model files are not automatically uploaded, and the final status is set to ModelNeedsReview so that the models can pass validation procedures.
- **Set SSGs to Model Needs Review for New Substation:** If this option is selected, SSG files for new substations are automatically set to the ModelNeedsReview state after the model is converted (with or without converter fatal errors), even if the AutoProcess flag is set to true.
- **Trigger Parameters:** These parameters configure the value change thresholds for triggering the ModelNeedsReview state. Each threshold value can be specified in absolute or percentage units.
 - **Change in Feeder Length (backbone only):** The ModelNeedsReview state is triggered if a change in the backbone feeder length is greater than or equal to the configured value. If this value is set to 0, any changes to the feeder length are ignored.
 - **Change in Transformer Count (per feeder):** The ModelNeedsReview state is triggered if a change in the distribution transformer count (per feeder) is greater than or equal to the configured value. If this value is set to 0, any changes to the distribution transformer count are ignored.
 - **Change in Load (in kW per feeder):** The ModelNeedsReview state is triggered if a change in the feeder load (per feeder, in kW) is greater than or equal to the configured value. If this value is set to 0, any changes to the feeder load are ignored.
 - **Change in Switching Device Count (per feeder):** The ModelNeedsReview state is triggered if a change in the switching device count (per feeder) is greater than or equal to the configured value. If this value is set to 0, any changes to the switching device count are ignored.

- **Change in Customer Count (per feeder):** The ModelNeedsReview state is triggered if a change in the customer count (per feeder) is greater than or equal to the configured value. If this value is set to 0, any changes to the customer count are ignored.
 - **Change in Downstream Traced Transformer Count (per mainline switching device):** The ModelNeedsReview state is triggered if a change in the downstream traced transformer count (per backbone switching device) is greater than or equal to the configured value. If this value is set to 0, any changes to the downstream traced transformer count are ignored.
 - **Change in Downstream Traced Customer Count (per mainline switching device):** The ModelNeedsReview state is triggered if a change in the downstream traced customer count (per backbone switching device) is greater than or equal to the configured value. If this value is set to 0, any changes to the downstream traced transformer count are ignored.
 - **De-energized Customer Count (Station level):** The ModelNeedsReview state is triggered if the de-energized customer count is greater than the configured value.
 - **Change in Addition or Deletion of Regulator:** The ModelNeedsReview state is triggered if a regulator is added or removed from the station model. (The system compares the devices' ID and Name properties, not the number of devices.)
 - **Change in Addition or Deletion of Recloser:** The ModelNeedsReview state is triggered if a recloser is added or removed from the station model. (The system compares the devices' ID and Name properties, not the number of devices.)
 - **Change in Addition or Deletion of Capacitor:** The ModelNeedsReview state is triggered if a capacitor is added or removed from the station model. (The system compares the devices' ID and Name properties, not the number of devices.)
- **DNOM Object ID Changed Review:** These parameters are used to identify DNOM Object ID changes that occurred in the new model compared to the existing model. If the Model Manager finds changes in Object ID, the tool triggers the ModelNeedsReview state for the processing model. The Trigger Type is set to "Change in Device Object ID" in the Model Review Log Tab.
 - **Find All Matches By:** This option list configures the type of fields that are matched to identify Object ID changes when an unrecognized Object ID is found in the processed model.
 - **Device Name:** If selected, the existing model is searched for the Device Name value of an entry with an unrecognized Object ID. If a match is found, the corresponding information is displayed in the Model Review Log Tab.
 - **SecondId:** If selected, the existing model is searched for the SecondId value of an entry with an unrecognized Object ID. If a match is found, the corresponding information is displayed in the Model Review Log Tab.
 - **Geometry:** If selected, the existing model is searched for the geometry location value of an entry with an unrecognized Object ID. If a match is found, the corresponding information is displayed in the Model Review Log Tab.
 - **Geometry Within a Radius:** If selected, the existing model is searched for the geometry location value of an entry with an unrecognized Object ID. If a match is found, the corresponding information is displayed in the Model Review Log Tab.
 - Search Radius (Main):** Defines the radius range of possible changes in geometry location for station externals (network view) in the existing model on the search.
 - Search Radius (Internal):** Defines the radius range of possible changes in geometry location for station internals in the existing model on the search.

- **Exclude Line Objects:** If selected, Line objects are not considered when identifying Object ID changes.

4.10 RESTful Interface Tab

The RESTful Interface tab allows you to configure the REST web interface functionality of the Model Manager.

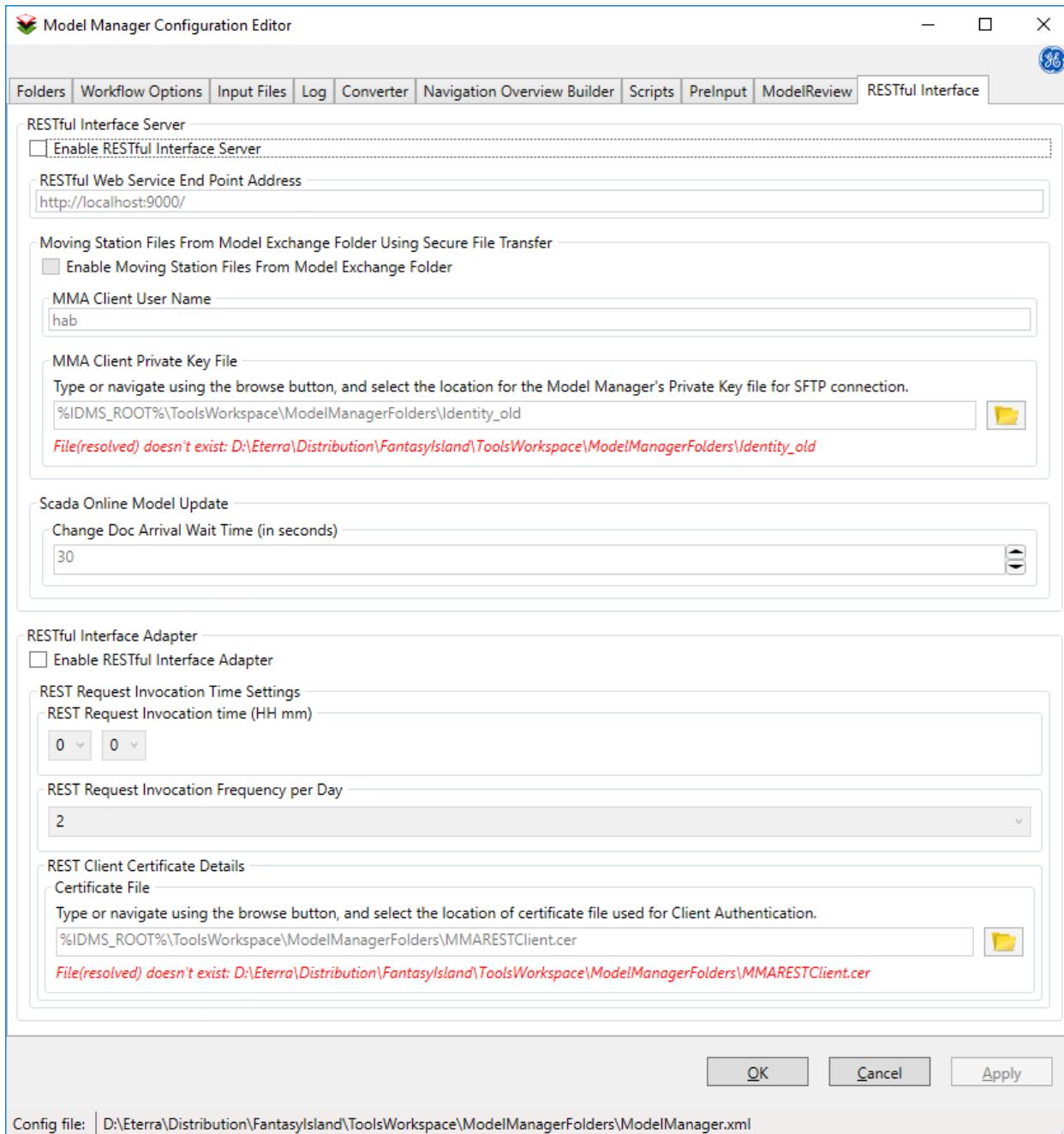


Figure 30. Model Manager Configuration Editor RESTful Interface Tab

The RESTful Interface tab includes the following fields and controls:

- **Enable RESTful Interface Server:** If this option is selected, the Model Manager REST web interface functionality is enabled.
 - **RESTful Web Service End Point Address:** The Model Manager RESTweb service endpoint URL.
- **Enable Moving Station Files from Model Exchange Folder:** Enables moving station files from the Model Exchange folder to the PreInput folder using SFTP.
 - **MMA Client User Name:** The username that is used for initiating SFTP connections.
 - **MMA Client Private Key File for SFTP:** The path to the Model Manager's private key file for SFTP.
- **Scada Online Model Update:**
 - **Change Doc Arrival Wait Time:** Time period, in seconds, for monitoring the Change Doc folder for Change Document files.
- **Enable RESTful Interface Adapter:** If this option is selected, the Model Manager REST web interface adapter is enabled.
 - **REST Request Invocation Time (HH mm):** The time span for processing REST client requests, in hours and minutes. The default value is 00:00, which means that the REST Client invokes the External System RESTful Server at 12 AM midnight.
 - **REST Request Invocation Frequency Per Day:** Number of times per day the REST Client invokes External System's RESTful Server. The default value is 1, which means that the REST Client invokes the External System RESTful Server only once per day (at 12 AM midnight).
 - **REST Client Certificate File:** The path to the certificate file that is used for REST client authentication while calling the external web service end points.

When processing REST client requests, the REST web interface adapter calls REST APIs to get the list of modified (create/update/delete) feeders, and then calls REST APIs to get the full feeder file for each modified feeder. After getting the full feeder files, the Model Manager drops full feeder files received in the input folder, updates the station-to-feeder mapping file for modified feeders, identifies stations that require conversion, and converts these stations.

5. Using the Model Manager

This chapter describes how to operate the Model Manager application.

5.1 Starting the Model Manager

To start the Model Manager Service Client:

1. Start the Model Manager Service from the Windows Start menu using the following path:
Start > Programs > Eterra > Distribution > Tools > Modeling Tools > Start Model Manager Service.
2. Start the Model Manager Service Client from the Windows Start menu using the following path:
Start > Programs > Eterra > Distribution > Tools > Modeling Tools > Model Manager Service Client

The Model Manager Service Client shortcut passes the Model Manager Service connection string to the application, as specified in the Target field of the shortcut's properties dialog box. The default Target parameter is:

```
D:\Eterra\Distribution\Tools\bin\ModelManagerService.Client.exe /SERVICE  
"net.tcp://localhost:8085/ModelManagerService"
```

You can review and modify the startup parameters as necessary.

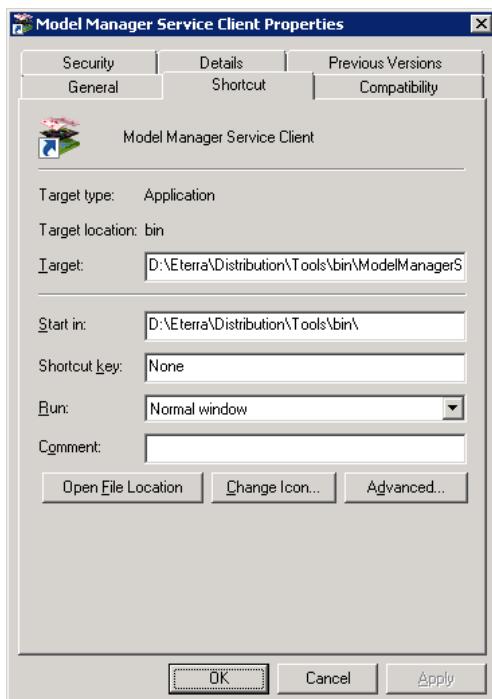
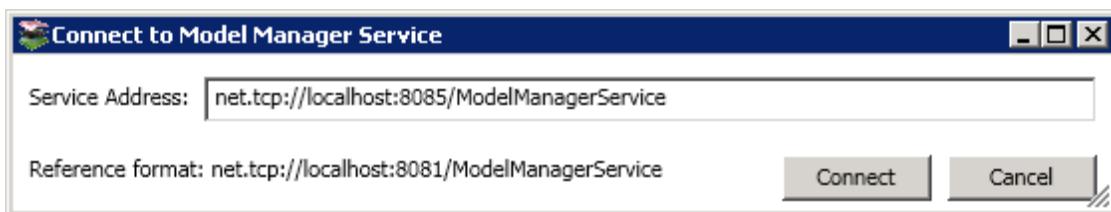


Figure 31. Model Manager Service Client Properties

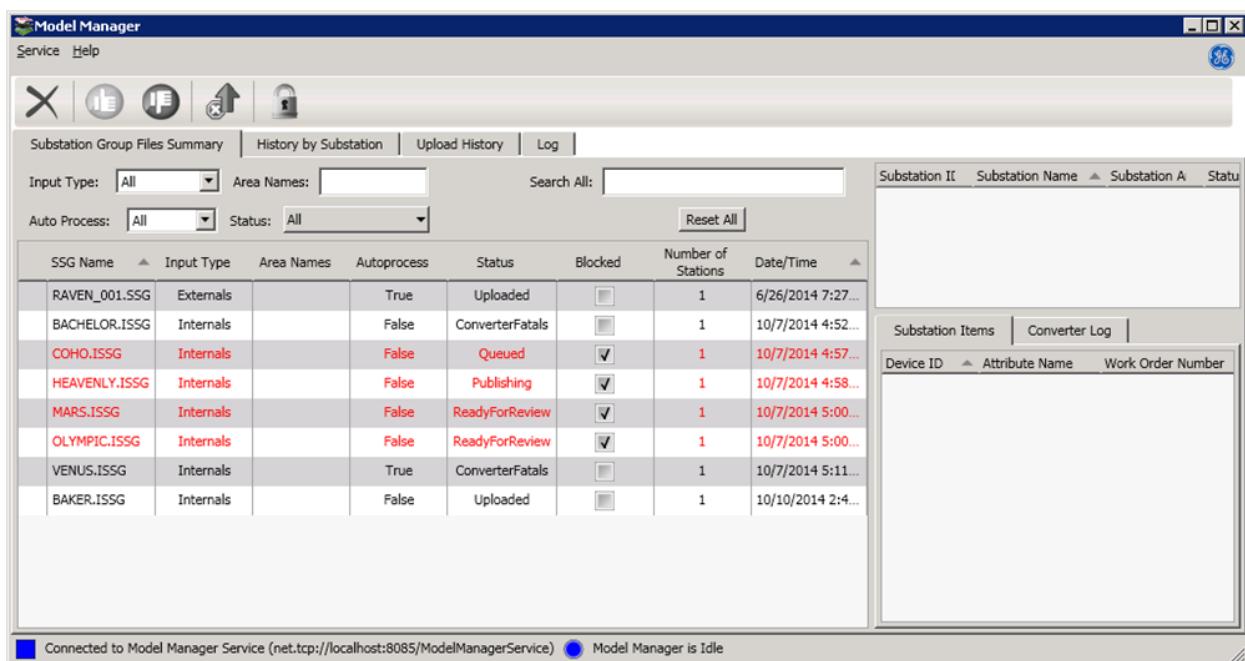
If you started the Service Client before the Service, the client's status bar displays "Cannot connect to Model Manager Service".

To connect to the Model Manager Service:

1. Navigate to the Service > Connect menu option and select the Connect button in the Connect to the Model Manager Service dialog box.



2. The status bar shows "Connected to Model Manager Service".



To start the Model Manager Application:

Start the Model Manager application from the Windows Start menu using the following path:

Start > Programs > Eterra > Distribution > Tools > Modeling Tools > Model Manager Application.

The Model Manager Application shortcut passes a configuration file to the application as specified in the Target field of the shortcut's properties dialog box:

```
D:\Eterra\Distribution\Tools\bin\ModelManagerApplication.exe /CONFIG  
"%IDMS_ROOT%\ToolsWorkspace\ModelManagerFolders\ModelManager.xml"
```

You can review and modify the startup parameters as necessary.

The configuration file specifies (among other things) the directory structure and the location of the Converter and Navigation Overview Builder tool by default. You can change the default configuration by following the instructions in chapter 4 Configuring Model Manager.

5.2 Stopping the Model Manager

To stop the Model Manager Service Client:

1. Close the Service Client.
2. Navigate to Start > Programs > Eterra > Distribution > Tools > Modeling Tools and select “Stop Model Manager Service”.

A command prompt appears showing that the service is stopping. Once the Model Manager Service is stopped, the command window closes automatically.

To close the Model Manager Application:

- Select Exit from the File menu.

5.3 Automatically Processing Substation Files

Automatic processing occurs when the (I)SSG file’s AutoProcess flag is set to True.

To automatically process a single station file:

1. Open Microsoft Windows Explorer and browse to the Model Manager Input folder.

The path to the default location of the Model Manager input directory is
%IDMS_ROOT%\ToolsWorkspace\ModelManagerFolders\Input

2. Place all the required files for a substation conversion into the Input folder.

A possible list of required files includes a substation (I)SSG file, an external GIS extract (<substation name>.xml/<substation name>_sub.csv), a Substation Editor internals file (<substation name>_internals.xml), a Substation Editor station externals file (<substation name>.xml), a Substation Editor placement file (<substation name>_placements.xml), a global.csv file, a global_internals.xml file, a compiled binary customer list file *.bpcl, and power flow files for the Converter.

3. Start the Model Manager application, as described in section 5.1 Starting the Model Manager.

Monitor the substation group files summary list:

- The (I)SSG file appears on the list.
- The AutoProcess flag value states True.
- The status moves from Queued to Converting to Uploaded. Additional statuses may be visible.

Substation Group Files Summary		History by Substation		Upload History		Log			
Input Type: All		Area Names:		Search All:					
Auto Process: All		Status: All		Reset All					
SSG Name	Input Type	Area Names	Autoprocess	Status	Blocked	Number of Stations	Date/Time		
RAVEN_001.SSG	Externals		True	Uploaded	<input type="checkbox"/>	1	6/26/2014 7...		
VENUS.ISSG	Internals		True	ConverterFatal	<input type="checkbox"/>	1	10/7/2014 5...		
BACHELOR.ISSG	Internals		False	ConverterFatal	<input type="checkbox"/>	1	10/7/2014 4...		
COHO.ISSG	Internals		False	Queued	<input checked="" type="checkbox"/>	1	10/7/2014 4...		
HEAVENLY.ISSG	Internals		False	Publishing	<input checked="" type="checkbox"/>	1	10/7/2014 4...		
MARS.ISSG	Internals		False	ReadyForRev...	<input checked="" type="checkbox"/>	1	10/7/2014 5...		
OLYMPIC.ISSG	Internals		False	ReadyForRev...	<input checked="" type="checkbox"/>	1	10/7/2014 5...		
BAKER.ISSG	Internals		False	Uploaded	<input type="checkbox"/>	1	10/10/2014...		

Once the status is Uploaded:

- The Uploaded message is logged for the (I)SSG file.
- The .mod file is copied with a current timestamp for the substation that is processed in the (I)SSG file in the Review folder.
- The .mod file is stored with a current timestamp for the substation processed in the (I)SSG file in the Upload folder.
- The backup of the input files is created in the Backup folder.

5.4 Reviewing Substation Files

Reviewing is required when the (I)SSG file's AutoProcess flag is set to False. Also, when the AutoProcess flag is set to True, a copy of the output files is available in the Review folder.

To process a single station file manually:

1. Open Windows Explorer and browse to the Model Manager Input folder.

The path to the default location of the Model Manager directory is
%IDMS_ROOT%\ToolsWorkspace\ModelManagerFolders\Input

2. Place all the required files for a substation conversion into the Input folder.

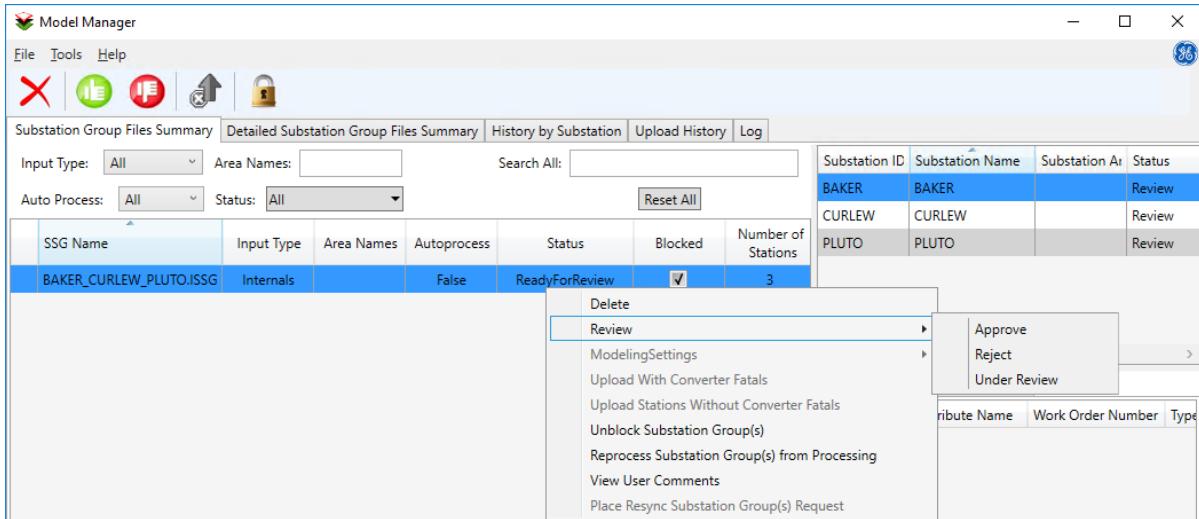
A possible list of required files includes a substation (I)SSG file, an external GIS extract (<substation name>.xml/<substation name>_sub.csv), a Substation Editor internals file (<substation name>_internals.xml), a Substation Editor station externals file (<substation name>.xml), a Substation Editor placement file (<substation name>_placements.xml), a global.csv file, a global_internals.xml file, a compiled binary customer list file *.bpcl, and power flow files for the Converter.

3. Start the Model Manager application, as described in section 5.1 Starting the Model Manager.

Monitor the substation group files summary list:

- The (I)SSG file appears on the list.
- The status moves from Queued, to Converting, to Ready for Review (or Model Needs Review if the Model Review feature is enabled and the review state is triggered based on the configuration). Additional statuses may be visible.

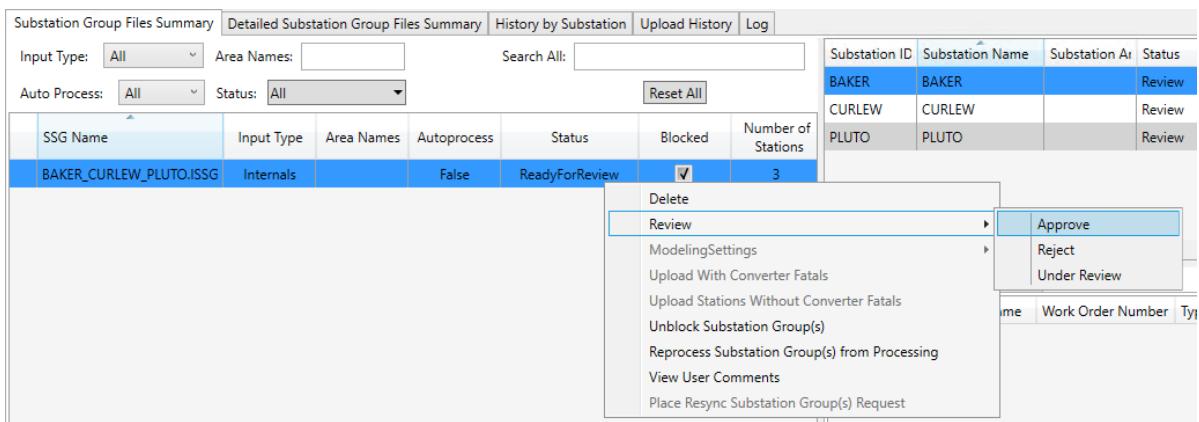
- If the AutoProcess flag value is False, and the file passes validation procedures, the final status is Ready for Review (or Model Needs Review).
- Open the History by Substation tab.
- Review the information, as described in section 3.1.4 History by Substation Tab.
- On the Substation Group Files Summary tab, right-click the (I)SSG file, point to the Review item, and select the Review option.



5.4.1 Approving Substation Files

To approve a substation file:

- Open the Substation Group Files Summary tab and right-click the (I)SSG file that is Ready for Review (or Model Needs Review).
The right-click menu appears with the enabled Review option.
- Point to the Review item, and then select the Approve option to approve the file conversion.



A confirmation message appears.

- Click Yes.
The status of the file changes to Publishing and then to Uploaded.

The Uploaded message is logged for the (I)SSG file.

The .mod file is stored with a current timestamp for the substation processed in the (I)SSG file in the Upload folder.

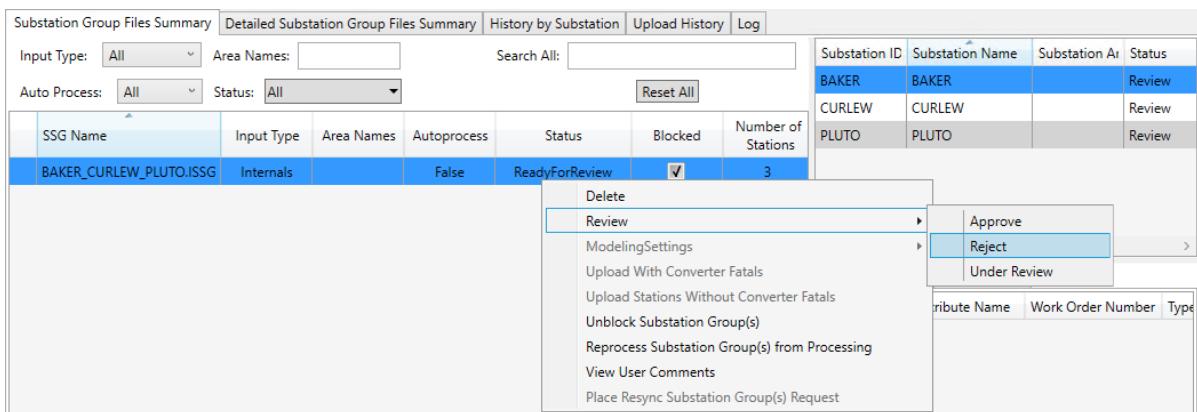
5.4.2 Rejecting Substation Files

To reject a substation file:

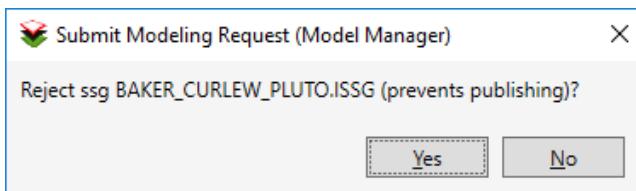
1. Open the Substation Group Files Summary tab and right-click the (I)SSG file that is Ready for Review (or Model Needs Review).

The right-click menu appears with the enabled Review option.

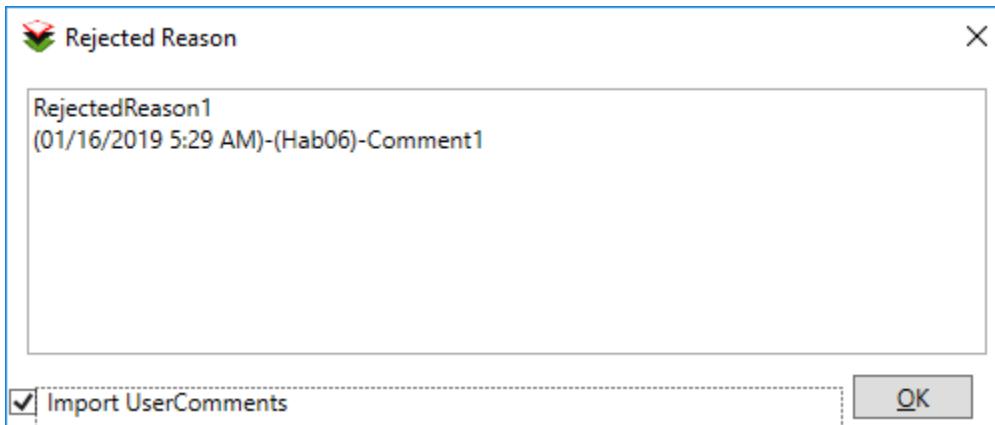
2. Point to the Review option, and then click Reject to reject the file conversion.



A confirmation message appears.



3. Click Yes and enter the rejection reason in the new prompt. Optionally, select the Import User Comments option to show previous review comments for this (I)SSG file.



The status of the (I)SSG file changes to Rejected.

The Rejected message is logged for the (I)SSG file. You can later reprocess this file with new inputs, as appropriate.

5.4.3 Setting Status to Under Review

To set a substation file status to Under Review:

1. Open the Substation Group Files Summary tab and right-click the (I)SSG file that is Ready for Review (or Model Needs Review).

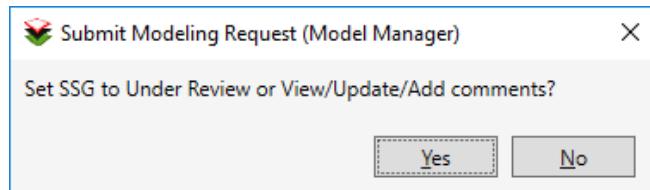
The right-click menu appears with the enabled Review option.

2. Point to the Review item and select the Under Review option to indicate that the (I)SSG is being reviewed.

The screenshot shows the 'Substation Group Files Summary' tab. A context menu is open over a row for 'BAKER_CURLEW_PLUTO.ISSG'. The menu path 'Review > Under Review' is highlighted. The main table displays two rows: 'BAKER_CURLEW_PLUTO.ISSG' (Status: Rejected) and 'PLUTO.ISSG' (Status: ReadyForReview). The right side of the screen shows a detailed view of the 'PLUTO' substation.

SSG Name	Input Type	Area Names	Autoprocess	Status	Blocked	Number of Stations
BAKER_CURLEW_PLUTO.ISSG	Internals		False	Rejected	<input checked="" type="checkbox"/>	3
PLUTO.ISSG	Internals		False	ReadyForReview	<input checked="" type="checkbox"/>	1

A confirmation message appears.



3. Click Yes.

The (I)SSG status is changed to Under Review.

If the Allow Popup for Viewing and Entering User Comments for SSG Files under Review option on the Workflow Options Tab is selected, the SSG Under Review dialog box opens. You can enter a new comment and view the previously entered comments.

 SSG Under Review

X

Comment History

Date and Time	Username	SSG Name	Comments
1/16/2019 5:48:51 AM	Hab06	PLUTO.ISSG	Comment1

Enter Comments:

Comment2

Comment details (SSG name, username, comment, and timestamp) are written to the SSG history file for each comment. Also, the appropriate log entries are added on the Log tab.

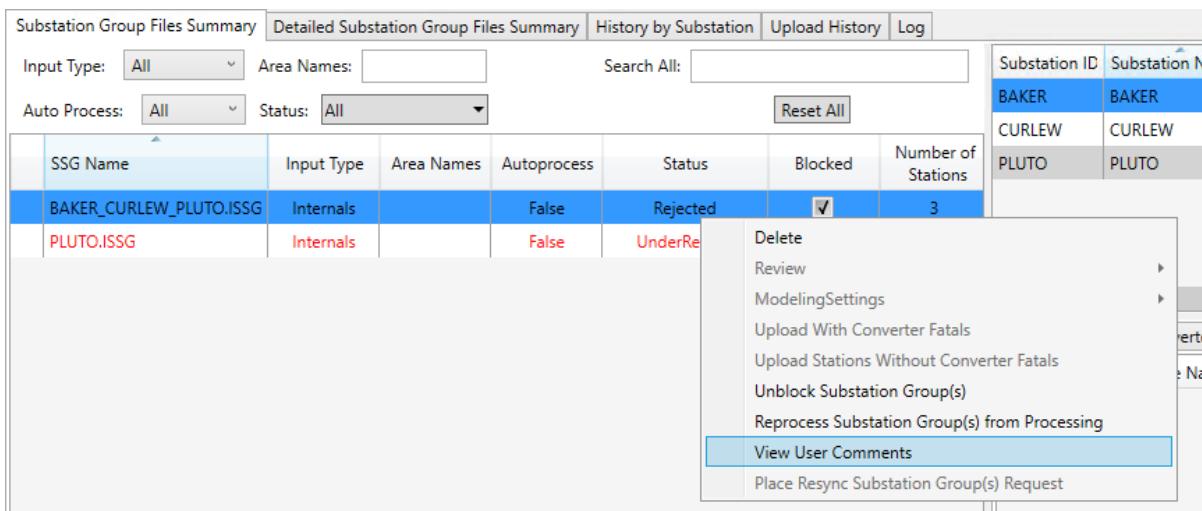
For an (I)SSG file under review, you can add more comments, approve, or reject the file, as appropriate.

Note: If the "Check Review Completed for All Stations before Allowing SSG to be Approved" is enabled on the Workflow Options Tab, you must set the Review Completed flag on the Substation Status pane of the Detailed Substation Group Files Summary tab to complete the review and enable approving or rejecting (I)SSG files.

5.4.4 Reviewing the Substation File Comments

To view the substation file comments:

1. Open the Substation Group Files Summary tab and right-click an (I)SSG file.
The right-click menu appears.
2. Click the View User Comments item.



The SSG User Comments dialog box opens, showing the previously entered comments.

The dialog box has a title bar 'SSG User Comments' and a close button 'X'. It contains a table titled 'Comment History' with columns: Date and Time, Username, SSG Name, and Comments. There are two entries: one from 'Hab06' at 1/16/2019 5:42:34 AM with comment 'RejectedReason1', and another from 'Hab06' at 1/16/2019 5:29:53 AM with comment 'Comment1'. Below the table is a section 'Enter Comments:' with a large text input area. At the bottom are three buttons: 'Add Comment', 'Ok', and 'Cancel'.

Date and Time	Username	SSG Name	Comments
1/16/2019 5:42:34 AM	Hab06	BAKER_CURLEW_PLUTO.ISSG	RejectedReason1
1/16/2019 5:29:53 AM	Hab06	BAKER_CURLEW_PLUTO.ISSG	Comment1

5.5 Unblocking Stations

To remove a blocked station from the Blocked Stations List:

1. Review the (I)SSG file that the station belongs to, as described in section 5.4 Reviewing Substation Files.
2. Navigate to the History by Substation tab.
3. Right-click a blocked station and select Unblock Substation Group(s) on the context menu.

SSG Name	Input Type	Status	Blocked	Area Names	Autoprocess	Number of Stations	Date/Time
BAKER_001.SSG	Externals	Uploaded	<input type="checkbox"/>	NORTH	True	1	11/29/201
DURBAN_002.ISSG	Internals	Rejected	<input checked="" type="checkbox"/>		False	1	7/15/2015
CURLEW_002.ISSG	Internals	ReadyForRevi...	<input checked="" type="checkbox"/>		False	1	7/15/2015
MULTIPLE_001.SSG	Externals	Queued	<input checked="" type="checkbox"/>		False	2	7/15/2015
3LIONS_001.SSG	Externals	Publishing	<input checked="" type="checkbox"/>		False	2	7/29/201
EQSI.ISSG	Internals	Publishing	<input checked="" type="checkbox"/>		False	2	5/20/2015
RAVEN_001.SSG	Externals	ConverterFatal	<input type="checkbox"/>		False	1	7/15/201
PLUTO.ISSG	Internals	ConverterFatal	<input type="checkbox"/>		False	1	4/2014

Or uncheck the Blocked check box.

Or click the Unblock toolbar button

The station is now removed from the Blocked Stations List.

5.6 Processing Files with Converter Fatal Errors

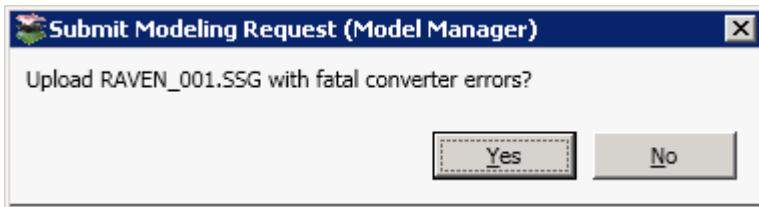
In some situations, you may need to continue processing (I)SSG files even if fatal converter errors occur during processing and the Model Manager moves the files into the Review folder.

To continue processing (I)SSG files:

1. Navigate to the Substation Group Files Summary tab.
2. Right-click the (I)SSG file with ConverterFatal status and select the “Upload with Converter Fatal” option.

SSG Name	Input Type	Status	Blocked	Area Names	Autoprocess	Number of Stations
RAVEN_001.SSG	Externals	ConverterFatal	<input type="checkbox"/>		True	1
MULTIPLE_001.SSG	Externals	Queued			False	3
BAKER_001.SSG	Externals	Uploaded			False	1
3LIONS_001.SSG	Externals	Publishing			False	1
CURLEW_002.ISSG	Internals	ReadyForRevi...			False	1
DURBAN_002.ISSG	Internals	Rejected			False	1

A confirmation message appears.



3. Click Yes. If the Model Review feature is disabled or the Model Needs Review state is not triggered based on the configuration, the file is moved to the Upload folder (the file's status changes to Publishing and then to Uploaded). If the Model Review feature is enabled and a review is triggered, the file status changes to "Model Needs Review." The review state is processed as described in section 5.4 Reviewing Substation Files.

5.7 Partial Model Upload for ISSG Files

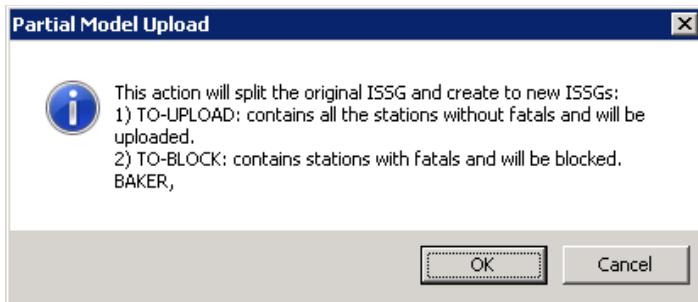
ISSG files may include both stations with and without fatal converter errors. For this situation the Model Manager tool includes an option to upload only stations without fatsals and block stations with fatsals.

To upload stations without fatsals:

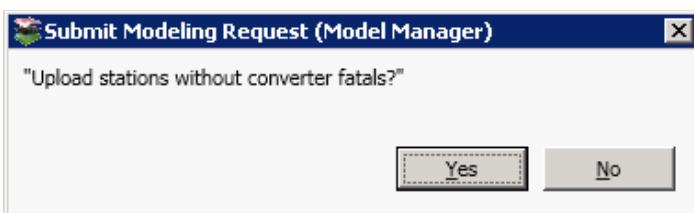
1. Open the Model Manager.
 2. Click Tools > Configuration.
 - Model Manager Configuration Editor opens.
 3. On the Workflow Options Tab select the Partial Model Upload for ISSG option.
 4. Click OK to save changes.
 5. Copy an ISSG file with station internal files to the Input folder.
 6. Open the Substation Group Files Summary tab and wait until the ISSG file is processed.
- If the ISSG file contains stations with fatal converter errors, the ConverterFatsals status is displayed.
7. Right-click the ISSG file record and select the Upload Stations Without Converter Fatsals option.

Substation II	Substation Name	Substation A	Status	Description
BAKER	BAKER		ConverterFa...	
CURLEW	CURLEW		Converted	
PLUTO	PLUTO		Converted	

A confirmation message appears.



8. Click OK to create two new ISSG files using the existing one:
 - **TO-UPLOAD:** This ISSG file will include stations without converter fatalities.
 - **TO-BLOCK:** This ISSG file will include stations with converter fatalities.
- After these two files are created, the Model Manager processes them and then uploads stations without fatalities and blocks stations with fatalities. Before the upload, a confirmation message appears.



9. Click Yes. If the Model Review feature is disabled or the Model Needs Review state is not triggered based on the configuration, the stations without converter fatalities are moved to the Upload folder (the file's status changes to Publishing and then to Uploaded). If the Model Review feature is enabled and a review is triggered, the file status changes to "Model Needs Review." The review state is processed as described in section 5.4 Reviewing Substation Files.

The new ISSG file records are updated accordingly.

This screenshot shows the "Substation Group Files Summary" tab. It displays a list of substation records with columns for SSG Name, Input Type, Area Names, Autoprocess, Status, Blocked, Number of Stations, and Date/Time. A secondary table on the right shows Substation ID, Substation Name, Substation A, Status, and Description. The "Status" column in the main table includes dropdown menus for "Input Type" and "Status". The "Blocked" column has checkboxes. The "Number of Stations" and "Date/Time" columns are also present.

SSG Name	Input Type	Area Names	Autoprocess	Status	Blocked	Number of Stations	Date/Time
BAKER_CURLEW_P...	Internals		True	ConverterFa...	<input checked="" type="checkbox"/>	3	3/16/2015 4...
BAKER_CURLEW_P...	Internals		True	ConverterFa...	<input checked="" type="checkbox"/>	1	3/16/2015 7...
BAKER_CURLEW_P...	Internals		True	Uploaded	<input type="checkbox"/>	2	3/16/2015 7...

Substation ID	Substation Name	Substation A	Status	Description
BAKER	BAKER		Converted	
CURLEW	CURLEW		Converted	
PLUTO	PLUTO		Converted	

5.8 Deleting Records

To delete a history record from the Model Manager Summary tab:

1. Open the Substation Group Files Summary tab, and right-click the record you need to delete. The right-click menu includes the enabled Delete option.
 2. Click the Delete option.
- A confirmation message appears.

The screenshot shows a table titled "Substation Group Files Summary" with columns: SSG Name, Input Type, Area Names, Autoprocess, Status, Blocked, Number of Stations, and Date/Time. A modal dialog box titled "Submit Modeling Request (Model Manager)" is overlaid on the table, asking "Delete ssg record MARS.ISSG permanently?". The "Yes" button is highlighted.

SSG Name	Input Type	Area Names	Autoprocess	Status	Blocked	Number of Stations	Date/Time
RAVEN_001.SSG	Externals		True	Uploaded	<input type="checkbox"/>	1	6/26/2014 7...
BACHELOR.ISSG	Internals		False	Uploaded	<input checked="" type="checkbox"/>	1	10/7/2014 4...
COHO.ISSG	Internals		False	Uploaded	<input checked="" type="checkbox"/>	1	10/7/2014 4...
HEAVENLY.ISSG	Internals		False	Publishing	<input checked="" type="checkbox"/>	1	10/7/2014 4...
MARS.ISSG	Internals		False	Queued	<input checked="" type="checkbox"/>	1	10/7/2014 5...
OLYMPIC.ISSG	Internals						10/7/2014 5...
ROYAL TROON.ISSG	Internals						10/7/2014 5...
VENUS.ISSG	Internals						10/7/2014 5...

- Click Yes.

The selected record is deleted from the Model Manager Summary tab.

The Deleted History message is logged for the (I)SSG file.

Note: The Delete option removes the .ssgh history record file from the Model Manager history folder. It does not affect the (I)SSG file and other input files stored in the Previous folder.

5.9 Performing a Manual Rollback

To roll back and deploy online the previous version of the processed files:

- Open Windows Explorer and browse to the Model Manager Previous folder.

The path to the default location of the Model Manager directory is
%IDMS_ROOT%\Tools\Workspace\ModelManagerFolders\Previous.

- In the Previous folder, select the following files and copy them to the Upload folder:

- <substationname>.mod
- <substationname>.AOR.mod
- Overview.nvw

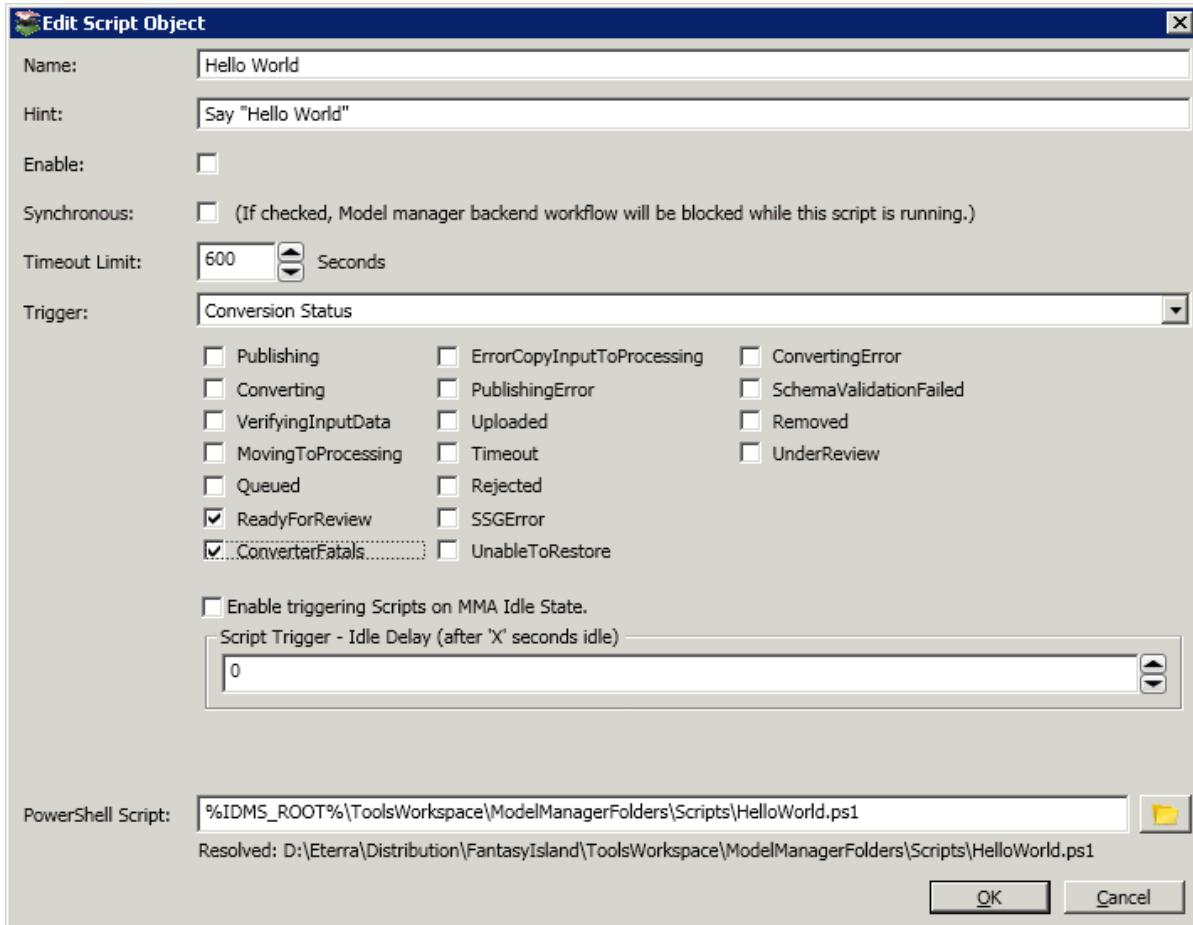
- In the Previous folder, select the following files and copy them to the Processing folder:

- A substation (I)SSG file
- An external <substation name>.xml/csv
- A Substation Editor internals file <substation name>_internals.xml
- A Substation Editor placements file <substation name>_placements.xml
- A Substation Editor station externals file <substation name>.xml
- A global.csv file
- A global_internals.xml file
- DPF Results_<substationname>.dat

5.10 Using Script Objects

Create a script object as follows:

1. Open the Model Manager Configuration Editor and navigate to the Scripts tab.
2. Click the Add button to open the New Script Object dialog box.



3. Complete the fields and select a trigger and a PowerShell script.
4. Click the OK button to accept the new script object.

Note: By default, PowerShell's execution policy is set to Restricted, which means that scripts (including scripts you write) cannot run. The Set-ExecutionPolicy command allows you to determine which Windows PowerShell scripts (if any) are allowed to run on your computer. Windows PowerShell has four execution policies:

- **Restricted:** No scripts can run. Windows PowerShell can be used only in interactive mode.
- **AllSigned:** Only scripts signed by a trusted publisher can run.
- **RemoteSigned:** Downloaded scripts must be signed by a trusted publisher before they can run.
- **Unrestricted:** No restrictions; all Windows PowerShell scripts can run.

To assign a particular policy, simply run Set-ExecutionPolicy followed by the appropriate policy name. For example, the following command sets the execution policy to AllSigned:

```
> Set-ExecutionPolicy AllSigned
```

A sample script, HelloWorld.ps1, is located in the ModelManagerFolders\Scripts folder. The script file is unsigned. To use this script, PowerShell's execution policy must be set to Unrestricted.

There are also two other sample scripts:

- **Build_Customer_File.ps1:** Performs an incremental update of a customer file (*.txt) and compiles a binary customer list file (*.bpcl).
- **Load_Shift_Demo_ModFile_Move.ps1:** Demonstrates a load shift from one feeder to another.

5.11 Monitoring the Input Folder for Unprocessed Files

The Model Manager has a mechanism to monitor the input folder for station files left unprocessed or waiting for an (I)SSG file and notify the user:

1. Open the Model Manager Configuration Editor.
2. Open the Workflow Options Tab.
3. Select the Input Folder Monitor option.
4. In the Set Timer for Input Folder Monitoring (In Minutes) field specify the timer value between 5 and 120 minutes.
5. Save your changes and restart the Model Manager.

Starting the Model Manager Application or the Model Manager Service Client triggers an internal timer. When the timer elapses, the Model Manager checks the files in the Input folder. If any unprocessed files are found, the Model Manager shows an alert message at the bottom of the display. Clicking the alert message shows the details. A new timer starts immediately after the previous timer elapses.

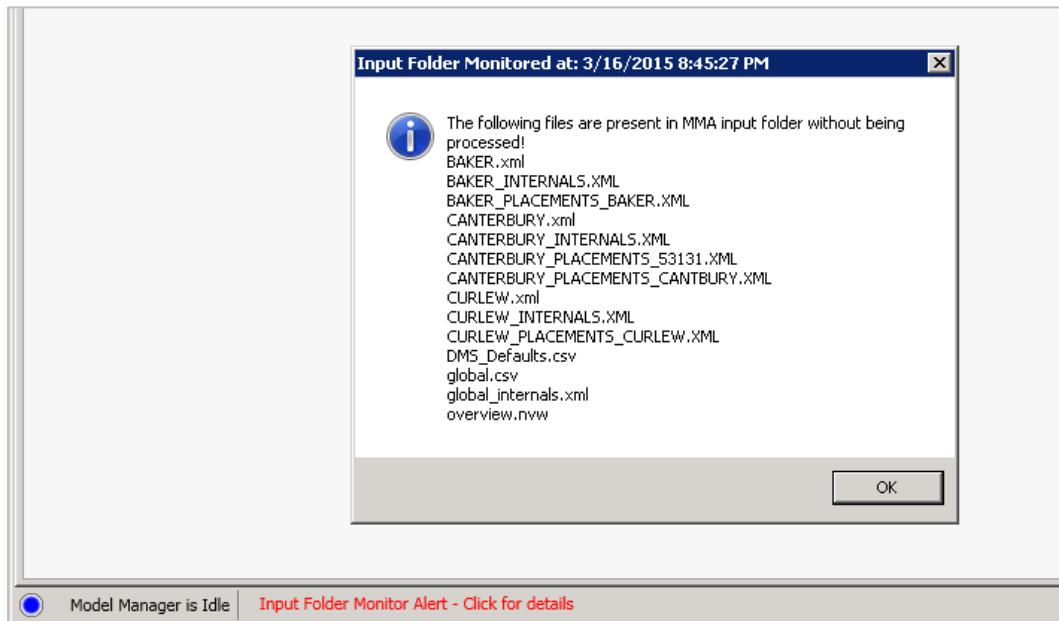


Figure 32. Input Folder Monitoring

5.12 Reprocessing Substation Files

The reprocess feature allows the model manager to restart processing of the station files to recreate the station model.

To reprocess a substation using substation files in the Processing folder:

1. Navigate to the Substation Group Files Summary tab.
2. Right-click a station and select “Reprocess Substation Group(s) from Processing” on the context menu.

This feature is useful to restart the station processing when input files have changed or missing files (for example, global.csv) have been added to the Processing folder.

Note: The "Reprocess Substation Group(s) from Processing" action forces the corresponding .ssgh file to be processed one more time, but only the selected substation contained in that .ssgh file is processed.

Note: The Reprocess Substation Group(s) from Processing option is active until the station model is uploaded. After that, the station files can be reprocessed from the backup folder.

To reprocess a substation using substation files in the Backup folder:

1. Navigate to the Upload History tab.
2. Select a substation group record.
3. Click the “Reprocess” button.

This option is available for zipped station group files contained in the Backup folder.

This feature allows the Model Manager to use a “last known good” set of substation source files from the backup folder if there are some issues with the newly created set of substation models.

5.13 Incrementally Updating Station Externals Files

To process incremental station externals file updates from the GIS Adapter:

1. Place a CIMDNOM incremental XML file in the PreInput folder of the Model Manager.
2. The Model Manager checks whether the incremental file's full station externals file (<SubstationName>.xml) appears in the Processing folder.
3. The Model Manager updates the full station externals file with changes from the incremental file.
 - o If the incremental update file and the full station externals file do not conflict, the Model Manager places both the updated file and the incremental file in the Input folder.
 - o If the incremental update file and the full station externals file conflict (for example, a specific attribute needs to be updated but the associated device does not exist in the full station externals file), the Model Manager places both the updated file and the incremental file in the Backup folder.
4. If the Automatic Creation of SSG option on the PreInput Tab of the Model Manager Configuration Editor is selected, the Model Manager creates an SSG file.
 - o If the incremental update file and the full station externals or internals file do not conflict, the Model Manager places the created SSG file in the Input folder.

- If the incremental update file and the full station externals or internals file conflict, the Model Manager places the created SSG file in the Backup/RejectedAutoSSGS folder.
5. The Model Manager processes the SSG file in the Input folder using the standard workflow.

Incremental update files must correspond to the following requirements:

- Use the file name format: <SubstationName>_<ExtractID>_incremental.xml, where ExtractID is optional and depends on the configuration.
- The content of the file follows the CIMDNOM XML format.
- The incremental file only contains the devices (DNOMObject elements) that need to be changed (updated, created, or deleted) in the associated station externals file.
- The changed devices are defined under the “Substation” root element (same as a regular CIMDNOM XML file).
- The "IncrChangeType" XML attribute (Create, Update, Delete) is added to the changed devices at the CIMDNOM level.
- The device XML element inside the incremental file has a complete set of modeled attributes.

For example, if a switch attribute is changed, the incremental file must have the full switch XML element, not only the new attribute value. In this case, the Model Manager updates the attributes of the switch with the new attributes provided in the incremental file.

Note: The ID of an existing device cannot be updated using the incremental update feature. To change an ID, the device needs to be deleted and then recreated with the new ID.

An example of a station externals incremental update file is shown below. It includes updates for a switch, removal of a feeder, and the addition of a line for a substation.

```
<?xml version="1.0" encoding="utf-8"?>
<Substation xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Id>146654391</Id>
  <Name>CANTERBURY</Name>
  <Switch IncrChangeType = "Update">
    <Id>19725999</Id>
    <Name>53030</Name>
    <PermitAreaId>CANTERSE</PermitAreaId>
    <FromNode>19726008</FromNode>
    <ToNode>19726012</ToNode>
    <ToStation/>
    <Phase>ABC</Phase>
    <Future/>
    <LabelPlacement>
      <GeometryType>DETAIL</GeometryType>
      <GeometryLevel/>
      <X>916614.15</X>
      <Y>757477.59</Y>
      <Rotation>0</Rotation>
      <FontSize>18</FontSize>
      <Text/>
    </LabelPlacement>
    <NormallyLooped>false</NormallyLooped>
    <NormallyDeenergized>false</NormallyDeenergized>
  </Switch>
</Substation>
```

```

<ServiceCenter>CTLB</ServiceCenter>
<SecondId>bla bla bla</SecondId>
<SwitchModel>1200_Standard</SwitchModel>
<NormallyOpen>true</NormallyOpen>
<SymbolPlacement>
    <GeometryType>DETAIL</GeometryType>
    <GeometryLevel/>
    <X>916694.061</X>
    <Y>757463.16</Y>
    <Rotation>3</Rotation>
    <Size>18</Size>
</SymbolPlacement>
</Switch>
<Feeder IncrChangeType = "Delete">
    <Id>146654973</Id>
</Feeder>
<Line IncrChangeType = "Create">
    <Id>NewLine</Id>
    <Name>Blalalala</Name>
    <FromNode>Node1</FromNode>
    <ToNode>Node2</ToNode>
    <Phase>ABCN</Phase>
    <ConstructionModel>Crossarm_Crossarm</ConstructionModel>
    <ConductorModelA>568_ACAR_Bare Stranded</ConductorModelA>
    <ConductorModelB>568_ACAR_Bare Stranded</ConductorModelB>
    <ConductorModelC>568_ACAR_Bare Stranded</ConductorModelC>
    <ConductorModelN>3/0_ACSR_Bare Stranded</ConductorModelN>
    <Length>335.89</Length>
    <LinePlacement>
        <GeometryType>DETAIL</GeometryType>
        <GeometryLevel/>
        <Vertex>
            <X>917069.06</X>
            <Y>754984.32</Y>
        </Vertex>
        <Vertex>
            <X>916733.17</X>
            <Y>754984.19</Y>
        </Vertex>
    </LinePlacement>
</Line>
</Substation>

```

Additional Notes:

- For an incremental file to update the attributes of a device, the corresponding XML element in the incremental file must contain the complete set of modeling attributes used. For example, if an attribute value changes for a switch object, the incremental file must contain all the switch attributes currently modeled instead of just providing the attribute that changes.
- The Model Manager uses a device's ID and DeviceType as the key to merge incremental and master files. For example, to update an existing Feeder object, the <Feeder> device type and <Id> of the feeder are used as the key within the substation.

- The Model Manager processes IncrChangeType attribute values at the Substation level as follows:
 - **Create:** The Model Manager does nothing because new substations must be defined by full substation files, not incremental files.
 - **Update:** The Model Manager updates only the elements at the Substation level, not the device elements below the Substation level, unless the elements have their own IncrChangeType XML attributes defined.
 - **Delete:** The Model Manager deletes the master substation file stored by the Model Manager.
- Station internals files cannot be updated using incremental update files.
- If a device is deleted in an incremental file, all possible occurrences of the device are deleted from the full file. For a switch that exists inside a container object (for example Cabinet, Vault, or UIContainer), a delete operation removes the Switch element and the Switch element references within the container object definition.

5.14 Automatically Creating SSG Files

When GISA sends new substation externals or internals files, the Model Manager can automatically create SSG files and proceed with creating and uploading models to complete the workflow.

If the “Automatic Creation of SSG” option on the PreInput tab is disabled, when a substation XML file appears in the Model Manager’s PreInput folder without an accompanying SSG file, the Model Manager does not process the XML file.

If the “Automatic Creation of SSG” option on the PreInput tab is enabled, when a substation file appears in the PreInput folder (either manually added by a user, or automatically published to the folder by GISA), the Model Manager detects the substation externals or internals file and automatically creates a corresponding SSG file for the substation. The file name format is <SubstationName>_<ExtractID>_AUTO.SSG, where ExtractID is optional and depends on the configuration. If the SSG file is successfully created, the Model Manager moves the SSG file and substation XML files to the Input folder for further processing and model conversion and validation.

An example of an auto-created SSG file is shown below. The Substation ID, Name, and Area Name (if available) values are extracted from the Substation XML file.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<SsgFile>
  <ID>MMA_SSG_RAVEN_2018_02_27_11_15_35</ID>
  <AreaNames>WST</AreaNames>
  <AutoProcess>True</AutoProcess>
  <DateTime>2018-02-27T11:15:35</DateTime>
  <TypeOfInput>Externals</TypeOfInput>
  <Substations>
    <Substation>
      <Name>RAVEN</Name>
      <ID>RAVEN</ID>
      <Description></Description>
      <AreaName>WST</AreaName>
    </Substation>
  </Substations>
</SsgFile>
```

For a substation incremental file, the Model Manager parses the CIM incremental file to extract relevant information (such as the work order number, changes in GIS, device IDs, change type, and descriptions) and uses it to create an SSG file.

```

<SsgFile xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <ID>MMA_SSG_BETHPAGE_2018_02_27_11_15_35</ID>
  <AutoProcess>True</AutoProcess>
  <DateTime>2018-02-27T11:15:35</DateTime>
  <TypeOfInput>Externals</TypeOfInput>
  <Substations>
    <Substation>
      <ID>BETHPAGE</ID>
      <Name>BETHPAGE</Name>
      <Description />
      <ChangedItems>
        <SubstationItem>
          <DeviceID>10081746XP</DeviceID>
          <AttributeName>Switch</AttributeName>
          <WorkOrderNumber>12345</WorkOrderNumber>
          <TypeChange>Added</TypeChange>
          <Description>Element Switch with ID 10081746XP was added</Description>
        </SubstationItem>
        <SubstationItem>
          <DeviceID>10082076XP</DeviceID>
          <AttributeName>Switch</AttributeName>
          <WorkOrderNumber>67890</WorkOrderNumber>
          <TypeChange>Deleted</TypeChange>
          <Description>Element Switch with ID 10082076XP was deleted</Description>
        </SubstationItem>
      </ChangedItems>
    </Substation>
  </Substations>
</SsgFile>

```

5.15 Customizing Model Manager

This section describes how to customize some aspects of the data displays in Model Manager and how to save your preferences.

5.15.1 Configuring Column Width and Order

Model Manager allows you to configure column width and column order.

To set the column width:

1. Hover the mouse cursor over the column header border.

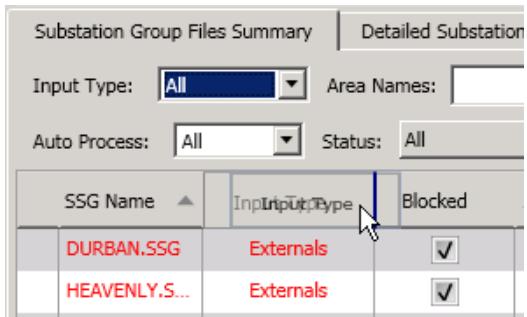
A double-sided arrow appears.

SSG Name	Input Type
DURBAN.SSG	Externals
HEAVENLY CCC	Externals

2. Drag the arrow to adjust the column width.

To configure the column order:

- Drag the column header to the required position.



5.15.2 Sorting Data

All data grids in the Model Manager support multiple sorting by columns.

Clicking the column header sorts the display by the data in that column, in descending order .

Clicking the column header again changes the sorting to ascending order . For multiple sorting, hold Shift and continue clicking on the required column headers.

SSG Name	Input Type	Area Names	Autoprocess	Status	Blocked	Number of Stations	Date/Time
MULTIPLE_001.SSG	Externals		True	ConvertingError	<input checked="" type="checkbox"/>	3	6/24/2014 3...
JUPITER.SSG	Externals		False	Publishing	<input checked="" type="checkbox"/>	1	5/19/2015 1...
DURBAN.SSG	Externals		False	Queued	<input checked="" type="checkbox"/>	1	5/19/2015 8...
MERCURY.SSG	Externals		False	ReadyForRevi...	<input checked="" type="checkbox"/>	1	5/19/2015 1...
HEAVENLY.SSG	Externals		False	UnderReview	<input checked="" type="checkbox"/>	1	5/19/2015 8...
RAVEN_001.SSG	Externals		False	ConvertingError	<input type="checkbox"/>	1	11/21/2014...
PLUTO.ISSG	Internals		True	Uploaded	<input type="checkbox"/>	1	6/24/2014 8...

Figure 33. Sorting Model Manager Data

5.15.3 Saving User Preferences

The Model Manager can store user preferences. The tool detects any changes to the user interface. When you close the tool, it prompts you about saving the changes. If you save the changes, the information is stored in an XML file and loaded when the Model Manager starts. The XML files are stored in the C:\Users\<User>\AppData\Local\General_Electric_Technology\ModelManagerApplication directory.

The following preferences are saved:

- Filter options from the Substation Group Files Summary, Detailed Substation Group Files Summary, and the History by Substation tabs.
- Column order and column sorting (ascending/descending) order from all data grids in the Model Manager.

6. Running the Stand-Alone Model Manager

This chapter describes the controls and features of the stand-alone Model Manager that differ from the Service Client.

6.1 Starting the Stand-Alone Client

You can access the Model Manager application from the Windows Start menu using the following path:

Start > Programs > Eterra > Distribution > Tools > Modeling Tools > Model Manager Application

This shortcut passes a configuration file to the application in the format:

[Model Manager Executable Path] [Model Manager Configuration XML file name]

The default configuration file parameter is:

```
D:\ettera\Distribution\Tools\bin\ModelManagerApplication.exe /CONFIG  
"%IDMS_ROOT%\Tools\Workspace\ModelManagerFolders\ModelManager.xml"
```

You can view and modify this parameter in the Target field of the shortcut's properties dialog box.

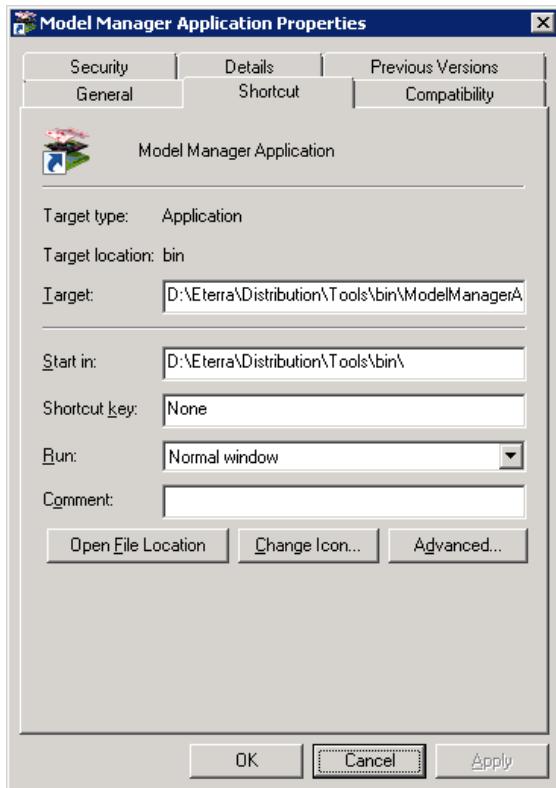


Figure 34. Model Manager Properties

The configuration file specifies (among other things) the directory structure and the location of the Converter and Navigation Overview Builder tool. You can change the default configuration by following the instructions in chapter 4 Configuring Model Manager.

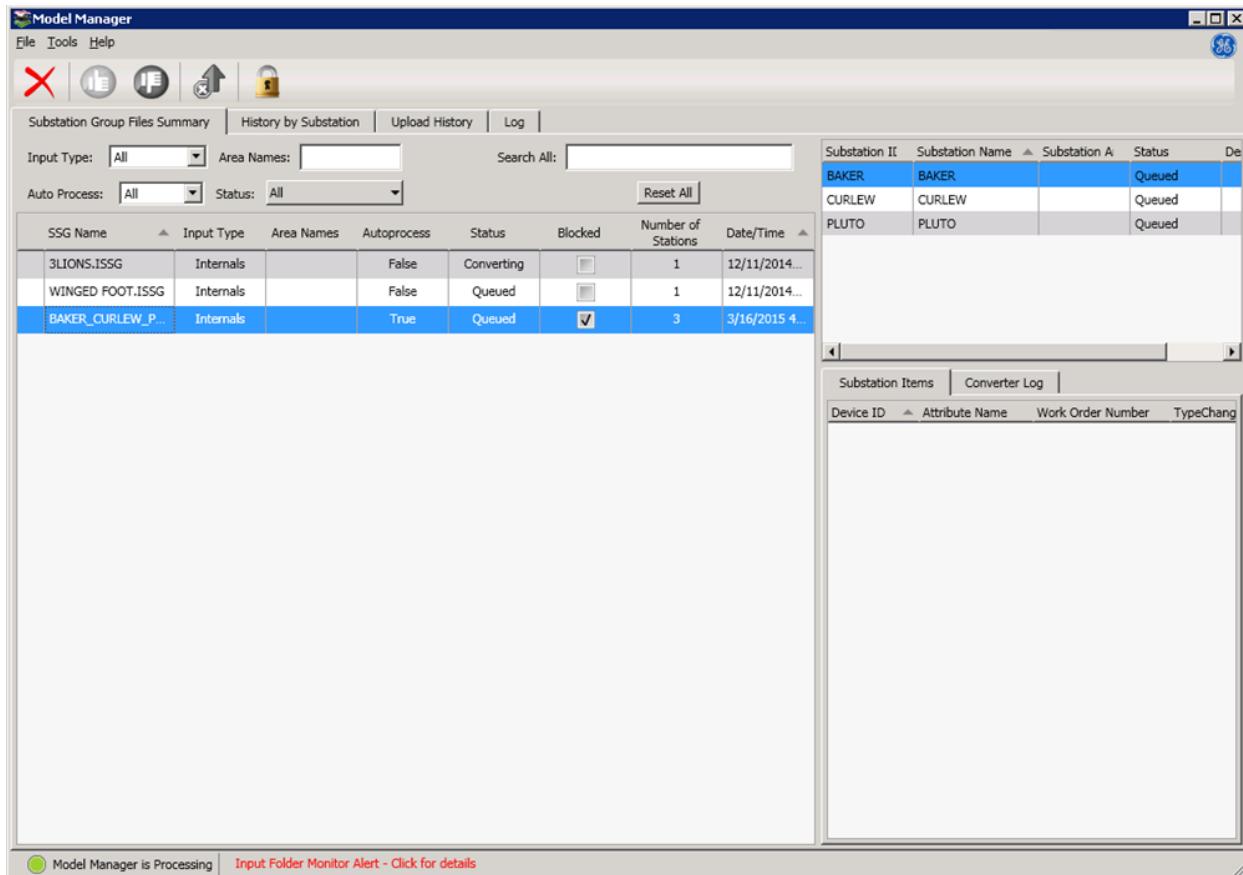


Figure 35. The Model Manager Application

6.2 Controls and Menus

In the stand-alone Model Manager client, the menu includes the following commands:

- **File > Exit:** Closes the application.
- **Tools > Configurations:** Opens the Model Manager Configuration pop-up display, where you can modify the locations for the Model Manager directory structure, the log file, the Converter tool, and the Navigation Overview Builder tool.
- **Tools > Build Nav Overview:** Opens a dialog box to build the overview.nvw file.
- **Help > About Model Manager:** Opens the About Model Manager dialog box with the copyright and version information.
- **Help > User's Guide:** Opens the Model Manager help.

6.3 Configuring the Stand-Alone Client

You can access the configuration pop-up display via the Tools > Configuration menu option.

The Configuration pop-up display is identical to the Model Manager Configuration Editor tool and uses the same configuration file. The latest changes saved in either the pop-up display or the Configuration

Editor override the previous parameters. For more information, refer to chapter 4 Configuring Model Manager.

The Configuration pop-up contains three controls that the Configuration Editor does not provide.

- **OK:** Saves the changes into the ModelManager.xml file and closes the pop-up display.
- **Cancel:** Discards the changes and closes the pop-up display.
- **Apply:** Saves the changes into the ModelManager.xml file without closing the pop-up display.

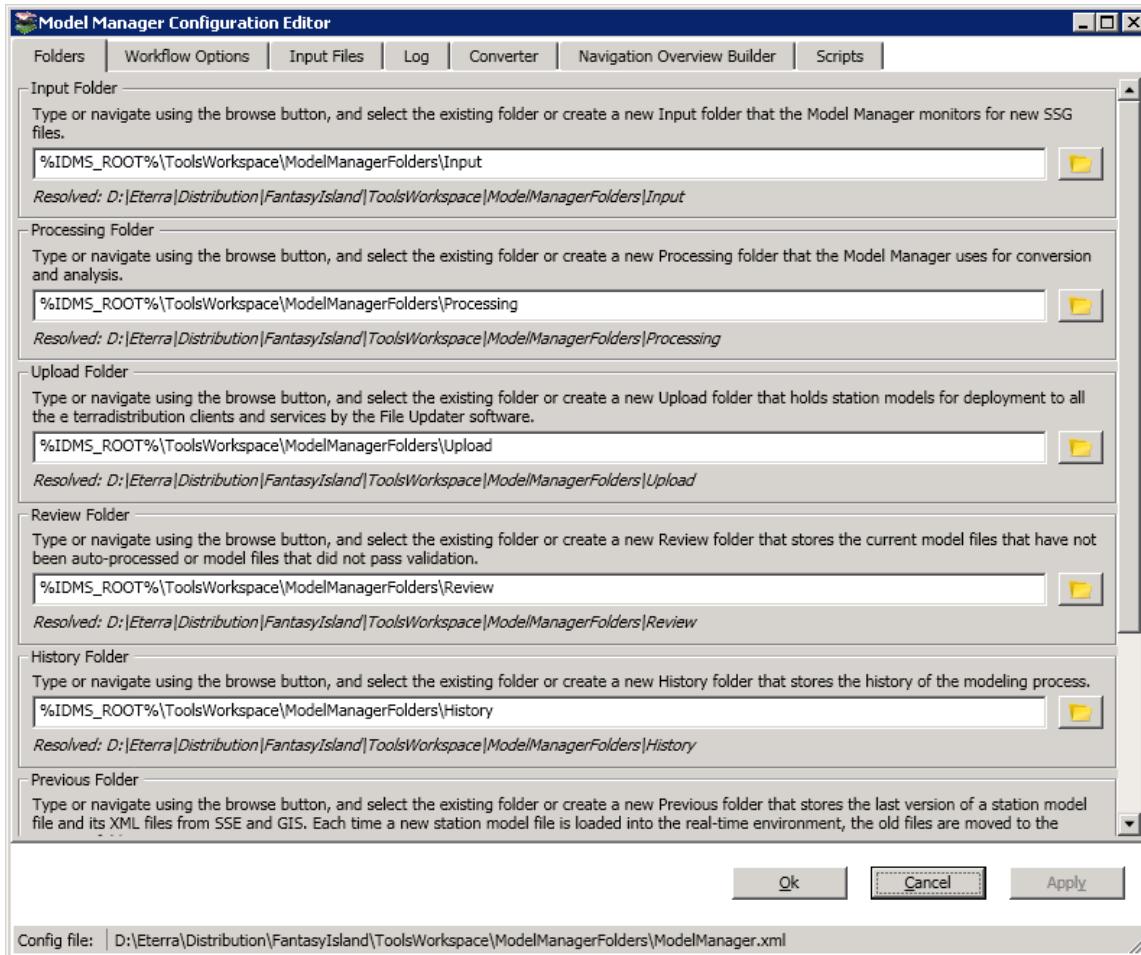


Figure 36. Model Manager Configuration Pop-Up

Appendix A. (I)SSG Validation Process

This appendix describes the validation by the Converter tool and the corresponding actions of the Model Manager tool for processing substation group files.

The table below summarizes the Converter validation process, including the resulting actions taken by the Model Manager for single and multiple station groups.

Table 1. Validation Rules for Single and Multiple Station Groups Processing

Converter Validation	Auto-Process?	Result
(I)SSG Valid	True	Station model files are copied to the Review folder and to the Upload folder. Station model files are then transferred (via SFTP or FTP) from the File Updater folder to the real-time ADMS and loaded online.
(I)SSG Invalid	True	<p>The Station Summary tab presents the UNBLOCK and DELETE buttons for this (I)SSG.</p> <ul style="list-style-type: none">• By clicking the UNBLOCK button, the (I)SSG file is unblocked and can be reprocessed.• By clicking the DELETE button, the (I)SSG file is removed from the list.
(I)SSG Valid	False	<p>The input files and station model files are copied to the Review folder.</p> <p>The Station Summary tab presents the APPROVE, REJECT, and DELETE buttons for this (I)SSG.</p> <ul style="list-style-type: none">• By clicking the APPROVE button, these station model files are approved and then copied to the Upload folder.• By clicking the REJECT button, uploading of these station model files is rejected.• By clicking the DELETE button, the (I)SSG file is removed from the list.
(I)SSG Invalid	False	<p>The Station Summary tab presents the UNBLOCK and DELETE buttons for this (I)SSG.</p> <ul style="list-style-type: none">• By clicking the UNBLOCK button, the (I)SSG file is unblocked and can be reprocessed.• By clicking the DELETE button, the (I)SSG file is removed from the list.

Appendix B. Model Manager SOAP Web Service

This appendix describes how the Model Manager SOAP web service supports communicating with the GIS Adapter to receive full and incremental station externals and internals files. The web service is loaded by the Model Manager application.

To enable the Model Manager SOAP web service, select the Enable SOAP Message Handling option on the PreInput Tab of the Model Manager Configuration Editor.

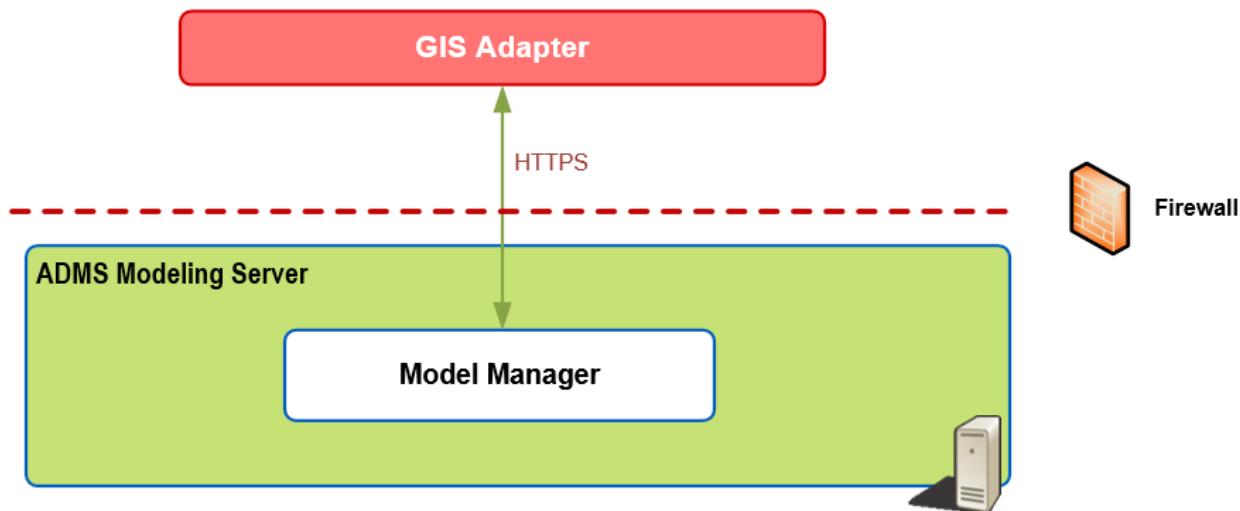


Figure 37. Model Manager - GIS Adapter Data Flow

The Model Manager web service is implemented using SOAP. For details about SOAP, refer to <https://www.w3.org/TR/soap/>.

The Model Manager web service is implemented using SOAP. For details about SOAP, refer to <https://www.w3.org/TR/soap/>. To secure the SOAP message exchange, HTTP Transport Layer Security (TLS) with mutual certificate authentication is used. The Model Manager web service is configured with a Secure Sockets Layer (SSL) X.509 certificate to allow the GIS Adapter client to verify the identity of the Model Manager server. The GISA client is also configured with an X.509 certificate that allows the Model Manager web service to verify the identity (client authentication) of the GISA client. The Model Manager server's certificate must be trusted by the GIS Adapter client and vice versa.

B.1 Service Messages

Model Manager web service messages are listed in the table below.

Request	Type	Response (Synchronous)	Response (Asynchronous)	Description
Consumer	Input	ConsumerResp onse	JobStatus	The GIS Adapter sends substation model data to the Model Manager.

Request	Type	Response (Synchronous)	Response (Asynchronous)	Description
GetJobStatusRequest	Input	GetJobStatusResponse	JobStatus	The GIS Adapter requests the processing status from the Model Manager.
JobStatus	Output	JobStatusResponse	NA	The Model Manager sends the processing status to the GIS Adapter.
GetNetworkDataSetResync	Output	GetNetworkDataSetResyncResponse	Consumer	The Model Manager requests resending substation model data from the GIS Adapter.

The details of each SOAP request are provided below.

Consumer

This request supports publishing full or incremental substation externals files from the GIS Adapter to the Model Manager.

The GIS Adapter publishes CIMDNON full and incremental substation model data files to the shared “Model Exchange” folder. The Consumer message conveys the names and publish location of the files.

Note: You can configure the File Updater to move these files to the PreInput folder.

The Consumer message contains details about the GIS Adapter Job and Substation names that are exported. The Model Manager parses this SOAP message, performs basic validation, and sends a synchronous ConsumerResponse message back to the GIS Adapter to indicate a successful receipt or an error. The Model Manager then creates an SSG file using the Consumer SOAP message and processes station files.

The key properties of the Consumer message are as follows:

- **JobId:** The unique identifier to reference a message exchange. This ID is derived from the message ID in the CIM metadata.
The ConsumerResponse message contains this identifier.
- **JobName:** For Initial Load and Resynch job states, this is the station name. For the AsBuilt job state, this is the Work Order name.
- **JobState:** The job state, which can be one of the following:
 - **InitialLoad:** Station initial load, for full model data.
 - **ReExport:** Station resynchronization, for full model data.
 - **AsBuilt:** Incremental update data, for model difference data.
 - **Proposed:** Incremental update data, for model difference data in the future network.
- **TimeStamp:** Date and time when the message exchange was performed.

ConsumerResponse

The Model Manager synchronously sends a ConsumerResponse message to the GIS Adapter in response to the Consumer message to indicate a successful message receipt or an error.

The ConsumerResponse message has the following properties:

- **Status:** Status shows 0 (zero) for success, or an integer value to represent an error.
- **ErrorMsg:** Shows a message describing the error. This value is stored in the CIM Metadata's Received Message Details field.

After sending a success synchronous ConsumerResponse message, the Model Manager processes the Consumer message internally.

The following steps show how the Model Manager processes a Consumer message with the “InitialLoad” or “ReExport” job state:

1. The Model Manager web service raises an event with Consumer message details as event arguments. The Model Manager server receives this event and adds it to the SOAP messages list.
2. The Model Manager processes each SOAP message in the message list. It checks for station data files specified in the Consumer message in the PreInput folder, creates an SSG file in the PreInput folder, and then moves the files to the Input folder.

If station data files are not available, the Model Manager waits for a configurable time period to allow transferring large files. If the time period elapses, the Model Manager sends an asynchronous JobStatus message to the GIS Adapter to communicate the status of the job.

The following steps are performed when the Model Manager processes a Consumer message with a job that has the “AsBuilt” job state:

1. The Model Manager web service raises an event with Consumer message details as event arguments. The Model Manager server receives this event and adds it to the SOAP messages list.
2. The Model Manager processes each SOAP message in the message list. It checks for station data files specified in the Consumer message in the PreInput folder.
3. When station data files are available in the PreInput folder, the Model Manager checks for a full substation externals file in the Input folder or in the Processing folder.
4. The Model Manager validates each incremental change before merging the change into a full substation externals file. If any discrepancy is found or the full station file is not available, the Model Manager sends an asynchronous JobStatus message to the GIS Adapter to convey the status of the job. After that, the message is updated with the Rejected status.
5. After successfully merging incremental changes to the full station file, the Model Manager creates an SSG file in the PreInput folder and moves the files to the Input folder.

JobStatus

The Model Manager uses the JobStatus message to send job status updates to the GIS Adapter.

After the Model Manager moves substation data and SSG files to the input folder, the Model Manager performs the model build and upload process. During this process, the Model Manager issues several JobStatus messages to communicate the progress of the export in the Model Manager.

The JobStatus message has the following key parameters:

- **JobId:** Contains the JobId from the Consumer message. The GIS Adapter uses this message to link the incoming JobStatus update and store it in the CIM Metadata.
- **JobStatus:** The job status (Queued, Creating, Created, Introducing, Ready, Rejected, or Accepted).
- **JobStatusComment:** This field is used to pass additional information, such as a Model Manager job rejection reason.

When the Model Manager performs the model build process, the tool goes through several workflow states. These internal states are not sent to the GIS Adapter. A consolidated state is sent instead. The following table shows the mapping between the internal and consolidated states.

Model Manager Internal State	Model Manager Consolidated State
Queued, VerifyingInputData, MovingToProcessing	Queued
Converting	Creating
ReadyForReview, ModelNeedsReview	Created
Publishing	Introducing
SSGError, ErrorCopyInputToProcessing, ConverterFatal, UnderReview, PublishingError	Ready
Timeout, SchemaValidationFailed, ConvertingError, Rejected, UnableToRestore	Rejected
Uploaded	Accepted

JobStatusResponse

After receiving a JobStatus message, the GIS Adapter sends a synchronous JobStatusResponse acknowledgement message with the processing status (success or error) to the Model Manager.

GetJobStatus

The GIS Adapter user can request an update of the model import state at any time using the Network Model Exchange dialog.

The GetJobStatus request passes details of the job of interest. After receiving the request, the Model Manager checks the status of the job and responds using an asynchronous JobStatus message.

The GetJobStatus message has the following key properties:

- **JobId:** The unique identifier to reference a message exchange. This ID is derived from the message ID in the CIM metadata.
- **Station MRID, Station ID, Station Name:** The identifiers assigned by the GIS Adapter as the unique reference for one of the stations associated with this job. This information is included to enable content-based routing when a customer has a single GIS database with multiple regional ADMS instances.
- **AsyncReplyFlag:** Indicates whether the GIS Adapter expects an asynchronous JobStatus service response message with the current job status or the GetJobStatusResponse message.

Note: Currently, the GIS Adapter is configured to expect the asynchronous JobStatus message and has not implemented the GetJobStatusResponse message.

GetJobStatusResponse

After receiving a GetJobStatus request from GISA, MMA performs the basic validation and sends a synchronous SOAP message with the name “GetJobStatusResponse” back to GISA to indicate a successful receipt or an error.

The response message has the following key properties:

- **Result:** Indicates the result of the request (OK or FAILED).
- **Code:** Error code.
- **Reason:** Reason for the error to occur.

After sending a synchronous response, MMA processes the GetJobStatus SOAP message internally as follows:

1. The Model Manager web service raises an event with SOAP message details as event arguments. The Model Manager server receives this event and adds it to the SOAP messages list.
2. The Model Manager processes each SOAP message in this list. Using the ID of the Job, it searches for the related SSG history file.
 - If the history file exists, the Model Manager uses it to create an asynchronous JobStatus SOAP message and sends the message to the GIS Adapter.
 - If the history file does not exist, the Model Manager creates an asynchronous JobStatus SOAP message with a “Rejected” job status and sends it to the GIS Adapter.

GetNetworkDataSetResync

The Model Manager uses the GetNetworkDataSetResync message (by selecting the Place Resync Substation Request option for substations) to request resynchronizing (resending) a substation data. The GIS Adapter creates a model exchange job, adds it to a job queue, and sends a synchronous GetNetworkDataSetResyncResponse response. This job is processed similar to a Consumer message with the InitialLoad job.

The GetNetworkDataSetResync message has the following key properties:

- **ID:** Identifies the required station. For Electric Office stations, this is the Station ID value.
- **Name:** Identifies the required station. For Electric Office stations, this is the Station Name value.

GetNetworkDataSetResyncResponse

After the GIS Adapter receives a GetNetworkDataSetResync message and creates a job, the GIS Adapter sends a GetNetworkDataSetResyncResponse message.

This response message has the following properties:

- **Result:** Describes the result, success, or error.
- **Code:** The error code.
- **Reason:** The description of the error.

B.2 Debug Service Messages

You may need to inspect SOAP messages that are exchanged between the Model Manager and GIS Adapter. To support this, use trace logs that contain all SOAP request messages responses among the Model Manager and GIS Adapter. SOAP messages are logged when the Trace Logging Level value is set to “Debug1” on the Log Tab of the Model Manager Configuration Editor.

B.2.1 Consumer Soap Message Request Fields

Element	Type	Nullable?	Description
MessageHeader	MessageHeaderStc	No	Header information.
Verb	string	No	This varies depending on whether it is pushing or pulling information. The allowable values are: create, created, get, update, and delete. Max Length = 25 characters.
Noun	string	No	The noun is NetworkDataSet for connection information. The noun is NetworkDataSet for both initial load and incremental load. Max Length = 64 characters.
Timestamp	DateTime	No	The time that the data is created for export.
Revision	string	No	The CIM Model version and Profile version. Max Length = 64 characters.
Source	string	No	The source system name. For example, GIS_Texas. Max Length = 64 characters.
ReplyAddress	string	No	The URI to which the system may reply if there is a reply message. Max Length = 2083 characters. Note: MMA does not currently use this field.
DataVersion	string	No	The version of the data. Max Length = 64 characters.
Nonce	string	No	A number generated once for security between client and server. Max Length = 32 characters. Note: MMA does not currently use this field.
JobID	string	No	A unique human-readable identifier. Examples: <ul style="list-style-type: none">• JOB-1• Project A: Design 1 Max Length = 64 characters.

Element	Type	Nullable?	Description
JobState	string	Yes	<p>One of the following values:</p> <ul style="list-style-type: none"> InitialLoad: The DNOM contains an initial load. AsBuilt: The update contains changes that reflect the as-built network. This results in MMA processing incremental station files. ReExport: This is a re-export of an initial load. Proposed: The update contains changes that reflect the future network. This results in MMA processing incremental station files. <p>For backwards compatibility, if this field is not provided, it is assumed to be AsBuilt.</p>
JobName	string	No	<p>A summary of the job. For Initial Load and ReExport job states, this is the station name. For the AsBuilt job state, this is the Work Order name.</p> <p>Examples:</p> <ul style="list-style-type: none"> Add pole at XXX Project A: Design 1: Add pole at XXX <p>Max Length = 256 characters.</p>
JobDescription	string	No	<p>A verbose description of the job. Carriage returns are allowed.</p> <p>Example:</p> <ul style="list-style-type: none"> Add pole type YYY to XXX. Ensure the pole is at least 5m away from the road. <p>Max Length = 512 characters.</p> <p>Note: This field is not currently used in MMA.</p>
JobComments	string	Yes	<p>This is extra information related to the job. Carriage returns are allowed. Optional field.</p> <p>Example:</p> <ul style="list-style-type: none"> Avoid disturbing the nearby wildlife. <p>Max Length = 2000 characters.</p> <p>Note: This field is not currently used in MMA.</p>
AutoIntroduce	string	Yes	<p>A flag to indicate whether or not the changes defined by this message should be auto-introduced.</p> <p>The allowable values are: true and false.</p> <p>Max Length = 5 characters.</p> <p>This field is case sensitive.</p> <p>Note: This field is not currently used in MMA.</p>

Element	Type	Nullable?	Description
CorrelationID	string	Yes	<p>This field is used to logically link messages together.</p> <p>It should only be populated when the Consumer message has been sent as an asynchronous response to MMA sending a GetNetworkDataSetResync message. In this case, the field will contain the CorrelationID or MessageID from the resync request message.</p>
AffectedDataSources	string	No	<p>Specifies the stations that are affected in this job. The value of the property contains comma separated station mRIDs.</p> <p>For example:</p> <ul style="list-style-type: none"> • sub_substation_1168768, sub_substation_1168768 <p>This field is useful while processing incremental updates where there could be multiple stations.</p>
SourceDataTimestamp	string	No	Specifies the GISA database server instance timestamp.
MessagePayload	MessagePayloadStc	No	Specifies the location of the DNOM files.
JobData	URIValuePairStc		<p>For external systems that can package up changes into a job, this field contains the DNOM and GML files for that job. For external systems that can only package up changes by circuit, use the CircuitData field instead.</p> <p>For an initial load, the CIM is in a DNOM file.</p> <p>For an incremental update, this field currently only supports a “combined” DNOM file, which contains the created, updated, and deleted DNOM objects. This field does not support the separation of created, updated, and deleted DNOM objects into different DNOM files.</p>
CircuitData	CircuitDataStc	No	<p>This array of CircuitData structures is primarily intended for systems that export by circuit.</p> <p>For an initial load, the CIM is in a RDF file and is specified in the Created field below.</p> <p>For an incremental update, this field only supports the separation of created, updated, and deleted CIM objects into different CIM RDF diffModel files. This field does not support a “combined” CIM file where this CIM file contains the created, updated, and deleted CIM object in one RDF diffModel file.</p> <p>Note: MMA does not currently use this field.</p>
CircuitID	string	No	<p>Unique identifier that specifies which circuit is being amended.</p> <p>Max Length = 32 characters.</p>

Element	Type	Nullable?	Description
Created	URIValuePairStc	Yes	Optional structure that contains URIs for the CIM and, if there is one, the GML file that contains information about components to be created in the given circuit.
Updated	URIValuePairStc	Yes	Optional structure that contains URIs for the CIM and, if there is one, the GML file that contains information about existing components in the given circuit that should be updated.
Deleted	URIValuePairStc	Yes	Optional structure that contains URIs for the CIM and, if there is one, the GML file that contains information about components to be deleted from the given circuit.

B.2.2 ConsumerResponse Soap Message Request Fields

Element	Type	Nullable?	Description
ConsumerResponse	ChangeNetworkDataSetResponseMessageType	No	Response for Consumer request.
Status	integer	No	An integer that is 0 for success or non-zero to represent an error.
ErrorMsg	string	No	A string that can be populated with a message describing the error. This will be stored in the CIM Metadata Received Message details field.

B.2.3 GetJobStatus Soap Message Request Fields

Element	Type	Nullable?	Description
Header	GetJobStatusHeaderType	No	Header information.
Verb	string	No	The allowable value is: get. Max Length = 25 characters.
Noun	string	No	The noun is: JobStatus. Max Length = 64 characters.
Timestamp	DateTime	No	The time that the job status was requested.
Source	string	No	The source system the message originated from. For example: GIS_Texas. Max Length = 64 characters.
AsyncReplyFlag	Boolean	No	Indicates whether a reply message will be sent asynchronously. The value will be set to true. MMA is configured to send an asynchronous jobStatus soap message reply to a GetJobStatus request.

Element	Type	Nullable?	Description
Circuit	CircuitStc	Yes	List of station details. Can be used for routing the message if necessary.
mRID	string	No	The unique master resource identifier for a station. Max Length = 128 characters. Note: GISA sends the ID of the station with this field.
ID	string	Yes	The ID of the station.
Name	string	Yes	The name of the station.
Request	GetJobStatusRequestType	Yes	Request information.
JobID	string	No	The unique human-readable identifier for a job. Max Length = 64 characters.
DataVersion	string	Yes	The version of the data. Max Length = 64 characters.

B.2.4 GetJobStatusResponse Soap Message Request Fields

Element	Type	Nullable?	Description
Header	GetJobStatusHeaderType	No	Header information.
Header	HeaderType	No	Header information.
Verb	string	No	The value is: reply.
Noun	string	No	The value is: JobStatus.
Timestamp	DateTime	No	The time that the reply is sent.
Source	string	No	The source system. The value populated is: ADMS DNOM.
Reply	ReplyType	No	Reply information. Includes the result or error information, as appropriate.
Result	string	No	A value that indicates the result of the request. The following will be returned: <ul style="list-style-type: none"> • OK: Fetched the job status successfully. • FAILED: An error occurred.
Error	ErrorType	Yes	Error information.
code	string	No	Error code.
reason	string	Yes	Free form detailed text description of the error.

Element	Type	Nullable?	Description
PayLoad	JobStatusPayLoa dType	Yes	<p>Contains the fetched job status.</p> <p>Note: Currently no PayLoad is sent with a GetJobStatusResponse message. Instead, job status is sent through an asynchronous JobStatus message.</p>
JobStatus	JobStatusType (an enumeration of values)	No	<p>The verbose status of the job.</p> <p>See appendix B.2.6 JobStatusType Enumerated Values for the possible values defined for the type.</p>

B.2.5 JobStatus Soap Message Fields

Element	Type	Nullable?	Description
MessageHeader	MessageHeaderStc	No	Header information.
MessageID	string	No	<p>Contains the same ID as the SOAP header and may be used to uniquely identify a message.</p> <p>Max Length = 32 characters.</p>
Verb	string	No	<p>The allowable value is: created.</p> <p>Max Length = 25 characters.</p>
Noun	string	No	<p>The noun is JobStatus when the message represents rejected circuits.</p> <p>Max Length = 64 characters.</p>
Timestamp	DateTime	Yes	The time that the data is created for export.
Source	string	Yes	<p>The source system. The value is: ADMS DNOM.</p> <p>Max Length = 64 characters.</p>
DataVersion	string	Yes	<p>The data version.</p> <p>Max Length = 64 characters.</p>
JobID	string	Yes	<p>A unique human-readable identifier.</p> <p>Max Length = 64 characters.</p>
JobStatus	JobStatusType	Yes	<p>For external systems that can package up changes into a job, this field contains the status of the job. For the possible values this field could contain, see appendix B.2.6 JobStatusType Enumerated Values.</p>
JobStatusComment	string	Yes	<p>For external systems that can package up changes into a job, this field contains additional information about its status (for example, if rejected, holds the reject reason).</p>
MessagePayload	MessagePayloadStc	Yes	<p>Sequence of ProcessedCircuit elements.</p> <p>This field is only applicable for source systems that have exported by station.</p>

Element	Type	Nullable?	Description
ProcessedCircuit (repeat)	ProcessedCircuitStc	Yes	This structure describes the success of an update for each station that was processed. If a station is rejected, a reason is associated with it.
CircuitID	String	No	Identifies the processed station.
JobStatus	JobStatusType	No	The status of the job. For the possible values this field could contain, see appendix B.2.6 JobStatusType Enumerated Values.
JobStatusComment	string	No	Additional information about its status (for example, if rejected, holds the reject reason).

B.2.6 JobStatusType Enumerated Values

JobStatusType	Meaning
Queued	A Substation Group (SSG) file is created for the GISA job and it is in the queue for processing.
Creating	The SSG file is being processed by the Converter tool.
Created	The SSG file is successfully processed by the Converter tool and its output (MOD files) is created.
Ready	The SSG file review is complete and is now ready for introducing.
Introducing	Model files are produced, and the Converter log is validated. The MOD file is moving to the Uploaded folder.
Accepted	The SSG file has been manually or automatically approved and uploaded and a Network Model Exchange Job has been successfully implemented.
Rejected	Rejected for one of the following reasons: <ul style="list-style-type: none"> • The SSG file did not pass schema validation. • The SSG file did not pass validation by the Converter tool. • All input files were not copied to the Input folder during the pre-defined time span. • The Ready for Review SSG file was rejected by the user. • The workflow could not be recovered from the SSG file.

B.2.7 GetNetworkDataSetResync Soap Message Fields

Element	Type	Nullable?	Description
MessageHeader	HeaderType	No	Header information.
Verb	String	No	The request verb. The value is: get.
Noun	String	No	The request noun. The value is: NetworkDataSetResync.
Timestamp	DateTime	No	The time that the request is being sent.

Element	Type	Nullable?	Description
Source	String	No	The source system. The value is: ADMS DNOM.
AsyncReplyFlag	Boolean	Yes	Indicates whether a reply message will be sent asynchronously. The value will be set to true. The external system should reply with an acknowledgement that the request was received. Then the actual resynchronization should be sent to MMA asynchronously as a Consumer message.
MessageID	String	Yes	A string that uniquely identifies the message.
CorrelationID	String	Yes	An ID that is used to logically link messages. When a value is provided in the CorrelationID field, the external system should use this value in the CorrelationID fields in all related responses (GetNetworkDataSetResyncResponse) to this message. If the field is not filled in, the external system should use MessageID instead in all related responses. Note: MMA will not populate this field. The definition is included to follow the IEC 61968-100 standard.
Request	GetNetworkData SetResyncReques tType	Yes	Request information.
Circuit (repeat)	CircuitStc	No	List of the circuits to resynchronize.
mRID	string	No	mRID of the circuit to resynchronize.
ID	string	Yes	Circuit ID.
Name	string	Yes	Circuit Name.
User	UserType	Yes	Username requesting the resynchronization.
UserID	string	No	ID of the user requesting the resynchronization. MMA will fill in the username of the operator in this field.
Description	string	Yes	A description associated to the resynchronization request.

B.2.8 GetNetworkDataSetResyncResponse Soap Message Fields

Element	Type	Nullable?	Description
Header	HeaderType	No	Header information
Verb	string	No	The value is: reply.
Noun	string	No	The value is: NetworkDataSetResync.

Element	Type	Nullable?	Description
Timestamp	DateTime	Yes	The time that the reply is sent.
Source	string	Yes	The source system name. The value is: GIS.
CorrelationID	string	Yes	An ID used to logically link messages together. See MessageID and CorrelationID in the Header of the GetNetworkDataSetResync message. If the request includes a CorrelationID, the external system uses that value in this field. Otherwise, it uses the MessageID from the request.
Reply	ReplyType	No	Reply information. Includes the result or error information, as appropriate.
Result	string	No	A value that indicates the result of the request. One of the following should be returned: OK, PARTIAL, or FAILED.
Error	ErrorType	Yes	Error Information. If the reply's Result is PARTIAL or FAILED, any related information will be provided by one or more Error Types.
code	string	No	Error code.
reason	string	Yes	Free form for a detailed text description of the error.

B.3 Message Schemas

ConsumerRequestSoapMessage

```

<?xml version="1.0" encoding="utf-8"?>
<xss:schema elementFormDefault="qualified"
  xmlns:xss="http://www.w3.org/2001/XMLSchema">
  <xss:element name="ConsumerRequestSoapMessage" nullable="true"
    type="ConsumerRequestSoapMessage" />
  <xss:complexType name="ConsumerRequestSoapMessage">
    <xss:sequence>
      <xss:element minOccurs="0" maxOccurs="1" name="JobId" type="xs:string" />
      <xss:element minOccurs="0" maxOccurs="1" name="JobName" type="xs:string" />
      <xss:element minOccurs="1" maxOccurs="1" name="TimeStamp" type="xs:dateTime" />
      <xss:element minOccurs="0" maxOccurs="1" name="Source" type="xs:string" />
      <xss:element minOccurs="0" maxOccurs="1" name="JobState" type="xs:string" />
      <xss:element minOccurs="0" maxOccurs="1" name="ReplyAddress" type="xs:string" />
      <xss:element minOccurs="0" maxOccurs="1" name="CorrelationID" type="xs:string" />
    </xss:sequence>
    <xss:element minOccurs="0" maxOccurs="1" name="CIMURL" type="xs:string" />
    <xss:element minOccurs="0" maxOccurs="1" name="unZippedFiles_GISA"
      type="ArrayOfString" />
      <xss:element minOccurs="0" maxOccurs="1" name="subStationsInvolved"
        type="ArrayOfString" />
        <xss:element minOccurs="0" maxOccurs="1" name="mRIDSIInvolved"
          type="ArrayOfString" />

```

```

        <xs:element minOccurs="1" maxOccurs="1" name="SoapMessageType"
type="eSoapMessageType" />
        <xs:element minOccurs="1" maxOccurs="1" name="SoapMessageState"
type="eSoapMessageState" />
        <xs:element minOccurs="0" maxOccurs="1" name="SSGFileName" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="Comment" type="xs:string" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="ArrayOfString">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="string" nillable="true"
type="xs:string" />
    </xs:sequence>
</xs:complexType>
<xs:simpleType name="eSoapMessageType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="None" />
        <xs:enumeration value="ConsumerRequest" />
        <xs:enumeration value="GetJobStatusRequest" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="eSoapMessageState">
    <xs:restriction base="xs:string">
        <xs:enumeration value="None" />
        <xs:enumeration value="Queued" />
        <xs:enumeration value="Processed" />
        <xs:enumeration value="Rejected" />
        <xs:enumeration value="InputDataProcessingError" />
        <xs:enumeration value="InputDataValidation Error" />
    </xs:restriction>
</xs:simpleType>
</xs:schema>
```

ConsumerRequest

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="ConsumerRequest" nillable="true" type="ConsumerRequest" />
    <xs:complexType name="ConsumerRequest">
        <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="consumerField"
type="ChangeNetworkDataSetRequestMessageType" />
            <xs:element minOccurs="0" maxOccurs="1" name="Consumer"
type="ChangeNetworkDataSetRequestMessageType" />
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="ChangeNetworkDataSetRequestMessageType">
        <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="MessageHeader"
type="ChangeNetworkDataSetHeaderType" />
            <xs:element minOccurs="0" maxOccurs="1" name="MessagePayload"
type="ChangeNetworkDataSetPayloadType" />
        </xs:sequence>
    </xs:complexType>
```

```

<xs:complexType name="ChangeNetworkDataSetHeaderType">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="DataVersion" type="xs:string" />
    <xs:element minOccurs="0" maxOccurs="1" name="JobComments" type="xs:string" />
    <xs:element minOccurs="0" maxOccurs="1" name="JobDescription" type="xs:string" />
  />
    <xs:element minOccurs="0" maxOccurs="1" name="JobID" type="xs:string" />
    <xs:element minOccurs="0" maxOccurs="1" name="JobName" type="xs:string" />
    <xs:element minOccurs="0" maxOccurs="1" name="JobState" type="xs:string" />
    <xs:element minOccurs="0" maxOccurs="1" name="Nonce" type="xs:string" />
    <xs:element minOccurs="0" maxOccurs="1" name="Noun" type="xs:string" />
    <xs:element minOccurs="0" maxOccurs="1" name="ReplyAddress" type="xs:string" />
    <xs:element minOccurs="0" maxOccurs="1" name="Revision" type="xs:string" />
    <xs:element minOccurs="0" maxOccurs="1" name="Source" type="xs:string" />
    <xs:element minOccurs="1" maxOccurs="1" name="Timestamp" type="xs:dateTime" />
    <xs:element minOccurs="0" maxOccurs="1" name="Properties"
      type="ArrayOfProperty" />
      <xs:element minOccurs="0" maxOccurs="1" name="Verb" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="AutoIntroduce" type="xs:string" />
    />
      <xs:element minOccurs="0" maxOccurs="1" name="CorrelationID" type="xs:string" />
    />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ArrayOfProperty">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="Property" nillable="true"
      type="Property" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="Property">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Name" type="xs:string" />
    <xs:element minOccurs="0" maxOccurs="1" name="Value" type="xs:string" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ChangeNetworkDataSetPayloadType">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="JobData" type="ArrayOfURIPair" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ArrayOfURIPair">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="URIPair" nillable="true"
      type="URIPair" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="URIPair">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="CIMURL" type="xs:string" />
    <xs:element minOccurs="0" maxOccurs="1" name="GMLURL" type="xs:string" />
  </xs:sequence>
</xs:complexType>
</xs:schema>

```

ChangeNetworkDataSetRequestMessageType

```
<?xml version="1.0" encoding="utf-8"?>
<xss:schema elementFormDefault="qualified"
  xmlns:xss="http://www.w3.org/2001/XMLSchema">
  <xss:element name="ChangeNetworkDataSetRequestMessageType" nillable="true"
    type="ChangeNetworkDataSetRequestMessageType" />
  <xss:complexType name="ChangeNetworkDataSetRequestMessageType">
    <xss:sequence>
      <xss:element minOccurs="0" maxOccurs="1" name="MessageHeader"
        type="ChangeNetworkDataSetHeaderType" />
      <xss:element minOccurs="0" maxOccurs="1" name="MessagePayload"
        type="ChangeNetworkDataSetPayloadType" />
    </xss:sequence>
  </xss:complexType>
  <xss:complexType name="ChangeNetworkDataSetHeaderType">
    <xss:sequence>
      <xss:element minOccurs="0" maxOccurs="1" name="DataVersion" type="xs:string" />
      <xss:element minOccurs="0" maxOccurs="1" name="JobComments" type="xs:string" />
      <xss:element minOccurs="0" maxOccurs="1" name="JobDescription" type="xs:string"
        type="ArrayOfProperty" />
      <xss:element minOccurs="0" maxOccurs="1" name="JobID" type="xs:string" />
      <xss:element minOccurs="0" maxOccurs="1" name="JobName" type="xs:string" />
      <xss:element minOccurs="0" maxOccurs="1" name="JobState" type="xs:string" />
      <xss:element minOccurs="0" maxOccurs="1" name="Nonce" type="xs:string" />
      <xss:element minOccurs="0" maxOccurs="1" name="Noun" type="xs:string" />
      <xss:element minOccurs="0" maxOccurs="1" name="ReplyAddress" type="xs:string" />
      <xss:element minOccurs="0" maxOccurs="1" name="Revision" type="xs:string" />
      <xss:element minOccurs="0" maxOccurs="1" name="Source" type="xs:string" />
      <xss:element minOccurs="1" maxOccurs="1" name="Timestamp" type="xs:dateTime" />
      <xss:element minOccurs="0" maxOccurs="1" name="Properties"
        type="Property" />
      <xss:element minOccurs="0" maxOccurs="1" name="Verb" type="xs:string" />
      <xss:element minOccurs="0" maxOccurs="1" name="AutoIntroduce" type="xs:string"
        type="CorrelationID" />
    </xss:sequence>
  </xss:complexType>
  <xss:complexType name="ArrayOfProperty">
    <xss:sequence>
      <xss:element minOccurs="0" maxOccurs="unbounded" name="Property" nillable="true"
        type="Property" />
    </xss:sequence>
  </xss:complexType>
  <xss:complexType name="Property">
    <xss:sequence>
      <xss:element minOccurs="0" maxOccurs="1" name="Name" type="xs:string" />
      <xss:element minOccurs="0" maxOccurs="1" name="Value" type="xs:string" />
    </xss:sequence>
  </xss:complexType>
  <xss:complexType name="ChangeNetworkDataSetPayloadType">
    <xss:sequence>
      <xss:element minOccurs="0" maxOccurs="1" name="JobData" type="ArrayOfURIPair" />
    </xss:sequence>
  </xss:complexType>
</xss:schema>
```

```

</xs:complexType>
<xs:complexType name="ArrayOfURIPair">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="URIPair" nillable="true"
type="URIPair" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="URIPair">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="CIMURL" type="xs:string" />
    <xs:element minOccurs="0" maxOccurs="1" name="GMLURL" type="xs:string" />
  </xs:sequence>
</xs:complexType>
</xs:schema>

```

ChangeNetworkDataSetResponseMessageType

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="ChangeNetworkDataSetResponseMessageType" nillable="true"
type="ChangeNetworkDataSetResponseMessageType" />
  <xs:complexType name="ChangeNetworkDataSetResponseMessageType">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="ErrorMsg" type="xs:string" />
      <xs:element minOccurs="1" maxOccurs="1" name="Status" type="xs:int" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>

```

ChangeNetworkDataSetHeaderType

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="ChangeNetworkDataSetHeaderType" nillable="true"
type="ChangeNetworkDataSetHeaderType" />
  <xs:complexType name="ChangeNetworkDataSetHeaderType">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="DataVersion" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="JobComments" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="JobDescription" type="xs:string"
/>
      <xs:element minOccurs="0" maxOccurs="1" name="JobID" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="JobName" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="JobState" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="Nonce" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="Noun" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="ReplyAddress" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="Revision" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="Source" type="xs:string" />
      <xs:element minOccurs="1" maxOccurs="1" name="Timestamp" type="xs:dateTime" />
      <xs:element minOccurs="0" maxOccurs="1" name="Properties"
type="ArrayOfProperty" />
      <xs:element minOccurs="0" maxOccurs="1" name="Verb" type="xs:string" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>

```

```

        <xs:element minOccurs="0" maxOccurs="1" name="AutoIntroduce" type="xs:string"
/>
        <xs:element minOccurs="0" maxOccurs="1" name="CorrelationID" type="xs:string"
/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="ArrayOfProperty">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="Property" nillable="true"
type="Property" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="Property">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="Name" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="Value" type="xs:string" />
    </xs:sequence>
</xs:complexType>
</xs:schema>

```

Property

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="Property" nillable="true" type="Property" />
    <xs:complexType name="Property">
        <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="Name" type="xs:string" />
            <xs:element minOccurs="0" maxOccurs="1" name="Value" type="xs:string" />
        </xs:sequence>
    </xs:complexType>
</xs:schema>

```

ChangeNetworkDataSetPayloadType

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="ChangeNetworkDataSetPayloadType" nillable="true"
type="ChangeNetworkDataSetPayloadType" />
    <xs:complexType name="ChangeNetworkDataSetPayloadType">
        <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="JobData" type="ArrayOfURIPair" />
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="ArrayOfURIPair">
        <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="unbounded" name="URIPair" nillable="true"
type="URIPair" />
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="URIPair">
        <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="CIMURL" type="xs:string" />

```

```

    <xs:element minOccurs="0" maxOccurs="1" name="GMLURL" type="xs:string" />
  </xs:sequence>
</xs:complexType>
</xs:schema>

```

URIPair

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="URIPair" nillable="true" type="URIPair" />
  <xs:complexType name="URIPair">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="CIMURL" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="GMLURL" type="xs:string" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>

```

GetJobStatusRequest

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="GetJobStatusRequest" nillable="true" type="GetJobStatusRequest"
/>
  <xs:complexType name="GetJobStatusRequest">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="Header"
        type="GetJobStatusHeaderType" />
      <xs:element minOccurs="0" maxOccurs="1" name="Request"
        type="GetJobStatusRequestType" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="GetJobStatusHeaderType">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="Verb" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="Noun" type="xs:string" />
      <xs:element minOccurs="1" maxOccurs="1" name="Timestamp" type="xs:dateTime" />
      <xs:element minOccurs="0" maxOccurs="1" name="Source" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="Station" type="ArrayOfStationStc"
/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ArrayOfStationStc">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="StationStc"
        nillable="true" type="StationStc" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="StationStc">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="ID" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="Name" type="xs:string" />
    </xs:sequence>
  
```

```

</xs:complexType>
<xs:complexType name="GetJobStatusRequestType">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="JobID" type="xs:string" />
    <xs:element minOccurs="0" maxOccurs="1" name="DataVersion" type="xs:string" />
  </xs:sequence>
</xs:complexType>
</xs:schema>

```

ConsumerResponse

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="ConsumerResponse" nillable="true" type="ConsumerResponse" />
  <xs:complexType name="ConsumerResponse">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="consumerResponse1"
        type="ChangeNetworkDataSetResponseMessageType" />
      <xs:element minOccurs="0" maxOccurs="1" name="ConsumerResponse1"
        type="ChangeNetworkDataSetResponseMessageType" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ChangeNetworkDataSetResponseMessageType">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="ErrorMsg" type="xs:string" />
      <xs:element minOccurs="1" maxOccurs="1" name="Status" type="xs:int" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>

```

GetJobStatusResponse

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="GetJobStatusResponse" nillable="true" type="GetJobStatusResponse" />
  <xs:complexType name="GetJobStatusResponse">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="JobStatusResponseMessageType"
        type="GetJobStatusResponseMessageType" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="GetJobStatusResponseMessageType">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="Header" type="HeaderType" />
      <xs:element minOccurs="0" maxOccurs="1" name="Reply" type="ReplyType" />
      <xs:element minOccurs="0" maxOccurs="1" name="PayLoad"
        type="JobStatusPayLoadType" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="HeaderType">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="Verb" type="xs:string" />
    </xs:sequence>
  </xs:complexType>

```

```

<xs:element minOccurs="0" maxOccurs="1" name="Noun" type="xs:string" />
<xs:element minOccurs="1" maxOccurs="1" name="Timestamp" type="xs:dateTime" />
<xs:element minOccurs="0" maxOccurs="1" name="Source" type="xs:string" />
</xs:sequence>
</xs:complexType>
<xs:complexType name="ReplyType">
<xs:sequence>
<xs:element minOccurs="0" maxOccurs="1" name="Result" type="xs:string" />
<xs:element minOccurs="0" maxOccurs="1" name="Error" type="ErrorType" />
</xs:sequence>
</xs:complexType>
<xs:complexType name="ErrorType">
<xs:sequence>
<xs:element minOccurs="0" maxOccurs="1" name="code" type="xs:string" />
<xs:element minOccurs="0" maxOccurs="1" name="reason" type="xs:string" />
</xs:sequence>
</xs:complexType>
<xs:complexType name="JobStatusPayLoadType">
<xs:sequence>
<xs:element minOccurs="1" maxOccurs="1" name="JobStatus" type="JobStatusType"
/>
</xs:sequence>
</xs:complexType>
<xs:simpleType name="JobStatusType">
<xs:restriction base="xs:string">
<xs:enumeration value="Queued" />
<xs:enumeration value="VerifyingInputData" />
<xs:enumeration value="MovingToProcessing" />
<xs:enumeration value="Converting" />
<xs:enumeration value="Publishing" />
<xs:enumeration value="Timeout" />
<xs:enumeration value="SSGError" />
<xs:enumeration value="SchemaValidationFailed" />
<xs:enumeration value="ConverterFatais" />
<xs:enumeration value="ConvertingError" />
<xs:enumeration value="ErrorCopyInputToProcessing" />
<xs:enumeration value="PublishingError" />
<xs:enumeration value="UnableToRestore" />
<xs:enumeration value="Uploaded" />
<xs:enumeration value="Rejected" />
<xs:enumeration value="ReadyForReview" />
<xs:enumeration value="UnderReview" />
<xs:enumeration value="ModelNeedsReview" />
<xs:enumeration value="Removed" />
</xs:restriction>
</xs:simpleType>
</xs:schema>

```

GetJobStatusHeaderType

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="GetJobStatusHeaderType" nullable="true"
type="GetJobStatusHeaderType" />

```

```

<xs:complexType name="GetJobStatusHeaderType">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Verb" type="xs:string" />
    <xs:element minOccurs="0" maxOccurs="1" name="Noun" type="xs:string" />
    <xs:element minOccurs="1" maxOccurs="1" name="Timestamp" type="xs:dateTime" />
    <xs:element minOccurs="0" maxOccurs="1" name="Source" type="xs:string" />
    <xs:element minOccurs="0" maxOccurs="1" name="Station" type="ArrayOfStationStc" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ArrayOfStationStc">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="StationStc" nillable="true" type="StationStc" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="StationStc">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="ID" type="xs:string" />
    <xs:element minOccurs="0" maxOccurs="1" name="Name" type="xs:string" />
  </xs:sequence>
</xs:complexType>
</xs:schema>

```

GetJobStatusRequestType

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="GetJobStatusRequestType" nillable="true" type="GetJobStatusRequestType" />
  <xs:complexType name="GetJobStatusRequestType">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="JobID" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="DataVersion" type="xs:string" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>

```

GetJobStatusResponseMessageType

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="GetJobStatusResponseMessageType" nillable="true" type="GetJobStatusResponseMessageType" />
  <xs:complexType name="GetJobStatusResponseMessageType">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="Header" type="HeaderType" />
      <xs:element minOccurs="0" maxOccurs="1" name="Reply" type="ReplyType" />
      <xs:element minOccurs="0" maxOccurs="1" name="PayLoad" type="JobStatusPayLoadType" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="HeaderType">

```

```

<xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Verb" type="xs:string" />
    <xs:element minOccurs="0" maxOccurs="1" name="Noun" type="xs:string" />
    <xs:element minOccurs="1" maxOccurs="1" name="Timestamp" type="xs:dateTime" />
    <xs:element minOccurs="0" maxOccurs="1" name="Source" type="xs:string" />
</xs:sequence>
</xs:complexType>
<xs:complexType name="ReplyType">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="Result" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="Error" type="ErrorType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="ErrorType">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="code" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="reason" type="xs:string" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="JobStatusPayLoadType">
    <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="1" name="JobStatus" type="JobStatusType"
/>
    </xs:sequence>
</xs:complexType>
<xs:simpleType name="JobStatusType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="Queued" />
        <xs:enumeration value="VerifyingInputData" />
        <xs:enumeration value="MovingToProcessing" />
        <xs:enumeration value="Converting" />
        <xs:enumeration value="Publishing" />
        <xs:enumeration value="Timeout" />
        <xs:enumeration value="SSGError" />
        <xs:enumeration value="SchemaValidationFailed" />
        <xs:enumeration value="ConverterFatais" />
        <xs:enumeration value="ConvertingError" />
        <xs:enumeration value="ErrorCopyInputToProcessing" />
        <xs:enumeration value="PublishingError" />
        <xs:enumeration value="UnableToRestore" />
        <xs:enumeration value="Uploaded" />
        <xs:enumeration value="Rejected" />
        <xs:enumeration value="ReadyForReview" />
        <xs:enumeration value="UnderReview" />
        <xs:enumeration value="ModelNeedsReview" />
        <xs:enumeration value="Removed" />
    </xs:restriction>
</xs:simpleType>
</xs:schema>

```

StationStc

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
```

```

<xs:element name="StationStc" nillable="true" type="StationStc" />
<xs:complexType name="StationStc">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="ID" type="xs:string" />
    <xs:element minOccurs="0" maxOccurs="1" name="Name" type="xs:string" />
  </xs:sequence>
</xs:complexType>
</xs:schema>

```

HeaderType

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="HeaderType" nillable="true" type="HeaderType" />
  <xs:complexType name="HeaderType">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="Verb" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="Noun" type="xs:string" />
      <xs:element minOccurs="1" maxOccurs="1" name="Timestamp" type="xs:dateTime" />
      <xs:element minOccurs="0" maxOccurs="1" name="Source" type="xs:string" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>

```

ReplyType

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="ReplyType" nillable="true" type="ReplyType" />
  <xs:complexType name="ReplyType">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="Result" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="Error" type="ErrorType" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ErrorType">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="code" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="reason" type="xs:string" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>

```

JobStatusPayLoadType

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="JobStatusPayLoadType" nillable="true" type="JobStatusPayLoadType" />
  <xs:complexType name="JobStatusPayLoadType">
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="JobStatus" type="JobStatusType" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>

```

```

        </xs:sequence>
    </xs:complexType>
    <xs:simpleType name="JobStatusType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="Queued" />
            <xs:enumeration value="VerifyingInputData" />
            <xs:enumeration value="MovingToProcessing" />
            <xs:enumeration value="Converting" />
            <xs:enumeration value="Publishing" />
            <xs:enumeration value="Timeout" />
            <xs:enumeration value="SSGError" />
            <xs:enumeration value="SchemaValidationFailed" />
            <xs:enumeration value="ConverterFatal" />
            <xs:enumeration value="ConvertingError" />
            <xs:enumeration value="ErrorCopyInputToProcessing" />
            <xs:enumeration value="PublishingError" />
            <xs:enumeration value="UnableToRestore" />
            <xs:enumeration value="Uploaded" />
            <xs:enumeration value="Rejected" />
            <xs:enumeration value="ReadyForReview" />
            <xs:enumeration value="UnderReview" />
            <xs:enumeration value="ModelNeedsReview" />
            <xs:enumeration value="Removed" />
        </xs:restriction>
    </xs:simpleType>
</xs:schema>
```

ErrorType

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="ErrorType" nillable="true" type="ErrorType" />
    <xs:complexType name="ErrorType">
        <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="code" type="xs:string" />
            <xs:element minOccurs="0" maxOccurs="1" name="reason" type="xs:string" />
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```

eJobState

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="eJobState" type="eJobState" />
    <xs:simpleType name="eJobState">
        <xs:restriction base="xs:string">
            <xs:enumeration value="None" />
            <xs:enumeration value="InitialLoad" />
            <xs:enumeration value="ReExport" />
            <xs:enumeration value="AsBuilt" />
            <xs:enumeration value="Proposed" />
        </xs:restriction>
    </xs:simpleType>
</xs:schema>
```

```

</xs:simpleType>
</xs:schema>

SSGHistoryStatusPublishType

<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="SSGHistoryStatusPublishType" type="SSGHistoryStatusPublishType">
/>
  <xs:simpleType name="SSGHistoryStatusPublishType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Publishing" />
      <xs:enumeration value="Converting" />
      <xs:enumeration value="Queued" />
      <xs:enumeration value="ReadyForReview" />
      <xs:enumeration value="ConverterFatal" />
      <xs:enumeration value="ErrorCopyInputToProcessing" />
      <xs:enumeration value="PublishingError" />
      <xs:enumeration value="Uploaded" />
      <xs:enumeration value="Timeout" />
      <xs:enumeration value="Rejected" />
      <xs:enumeration value="SSGError" />
      <xs:enumeration value="UnableToRestore" />
      <xs:enumeration value="ConvertingError" />
      <xs:enumeration value="SchemaValidationFailed" />
      <xs:enumeration value="Removed" />
      <xs:enumeration value="UnderReview" />
      <xs:enumeration value="ModelNeedsReview" />
    </xs:restriction>
  </xs:simpleType>
</xs:schema>

```

eSynchronousResponse

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="eSynchronousResponse" type="eSynchronousResponse" />
  <xs:simpleType name="eSynchronousResponse">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Good" />
      <xs:enumeration value="Bad" />
      <xs:enumeration value="NoResponse" />
      <xs:enumeration value="None" />
    </xs:restriction>
  </xs:simpleType>
</xs:schema>

```

JobStatusSoapMessage

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="JobStatusSoapMessage" nillable="true" type="JobStatusSoapMessage" />

```

```

<xs:complexType name="JobStatusSoapMessage">
  <xs:complexContent mixed="false">
    <xs:extension base="ModelManagerDataObject">
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="JobId" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="JobStatusType"
type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="JobStatusComment"
type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="MessageId" type="xs:string"
/>
        <xs:element minOccurs="0" maxOccurs="1" name="SubstnJobStatusHistory"
type="ArrayOfSubstationsJobStatusHistory" />
        <xs:element minOccurs="1" maxOccurs="1"
name="JobStatusSoapMessageSyncResponse" type="eSynchronousResponse" />
        <xs:element minOccurs="1" maxOccurs="1" name="RequestSoapMessageType"
type="eSoapMessageType" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="ModelManagerDataObject" abstract="true" />
<xs:complexType name="SubstationsJobStatusHistory">
  <xs:complexContent mixed="false">
    <xs:extension base="ModelManagerDataObject">
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="SubStationId"
type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="SubStationStatusComment"
type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="SubStationStatus"
type="xs:string" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="ArrayOfSubstationsJobStatusHistory">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded"
name="SubstationsJobStatusHistory" nillable="true" type="SubstationsJobStatusHistory"
/>
  </xs:sequence>
</xs:complexType>
<xs:simpleType name="eSynchronousResponse">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Good" />
    <xs:enumeration value="Bad" />
    <xs:enumeration value="NoResponse" />
    <xs:enumeration value="None" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="eSoapMessageType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="None" />
    <xs:enumeration value="ConsumerRequest" />
  </xs:restriction>
</xs:simpleType>

```

```

    <xs:enumeration value="GetJobStatusRequest" />
  </xs:restriction>
</xs:simpleType>
</xs:schema>

```

NetworkDataSetResyncMessage

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="NetworkDataSetResyncMessage" nillable="true"
    type="NetworkDataSetResyncMessage" />
  <xs:complexType name="NetworkDataSetResyncMessage">
    <xs:complexContent mixed="false">
      <xs:extension base="ModelManagerDataObject">
        <xs:sequence>
          <xs:element minOccurs="0" maxOccurs="1" name="MessageId" type="xs:string"
        />
          <xs:element minOccurs="1" maxOccurs="1" name="TimeStamp" nillable="true"
            type="xs:dateTime" />
            <xs:element minOccurs="0" maxOccurs="1" name="Source" type="xs:string" />
            <xs:element minOccurs="1" maxOccurs="1" name="AsyncReplyFlag"
              type="xs:boolean" />
              <xs:element minOccurs="0" maxOccurs="1" name="UserId" type="xs:string" />
              <xs:element minOccurs="1" maxOccurs="1"
                name="NetworkDataSetResyncSoapMessageSyncResponse" type="eSynchronousResponse" />
                <xs:element minOccurs="0" maxOccurs="1" name="NetworkDataSetResyncStations"
                  type="ArrayOfNetworkDataSetResyncStation" />
                  </xs:sequence>
                  </xs:extension>
                  </xs:complexContent>
                </xs:complexType>
                <xs:complexType name="ModelManagerDataObject" abstract="true" />
                <xs:complexType name="NetworkDataSetResyncStation">
                  <xs:complexContent mixed="false">
                    <xs:extension base="ModelManagerDataObject">
                      <xs:sequence>
                        <xs:element minOccurs="0" maxOccurs="1" name="Id" type="xs:string" />
                        <xs:element minOccurs="0" maxOccurs="1" name="mRID" type="xs:string" />
                        <xs:element minOccurs="0" maxOccurs="1" name="SubstationName"
                          type="xs:string" />
                        </xs:sequence>
                        </xs:extension>
                      </xs:complexContent>
                    </xs:complexType>
                    <xs:simpleType name="eSynchronousResponse">
                      <xs:restriction base="xs:string">
                        <xs:enumeration value="Good" />
                        <xs:enumeration value="Bad" />
                        <xs:enumeration value="NoResponse" />
                        <xs:enumeration value="None" />
                      </xs:restriction>
                    </xs:simpleType>
                    <xs:complexType name="ArrayOfNetworkDataSetResyncStation">
                      <xs:sequence>

```

```

        <xs:element minOccurs="0" maxOccurs="unbounded"
name="NetworkDataSetResyncStation" nillable="true" type="NetworkDataSetResyncStation"
/>
    </xs:sequence>
</xs:complexType>
</xs:schema>

```

NetworkDataSetResyncStation

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="NetworkDataSetResyncStation" nillable="true"
type="NetworkDataSetResyncStation" />
    <xs:complexType name="NetworkDataSetResyncStation">
        <xs:complexContent mixed="false">
            <xs:extension base="ModelManagerDataObject">
                <xs:sequence>
                    <xs:element minOccurs="0" maxOccurs="1" name="Id" type="xs:string" />
                    <xs:element minOccurs="0" maxOccurs="1" name="mRID" type="xs:string" />
                    <xs:element minOccurs="0" maxOccurs="1" name="SubstationName"
type="xs:string" />
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="ModelManagerDataObject" abstract="true" />
</xs:schema>

```

SubstationsJobStatusHistory

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="SubstationsJobStatusHistory" nillable="true"
type="SubstationsJobStatusHistory" />
    <xs:complexType name="SubstationsJobStatusHistory">
        <xs:complexContent mixed="false">
            <xs:extension base="ModelManagerDataObject">
                <xs:sequence>
                    <xs:element minOccurs="0" maxOccurs="1" name="SubStationId"
type="xs:string" />
                    <xs:element minOccurs="0" maxOccurs="1" name="SubStationStatusComment"
type="xs:string" />
                    <xs:element minOccurs="0" maxOccurs="1" name="SubStationStatus"
type="xs:string" />
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="ModelManagerDataObject" abstract="true" />
</xs:schema>

```

SubstationGISAInstanceDetails

```
<?xml version="1.0" encoding="utf-8"?>
```

```

<xs:schema elementFormDefault="qualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="SubstationGISAInstanceDetails" nillable="true"
    type="SubstationGISAInstanceDetails" />
  <xs:complexType name="SubstationGISAInstanceDetails">
    <xs:complexContent mixed="false">
      <xs:extension base="ModelManagerDataObject">
        <xs:sequence>
          <xs:element minOccurs="0" maxOccurs="1" name="SubStationId"
            type="xs:string" />
          <xs:element minOccurs="0" maxOccurs="1" name="SubStationName"
            type="xs:string" />
          <xs:element minOccurs="0" maxOccurs="1" name="GISAInstanceName"
            type="xs:string" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="ModelManagerDataObject" abstract="true" />
</xs:schema>

```

ModelingRequest

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="ModelingRequest" nillable="true" type="ModelingRequest" />
  <xs:complexType name="ModelingRequest">
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="Type" type="ModelingRequestType"
    />
      <xs:element minOccurs="0" maxOccurs="1" name="Station" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="SsgFile" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="Id" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="User" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="Reason" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="StationObj" type="Substation" />
      <xs:element minOccurs="0" maxOccurs="1" name="Stations"
        type="ArrayOfSubstation" />
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="ModelingRequestType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="BlockStation" />
      <xs:enumeration value="UnBlockStation" />
      <xs:enumeration value="DeleteSsg" />
      <xs:enumeration value="ReviewApprove" />
      <xs:enumeration value="ReviewReject" />
      <xs:enumeration value="UploadWithFatais" />
      <xs:enumeration value="BuildNavOverview" />
      <xs:enumeration value="ReprocessStation" />
      <xs:enumeration value="ReprocessStationGroup" />
      <xs:enumeration value="BlockStationGroup" />
      <xs:enumeration value="UnBlockStationGroup" />
      <xs:enumeration value="UploadWithoutFatais" />
    </xs:restriction>
  </xs:simpleType>
</xs:schema>

```

```

<xs:enumeration value="UnderReview" />
<xs:enumeration value="ReviewCompleted" />
<xs:enumeration value="RequestResyncStation" />
<xs:enumeration value="RequestResyncStationGroup" />
</xs:restriction>
</xs:simpleType>
<xs:complexType name="Substation">
  <xs:complexContent mixed="false">
    <xs:extension base="ModelManagerDataObject">
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="Name" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="ID" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="AreaName" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="Description" type="xs:string" />
      </xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="ChangedItems" type="ArrayOfSubstationItem" />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="ModelManagerDataObject" abstract="true" />
<xs:complexType name="SubstationItem">
  <xs:complexContent mixed="false">
    <xs:extension base="ModelManagerDataObject">
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="DeviceID" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="AttributeName" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="WorkOrderNumber" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="TypeChange" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="Description" type="xs:string" />
        <xs:element minOccurs="1" maxOccurs="1" name="DateTime" type="xs:dateTime" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="ArrayOfSubstationItem">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="SubstationItem" nillable="true" type="SubstationItem" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ArrayOfSubstation">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="Substation" nillable="true" type="Substation" />
  </xs:sequence>
</xs:complexType>
</xs:schema>

```

DevicesStationDetailsFile

```
<?xml version="1.0" encoding="utf-8"?>
<xss:schema elementFormDefault="qualified"
  xmlns:xss="http://www.w3.org/2001/XMLSchema">
  <xss:element name="DevicesStationDetailsFile" nillable="true"
    type="DevicesStationDetailsFile" />
  <xss:complexType name="DevicesStationDetailsFile">
    <xss:complexContent mixed="false">
      <xss:extension base="ModelManagerDataObject">
        <xss:sequence>
          <xss:element minOccurs="0" maxOccurs="1" name="DevicesStationDetails"
            type="ArrayOfDeviceStationDetails" />
        </xss:sequence>
      </xss:extension>
    </xss:complexContent>
  </xss:complexType>
  <xss:complexType name="ModelManagerDataObject" abstract="true" />
  <xss:complexType name="DeviceStationDetails">
    <xss:complexContent mixed="false">
      <xss:extension base="ModelManagerDataObject">
        <xss:sequence>
          <xss:element minOccurs="0" maxOccurs="1" name="DeviceId" type="xs:string" />
          <xss:element minOccurs="1" maxOccurs="1" name="DeviceType"
            type="eDeviceType" />
          <xss:element minOccurs="0" maxOccurs="1" name="StationName" type="xs:string"
            />
        </xss:sequence>
      </xss:extension>
    </xss:complexContent>
  </xss:complexType>
  <xss:simpleType name="eDeviceType">
    <xss:restriction base="xs:string">
      <xss:enumeration value="None" />
      <xss:enumeration value="Protection" />
      <xss:enumeration value="Regulation" />
    </xss:restriction>
  </xss:simpleType>
  <xss:complexType name="ArrayOfDeviceStationDetails">
    <xss:sequence>
      <xss:element minOccurs="0" maxOccurs="unbounded" name="DeviceStationDetails"
        nillable="true" type="DeviceStationDetails" />
    </xss:sequence>
  </xss:complexType>
</xss:schema>
```

NetworkDataSetUpdate_OutBound.wsdl

```
<wsdl:definitions xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:tns="ENMAC/NetworkDataSetUpdate" xmlns:s="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
```

```

targetNamespace="ENMAC/NetworkDataSetUpdate"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <s:schema elementFormDefault="qualified">
      targetNamespace="ENMAC/NetworkDataSetUpdate">
        <s:element name="JobStatus" type="tns:JobStatus" />
        <s:complexType name="JobStatus">
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="MessageHeader"
type="tns:MessageHeaderStc" />
            <s:element minOccurs="1" maxOccurs="1" name="MessagePayload"
type="tns:MessagePayloadStc" />
          </s:sequence>
        </s:complexType>
        <s:complexType name="MessageHeaderStc">
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="DataVersion" type="s:string"
/>
            <s:element minOccurs="1" maxOccurs="1" name="JobID" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="JobStatus"
type="tns:JobStatusType" />
            <s:element minOccurs="0" maxOccurs="1" name="JobStatusComment"
type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="MessageID" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="Noun" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="Source" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="Timestamp" type="s:dateTime"
/>
          </s:sequence>
        </s:complexType>
        <s:complexType name="MessagePayloadStc">
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="unbounded" name="ProcessedCircuit"
type="tns:ProcessedCircuitStc" />
          </s:sequence>
        </s:complexType>
        <s:complexType name="ProcessedCircuitStc">
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="CircuitID" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="JobStatus"
type="tns:JobStatusType" />
            <s:element minOccurs="0" maxOccurs="1" name="JobStatusComment"
type="s:string" />
          </s:sequence>
        </s:complexType>
        <s:simpleType name="JobStatusType">
          <s:restriction base="s:string">
            <s:enumeration value="Queued" />
            <s:enumeration value="Creating" />
            <s:enumeration value="RetryCreate" />
            <s:enumeration value="Created" />
            <s:enumeration value="Ready" />
            <s:enumeration value="Introducing" />
            <s:enumeration value="RetryIntroduce" />
          </s:restriction>
        </s:simpleType>
      </s:complexType>
    </s:schema>
  </wsdl:types>

```

```

<s:enumeration value="TempIntroduced" />
<s:enumeration value="Removed" />
<s:enumeration value="Accepted" />
<s:enumeration value="Rejected" />
</s:restriction>
</s:simpleType>
<s:element name="JobStatusResponse" type="tns:JobStatusResponse" />
<s:complexType name="JobStatusResponse">
<s:sequence>
<s:element minOccurs="1" maxOccurs="1" name="Status" type="s:int" />
</s:sequence>
</s:complexType>
<s:element name="GetNetworkDataSetResync">
<s:complexType>
<s:sequence>
<s:element minOccurs="1" maxOccurs="1" name="Header"
type="tns:HeaderType" />
<s:element minOccurs="1" maxOccurs="1" name="Request"
type="tns:GetNetworkDataSetResyncRequestType" />
</s:sequence>
</s:complexType>
</s:element>
<s:complexType name="HeaderType">
<s:sequence>
<s:element minOccurs="1" maxOccurs="1" name="Verb" type="s:string" />
<s:element minOccurs="1" maxOccurs="1" name="Noun" type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="Timestamp" nillable="true"
type="s:dateTime" />
<s:element minOccurs="0" maxOccurs="1" name="Source" nillable="true"
type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="AsyncReplyFlag"
type="s:boolean" />
<s:element minOccurs="0" maxOccurs="1" name="MessageID" type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="CorrelationID" type="s:string"
/>
</s:sequence>
</s:complexType>
<s:complexType name="GetNetworkDataSetResyncRequestType">
<s:sequence>
<s:element minOccurs="1" maxOccurs="unbounded" name="Circuit"
type="tns:CircuitStc" />
<s:element minOccurs="0" maxOccurs="1" name="User" type="tns:UserType" />
<s:element minOccurs="0" maxOccurs="1" name="Description" type="s:string"
/>
</s:sequence>
</s:complexType>
<s:complexType name="CircuitStc">
<s:sequence>
<s:element minOccurs="1" maxOccurs="1" name="mRID" type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="ID" nillable="true"
type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="Name" nillable="true"
type="s:string" />
</s:sequence>
</s:complexType>

```

```

<s:complexType name="UserType">
  <s:sequence>
    <s:element minOccurs="1" maxOccurs="1" name="UserID" type="s:string" />
  </s:sequence>
</s:complexType>
<s:element name="GetNetworkDataSetResyncResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1"
name="GetNetworkDataSetResyncResponseType" type="s:int" />
    </s:sequence>
  </s:complexType>
</s:element>
<s:complexType name="GetNetworkDataSetResyncResponseType">
  <s:sequence>
    <s:element minOccurs="1" maxOccurs="1" name="Header" type="tns:HeaderType" />
    <s:element minOccurs="1" maxOccurs="1" name="Reply" type="tns:ReplyType" />
  </s:sequence>
</s:complexType>
<s:complexType name="ReplyType">
  <s:sequence>
    <s:element minOccurs="1" maxOccurs="1" name="Result" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="Error" type="tns:ErrorType" />
  </s:sequence>
</s:complexType>
<s:complexType name="ErrorType">
  <s:sequence>
    <s:element minOccurs="1" maxOccurs="1" name="code" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="reason" type="s:string" />
  </s:sequence>
</s:complexType>
</s:schema>
</wsdl:types>
<wsdl:message name="JobStatusSoapIn">
  <wsdl:part name="request" element="tns:JobStatus" />
</wsdl:message>
<wsdl:message name="JobStatusSoapOut">
  <wsdl:part name="JobStatusResult" element="tns:JobStatusResponse" />
</wsdl:message>
<wsdl:message name="GetNetworkDataSetResyncSoapIn">
  <wsdl:part name="parameters" element="tns:GetNetworkDataSetResync" />
</wsdl:message>
<wsdl:message name="GetNetworkDataSetResyncSoapOut">
  <wsdl:part name="parameters" element="tns:GetNetworkDataSetResyncResponse" />
</wsdl:message>
<wsdl:portType name="NetworkDataSetUpdateSoap">
  <wsdl:operation name="JobStatus">
    <wsdl:input message="tns:JobStatusSoapIn" />
    <wsdl:output message="tns:JobStatusSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="GetNetworkDataSetResync">
    <wsdl:input message="tns:GetNetworkDataSetResyncSoapIn" />
    <wsdl:output message="tns:GetNetworkDataSetResyncSoapOut" />
  </wsdl:operation>
</wsdl:portType>

```

```

</wsdl:portType>
<wsdl:binding name="NetworkDataSetUpdateSoap" type="tns:NetworkDataSetUpdateSoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="JobStatus">
        <soap:operation soapAction="ENMAC/NetworkDataSetUpdate/JobStatus"
style="document" />
        <wsdl:input>
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetNetworkDataSetResync">
        <soap:operation soapAction="ENMAC/NetworkDataSetUpdate/GetNetworkDataSetResync"
style="document" />
        <wsdl:input>
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
</wsdl:binding>
<wsdl:service name="NetworkDataSetUpdate">
    <wsdl:port name="NetworkDataSetUpdateSoap"
binding="tns:NetworkDataSetUpdateSoap">
        <soap:address
location="http://localhost/SOAPOutBound/NetworkDataSetUpdate.asmx" />
    </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

Appendix C. Model Manager REST Web Service

REST messages are exchanged between the Model Manager Application (MMA) and External Systems over HTTPS. External systems could be any GIS systems, planning application, or other application within the utility.

To enable the Model Manager REST web service, select the Enable RESTful Interface Server option on the RESTful Interface Tab of the Model Manager Configuration Editor.

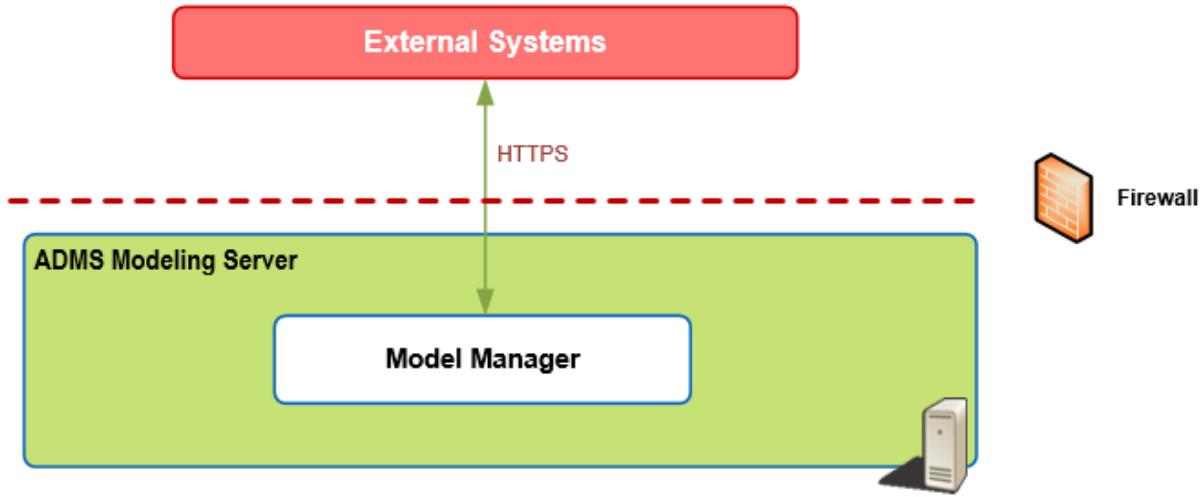


Figure 38. Model Manager - External Systems Data Flow

The Model Manager REST web service is implemented using RESTful Web API. For details about Representational State Transfer (REST), refer to <https://restfulapi.net/>.

To secure the REST message exchange, HTTP Transport Layer Security (TLS) with mutual certificate authentication is used.

C.1 Service Messages

The Model Manager REST web interface host and port values are configured on the RESTful Interface Tab of the Model Manager Configuration Editor.

Each Model Manager REST web interface URL has the following format:

<Host:Port><Root><Endpoint>

For example:

`https://localhost:9000/mmarest/v311/job-status/jobs`

The Model Manager REST web interface endpoints are:

- **/mmarest/v311/job-status/**: This is the root of the JobStatus web service.
 - **jobs**: External system uses this endpoint to request the job processing status from the Model Manager. The request body contains the JobIDList object, which lists job IDs. The Model

Manager takes list of job IDs and searches for the corresponding recent SSG history file and returns the status of the job. (POST)

Example URL:

<https://localhost:9000/mmarest/v311/job-status/jobs>

Example message body:

```
{"JobIds": ["570", "571", "572"]}
```

Example Response:

```
{
  "570": {
    "UserComments": [],
    "Job_Status": "Uploaded",
    "JobStatusComment": "Uploaded",
    "TimeStamp": "2020-09-09T19:04:45.8567806+05:30",
    "Source": "ADMS Model Manager",
    "Noun": "JobStatus",
    "Error": null
  },
  "571": {
    "UserComments": null,
    "Job_Status": null,
    "JobStatusComment": null,
    "TimeStamp": "0001-01-01T00:00:00",
    "Source": null,
    "Noun": null,
    "Error": "The Job with ID 571 is not found in Model Manager History"
  },
  "572": {
    "UserComments": [],
    "Job_Status": "ConverterFatal",
    "JobStatusComment": "ConverterFatal",
    "TimeStamp": "2020-08-24T10:48:51.9653707+05:30",
    "Source": "ADMS Model Manager",
    "Noun": "JobStatus",
    "Error": null
  }
}
```

- **station-status:** External systems use this endpoint to request the station status from the Model Manager. The request body contains the SubStationList, which lists the station names. The Model Manager takes the list of station names, checks the corresponding recent SSG history file, and then returns the status for the stations. (POST)

Example URL:

<https://localhost:9000/mmarest/v311/job-status/station-status>

Example message body:

```
{"StationNames": ["3LIONS", "COHO", "VENUS"]}
```

Example Response:

```
{
  "3LIONS": {
```

```

        "UserComments": [],
        "StationStatus": "Uploaded",
        "JobStatus": "Uploaded",
        "JobStatusComment": "Uploaded",
        "TimeStamp": "2020-10-09T20:11:49.4177275+05:30",
        "Source": "ADMS Model Manager",
        "Noun": "Station Status",
        "Error": null
    },
    "COHO": {
        "UserComments": null,
        "StationStatus": null,
        "JobStatus": null,
        "JobStatusComment": null,
        "TimeStamp": "0001-01-01T00:00:00",
        "Source": null,
        "Noun": null,
        "Error": "The Station with Name COHO is not found in Model Manager
History"
    },
    "VENUS": {
        "UserComments": [],
        "StationStatus": "Uploaded",
        "JobStatus": "Uploaded",
        "JobStatusComment": "Uploaded",
        "TimeStamp": "2020-10-09T20:11:50.3332625+05:30",
        "Source": "ADMS Model Manager",
        "Noun": "Station Status",
        "Error": null
    }
}

```

- **station-status/jobs:** External systems use this endpoint to request the station status in a job from the Model Manager. The request body contains the list of StationNameToJobID, which includes the station name and job ID. The Model Manager takes the list of station IDs and job IDs and returns the corresponding job status information. (POST)

Example URL:

<https://localhost:9000/mmarest/v311/job-status/station-status/jobs>

Example message body:

```
[{"StationName": "3LIONS", "JobID": "553"},  
 {"StationName": "BACHELOR", "JobID": "551"},  
 {"StationName": "VENUS", "JobID": "552"}]
```

Example Response:

```
{
    "3LIONS": {
        "UserComments": [],
        "StationStatus": "Uploaded",
        "JobStatus": "Uploaded",
        "JobStatusComment": "Uploaded",
        "TimeStamp": "2019-04-16T01:53:19.8265448+05:30",
        "Source": "ADMS Model Manager",
        "Noun": "Station Status",
    }
}
```

```

        "Error": null
    },
    "BACHELOR": {
        "UserComments": null,
        "StationStatus": null,
        "JobStatus": null,
        "JobStatusComment": null,
        "TimeStamp": "0001-01-01T00:00:00",
        "Source": null,
        "Noun": null,
        "Error": "Unable to find status for Station with Name BACHELOR in Job
with ID 551 in Model Manager History"
    },
    "VENUS": {
        "UserComments": [],
        "StationStatus": "Uploaded",
        "JobStatus": "Uploaded",
        "JobStatusComment": "Uploaded",
        "TimeStamp": "2019-04-16T01:12:50.9530262+05:30",
        "Source": "ADMS Model Manager",
        "Noun": "Station Status",
        "Error": null
    }
}

```

- **/mmarest/v311/model-request/**: This is the root of the ModelRequest web service.
 - **full-station**: When an external system is capable of publishing station file(s), the external system uses this endpoint to notify the Model Manager application about the publishing. After receiving this notification, the Model Manager application transfers the files via SFTP. The request body contains SubStationInfo, which specifies the SFTP URL. The username and private key used in SFTP authentication are configured on the RESTful Interface Tab of the Model Manager Configuration Editor. (POST)

Example URL:
<https://localhost:9000/mmarest/v311/model-request/full-station>

Example message body:

```
{"SFTP_URL":"http://g5cg73253nke:80/Model_Exchange_Folder/3LIONS.zip"}
```
- **/mmarest/v311/model-information/**: This is the root of the StationData web service.
 - **externals/zip**: External systems use this endpoint to request a station externals file from the Model Manager's "Previous" folder. The request body is SubStationList, which lists the station names. The Model Manager sends the file in a zip package. (POST)

Example URL:
<https://localhost:9000/mmarest/v311/model-information/externals/zip>

Example message body:

```
{"StationNames":["3LIONS", "COHO", "VENUS"]}
```
 - **internals/zip**: External systems use this endpoint to request a station internals file from the Model Manager's "Previous" folder. The request body is SubStationList, which lists the station names. The Model Manager sends the file in a zip package. (POST)

Example URL:

<https://localhost:9000/mmarest/v311/model-information/internals/zip>

Example message body:

```
{"StationNames": ["3LIONS", "COHO", "VENUS"]}
```

- **externals/internals/zip:** External systems use this endpoint to request station externals and station internals files from the Model Manager's "Previous" folder. The request body is SubStationList, which lists the station names. The Model Manager sends the files in a zip package. (POST)

Example URL:

<https://localhost:9000/mmarest/v311/model-information/externals/internals/zip>

Example message body:

```
{"StationNames": ["3LIONS", "COHO", "VENUS"]}
```

- **externals/{subStationFileName}:** External systems use this endpoint to request a station externals file from the Model Manager's "Previous" folder. The Model Manager sends the file data in the XML response. (GET)

Example URL:

<https://localhost:9000/mmarest/v311/model-information/externals/3LIONS>

- **internals/{subStationFileName}:** External systems use this endpoint to request a station internals file from the Model Manager's "Previous" folder. The Model Manager sends the file data in the XML response. (GET)

Example URL:

<https://localhost:9000/mmarest/v311/model-information/internals/3LIONS>

- **Scada Online Model Update Service Messages:** The following endpoints support the Scada Online Model Update functionality.

- **mmarest/api/v1/status/SCADA:** MMA uses this endpoint to process polled Scada Online Model Update ping messages (OnlineScadaModelUpdatePingMessage) and publish change documents (ScadaOnlineModelUpdateDetails.xml) to Scada for processing (as a response, if available). (PUT)

MMA also receives the information about deployments that have been processed by SCADA. The response message is OnlineScadaModelUpdateReplyMessage.

- **mmarest/api/v1/files/SCADA/deployments/{pendingdeploymentmrid}:** External Systems use this endpoint to request MMA to send available pending change documents from the Change Doc folder to SCADA (as a response, if available). (GET)
- **mmarest/api/v1/rollback/SCADA/deployments/{pendingdeploymentmrid}:** MMA uses this endpoint to send information about the roll back reference of deployments. (POST)

C.2 Message Schemas for Scada Online Model Update

OnlineScadaModelUpdatePingMessage

```
<?xml version="1.0" encoding="utf-8"?>
```

```

<xs:schema elementFormDefault="qualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="OnlineScadaModelUpdatePingMessage" nillable="true"
    type="OnlineScadaModelUpdatePingMessage" />
  <xs:complexType name="OnlineScadaModelUpdatePingMessage">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="targetCacheScopeId"
        type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="lastDeployment" type="xs:string"
        />
      <xs:element minOccurs="1" maxOccurs="1" name="isAvailableForDeployment"
        type="xs:boolean" />
      <xs:element minOccurs="1" maxOccurs="1" name="isRollbackAvailable"
        type="xs:boolean" />
      <xs:element minOccurs="0" maxOccurs="1" name="rollbackAvailabilityMessage"
        type="xs:string" />
      <xs:element minOccurs="1" maxOccurs="1" name="pastDeploymentsIncluded"
        type="xs:int" />
      <xs:element minOccurs="0" maxOccurs="1" name="loadedModelAuthoritySets"
        type="ArrayOfLoadedModelAuthoritySet" />
      <xs:element minOccurs="0" maxOccurs="1" name="pastDeployments"
        type="ArrayOfPastDeployment" />
      <xs:element minOccurs="0" maxOccurs="1" name="pendingDeployment"
        type="PendingDeployment" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ArrayOfLoadedModelAuthoritySet">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="LoadedModelAuthoritySet"
        nillable="true" type="LoadedModelAuthoritySet" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="LoadedModelAuthoritySet">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="name" type="xs:string" />
      <xs:element minOccurs="1" maxOccurs="1" name="version" type="xs:int" />
      <xs:element minOccurs="0" maxOccurs="1" name="mrid" type="xs:string" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ArrayOfPastDeployment">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="PastDeployment"
        nillable="true" type="PastDeployment" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="PastDeployment">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="mrid" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="rollbackReference"
        type="xs:string" />
      <xs:element minOccurs="1" maxOccurs="1" name="processedDate" type="xs:dateTime"
        />
      <xs:element minOccurs="0" maxOccurs="1" name="messages" type="ArrayOfMessage"
        />
    </xs:sequence>
  </xs:complexType>
</xs:schema>

```

```

        </xs:sequence>
    </xs:complexType>
<xs:complexType name="ArrayOfMessage">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="Message" nillable="true"
type="Message" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="Message">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="severity" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="text" type="xs:string" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="PendingDeployment">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="mrid" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="user" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="changeDocURL" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="expectedScopeId" type="xs:string"
/>
        <xs:element minOccurs="0" maxOccurs="1" name="expectedOriginDeployment"
type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="rollbackReference"
type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="projects" type="ArrayOfProject"
/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="ArrayOfProject">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="Project" nillable="true"
type="Project" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="Project">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="id" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="name" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="mrid" type="xs:string" />
    </xs:sequence>
</xs:complexType>
</xs:schema>

```

OnlineScadaModelUpdateReplyMessage

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="OnlineScadaModelUpdateReplyMessage" nillable="true"
type="OnlineScadaModelUpdateReplyMessage" />
    <xs:complexType name="OnlineScadaModelUpdateReplyMessage">
        <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="pendingDeployment"
type="PendingDeployment" />

```

```

        <xss:element minOccurs="1" maxOccurs="1" name="deploymentInfoRequest"
type="xs:boolean" />
        <xss:element minOccurs="1" maxOccurs="1" name="loadedModelAuthoritySetsRequest"
type="xs:boolean" />
    </xss:sequence>
</xss:complexType>
<xss:complexType name="PendingDeployment">
    <xss:sequence>
        <xss:element minOccurs="0" maxOccurs="1" name="mrid" type="xs:string" />
        <xss:element minOccurs="0" maxOccurs="1" name="user" type="xs:string" />
        <xss:element minOccurs="0" maxOccurs="1" name="changeDocURL" type="xs:string" />
        <xss:element minOccurs="0" maxOccurs="1" name="expectedScopeId" type="xs:string" />
    </xss:sequence>
    <xss:element minOccurs="0" maxOccurs="1" name="expectedOriginDeployment"
type="xs:string" />
        <xss:element minOccurs="0" maxOccurs="1" name="rollbackReference"
type="xs:string" />
            <xss:element minOccurs="0" maxOccurs="1" name="projects" type="ArrayOfProject" />
        </xss:sequence>
    </xss:complexType>
<xss:complexType name="ArrayOfProject">
    <xss:sequence>
        <xss:element minOccurs="0" maxOccurs="unbounded" name="Project" nillable="true"
type="Project" />
    </xss:sequence>
</xss:complexType>
<xss:complexType name="Project">
    <xss:sequence>
        <xss:element minOccurs="0" maxOccurs="1" name="id" type="xs:string" />
        <xss:element minOccurs="0" maxOccurs="1" name="name" type="xs:string" />
        <xss:element minOccurs="0" maxOccurs="1" name="mrid" type="xs:string" />
    </xss:sequence>
</xss:complexType>
</xss:schema>

```

eDeploymentStatus

```

<?xml version="1.0" encoding="utf-8"?>
<xss:schema elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xss:element name="eDeploymentStatus" type="eDeploymentStatus" />
    <xss:simpleType name="eDeploymentStatus">
        <xss:restriction base="xs:string">
            <xss:enumeration value="success" />
            <xss:enumeration value="processing" />
            <xss:enumeration value="failure" />
        </xss:restriction>
    </xss:simpleType>
</xss:schema>

```

C.3 Model Manager REST Interface Adapter

The Model Manager Application (MMA) can be configured to periodically invoke external webservices. To enable the Model Manager REST Interface Adapter, select the Enable RESTful Interface Adapter option on the RESTful Interface Tab of the Model Manager Configuration Editor. The invocation time of external webservice and the frequency of invocation per day is also configured in the same tab. External systems could be any GIS systems, planning application, or other application within the utility.

C.3.1 Model Manager and TIBCO Interface

The Model Manager REST Interface Adapter can be configured to periodically invoke the external system called TIBCO (The Information Bus Company) to receive static load curves data. Model Manager consumes the static load curves data which contains the schedule name and schedule points, and updates the global.csv file in the Processing folder.

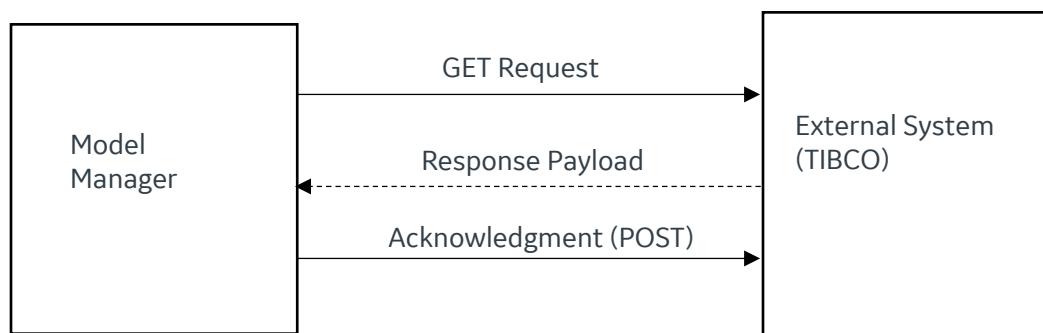


Figure 39. Model Manager - TIBCO System Interface

Along with the configurations in the RESTful Interface tab, the following configurations are added to the Model Manager Configuration file.

- **EnableInvokeExtRESTfulWebServiceForStaticLoadSchedules:** When this configuration value is set to true, the Model Manager RESTful Interface Adapter periodically invokes the external web service for receiving the static load schedules details. The default value of this configuration is false.
- **ExtRESTfulWebServiceEndPointAddressToGetStaticLoadSchedules:** This configuration holds the end point address of external web service that provides the static load schedule details.
- **ExtRESTfulWebServiceEndPointAddressToSendAcknowledgement:** This configuration holds the end point address of external web service to which the acknowledgement is sent after receiving and validating the static load schedule details.

After sending the GET Request through the Model Manager RESTful Adapter, the external system sends a JSON Response.

Following is the JSON Response Payload format from the external system:

```
{
  "Header": {
    "Verb": "reply",
    "Noun": "StaticLoadCurve",
```

```

    "Revision": "1.0",
    "Context": "DEV",
    "Timestamp": "2022-04-28T14:08:19.215+05:30",
    "Source": "EDL",
    "AckRequired": "N",
    "MessageId": "6fc6935d-8e47-41a3-81cc-f701357eb273",
    "CorrelationId": "99bc149f-07ab-45cb-88f6-0dc4c32be616"
},
"Reply": {
    "Result": "PARTIAL",
    "StaticLoadCurveMessage": [
        {
            "Header": {
                "Verb": "create",
                "Noun": "StaticLoadCurve",
                "Revision": "1.0",
                "Context": "DEV",
                "Timestamp": "2022-04-28T14:08:41.166+05:30",
                "Source": "EDL",
                "AckRequired": "N",
                "MessageId": "99bc149f-07ab-45cb-88f6-0dc4c32be616",
                "CorrelationId": "6780"
            },
            "Payload": {"SchedulesDetails": [
                {
                    "ScheduleName": "RESIDENTIAL_SPRING_WEEKDAY",
                    "SchedulePoints": [
                        {"XVALUE": "1:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9},
                        {"XVALUE": "2:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9},
                        {"XVALUE": "3:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9},
                        {"XVALUE": "4:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9},
                        {"XVALUE": "5:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9},
                        {"XVALUE": "6:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9},
                        {"XVALUE": "7:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9},
                        {"XVALUE": "8:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9},
                        {"XVALUE": "9:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9},
                        {"XVALUE": "10:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9},
                        {"XVALUE": "11:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9},
                        {"XVALUE": "12:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9},
                        {"XVALUE": "13:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9},
                        {"XVALUE": "14:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9},
                        {"XVALUE": "15:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9},
                        {"XVALUE": "16:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9},
                        {"XVALUE": "17:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9},
                        {"XVALUE": "18:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9},
                        {"XVALUE": "19:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9},
                        {"XVALUE": "20:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9},
                        {"XVALUE": "21:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9},
                        {"XVALUE": "22:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9},
                        {"XVALUE": "23:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9},
                        {"XVALUE": "0:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9}
                    ]
                }
            ]}
        }
    ]
}

```

```

},
{
  "Header": {
    "Verb": "create",
    "Noun": "StaticLoadCurve",
    "Revision": "1.0",
    "Context": "DEV",
    "Timestamp": "2022-04-28T14:34:41.166+05:30",
    "Source": "EDL",
    "AckRequired": "N",
    "MessageId": "99bc149f-07ab-45cb-88f6-0dc4c32be616",
    "CorrelationId": "6780"
  },
  "Payload": {"SchedulesDetails": [
    {
      "ScheduleName": "RESIDENTIAL_SPRING_WEEKEND",
      "SchedulePoints": [
        {"XVALUE": "1:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9},
        {"XVALUE": "2:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9},
        {"XVALUE": "3:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9},
        {"XVALUE": "4:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9},
        {"XVALUE": "5:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9},
        {"XVALUE": "6:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9},
        {"XVALUE": "7:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9},
        {"XVALUE": "8:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9},
        {"XVALUE": "9:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9},
        {"XVALUE": "10:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9},
        {"XVALUE": "11:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9},
        {"XVALUE": "12:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9},
        {"XVALUE": "13:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9},
        {"XVALUE": "14:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9},
        {"XVALUE": "15:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9},
        {"XVALUE": "16:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9},
        {"XVALUE": "17:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9},
        {"XVALUE": "18:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9},
        {"XVALUE": "19:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9},
        {"XVALUE": "20:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9},
        {"XVALUE": "21:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9},
        {"XVALUE": "22:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9},
        {"XVALUE": "23:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9},
        {"XVALUE": "0:00", "Y1VALUE": 0.944, "Y2VALUE": 0.9}
      ]
    }
  ]
}
]
}
}

```

The Model Manager Restful Interface Adapter consumes the response payload, does validation, and sends the acknowledgement (POST Request) to the external system.

Note: Minimal data format validation will be performed by the MMA, and the data validation is expected to be performed in the source system.

Following a failure, the JSON acknowledgement is sent to the external system indicating when the Interface Adapter is unable to deserialize the schedules payload data.

```
{"Status": "1", "ErrorMsg": "Unable to Deserialize Static Load Schedules Data"}
```

Following a failure, the acknowledgement is sent to the external system indicating when any schedule doesn't contain the 24 schedule points.

```
{"Status": "1", "ErrorMsg": "24 Schedule Points are not provided in a schedule(s)  
<schedule name1>, <schedulenam2>"}
```

Following a success, the acknowledgement is sent to the external system indicating when the Interface Adapter can deserialize the schedules payload data and when all the schedules can contain the necessary schedule points.

```
{"Status": "0", "ErrorMsg": "No Error"}
```