OpenFMB Protobuf Generator Programmer's Guide

Change Log:

Date	Version	Who	Comment
01/12/2018	0.1	LSC	Initial Draft
08/08/2018	1.0	LSC	Initial Release

12/20/2018	1.0	LSC	Updated the following sections:
			• Global
			  ○ Added new class variable informationalMessages
			  ○ Added new class variable warningMessages
			  ○ Added new class variable errorMessages
			  ○ Added new class variable INVALID_PROTOBUF_DATATYPE
			• Main
			  ○ Added new method outputProcessingMessages
			• DataTypeConversion
			  ○ Added new class variable invalidDataTypeCheckDictionary
			  ○ Added new class variable baseDataTypeDictionary
			  ○ Added new method checkInvalidDataType
			  ○ Added new method getBaseDataType

4/18/2019	1.0	CN	Added support for ProtobufTag_openfmb_profile on profile message
			• Main
			  ○ processGlobalElements now returns list of "Proto3MessageEnumeration"
			  ○ buildProto3MessageEnumeration checks for "ProtobufTag_openfmb_profile"
			• Proto3MessageEnumeration
			  ○ If openfmb flag is set, then write "option (uml.option_openfmb_profile) = true;"
			• Proto3ModuleInfo
			  ○ Support multiple "Proto3MessageEnumeration"

08/26/2021	2.0	CN	Update app name and about screenshot

1 Overview

The OpenFMB Protobuf Generator utilizes the Enterprise Architect Add-In facility to provide a menu that allows users to convert an OpenFMB UML model into Protocol Buffer definition files.

2 Classes

2.1 Global

Class containing global variables.

Class Variables	
EA.Repository	repository Reference to the Enterprise Architect Repository object representing the currently open Enterprise Architect model.
UserAction	userAction Reference to the UserAction class. This is an intermediary between the Windows Forms user actions and the Main class.
TextBoxOutput	textBoxOutput Reference to the TextBoxOutput class. This is used to write information to the TextBox (right pane) of the OpenFMBMessageProfileSelector User Interface.
String	umlFileName Name of the Enterprise Architect file that is being processed.
String	protoFileGenerationDateTime Generation Date/Time (in UTC format) of the Protobuf files.
HashSet<String>	checkedElements Reference to the HashSet<T> class. This is used to collect the names of the TreeNode names (UML item names) that will have a check box enabled.
HashSet<String>	hideCheckBoxList Reference to the HashSet<T> class. This is used to collect the names of the TreeNode names (UML item names) that will have their checked box disabled and hidden from view
List<String>	informationalMessages

	Reference to the List<T> class. This is used to collect the informational messages generated during processing. If this list is not empty then this list of messages will be written to the Text Box Output when the generation of the protobuf proto3 files has concluded.
List<String>	warningMessages Reference to the List<T> class. This is used to collect the warning messages generated during processing. If this list is not empty then this list of messages will be written to the Text Box Output when the generation of the protobuf proto3 files has concluded.
List<String>	errorMessages Reference to the List<T> class. This is used to collect the error messages generated during processing. If this list is not empty then this list of messages will be written to the Text Box Output when the generation of the protobuf proto3 files has concluded and the Save Protobuf button will be disabled.
Boolean	errorGeneratingProtobuf Determines if there were any errors during the generation of the protobuf files.
String	INVALID_PROTOBUF_DATATYPE Constant string "Invalid data type in Open FMB Model" defining the search string to determine an invalid Protobuf data type.

2.2 Main

Main control point of the application.

Class Variables	
String	menuHeader The top-level menu of the Enterprise Architect to Protocol Buffer (ProtoBuf) Exporter.
String	menuItem_Generate_proto3 Sub-menu of the top-level menu, when selected it will initiate the generate protobuf proto3 process.
String	menuItem_About Sub-menu of the top-level menu, when selected it will display a window showing the information about the Enterprise Architect to Protocol Buffer (ProtoBuf) Exporter.
String	logFileName Name of the log file where processing information will be saved. The log file name is determined by the absolute path, file name of the Enterprise Architect Project (.eap) file, and the current date and time. For example, if the Enterprise Architect Project (.eap) file is located in "C:\OpenFMB\UML\ OpenFMB Model (61850-CIM) - 2018-04-16.eap" then the corresponding log file will be located in "C:\OpenFMB\UML\ OpenFMB Model (61850-CIM) - 2018-04-16_HH.mm.ss.log".
OpenFMBMessageProfileSelector	profileSelector Reference to the OpenFMBMessageProfileSelector class. This is used to setup and show the user interface.
TreeNodeCollection	treeNodes Reference to the TreeNodeCollection class. This is used to represent a collection of TreeNode objects based on the Open FMB UML model.
TreeNode	parentPackageNode Reference to the TreeNode class. This is used to hold the current top-level UML package containing

	the Open FMB UML packages (Open FMB Parent Package) and will check or uncheck the check box accordingly.
HashSet<String>	unnestedElements Reference to the HashSet<T> class. This is used to collect the names of the UML elements that contain the name/value pair of "nested" and "FALSE" or "ProtobufTag_extend" and "TRUE". This combined with the recursiveProcessing flag will determine when to recursively process the UML elements.
Boolean	checkModelNode Determines if the check box at the UML model level is checked or not.
Boolean	checkParentPacakageNode Determines if the check box at the top-level UML package containing the Open FMB UML packages (Open FMB Parent Package) is checked or not.
Boolean	checkPackageNode Determines if the check box at the UML package containing an Open FMB module is checked or not.
Proto3ModuleInfo	proto3GlobalModuleInfo Reference to the Proto3ModuleInfo class containing the module header information for all protobuf proto3 files.
HashSet<int>	importPackageIDs Reference to the HashSet<T> class. This is used to collect the UML package IDs of the packages that need to be imported by the UML packages containing an Open FMB modules. This helps control the recursive processing so that we don't process the entire model.
List<String>	proto3FileNames Reference to the List<T> class. This is used to collect the packageName of the Proto3File classes that have been processed. This will prevent a packageName from being processed more than once.
List<Proto3File>	proto3Files

	Reference to the List<T> class. This is used to collect the Proto3File classes.
String	childPackageName Package name of the UML package containing an Open FMB module that is currently being processed. This helps to determine if a package import encountered during processing needs to be imported. If the name of the package import is the same as the childPackageName then we are within the same package and the import is not needed (a package cannot import itself).
Boolean	recursiveProcessing Determines if the protobuf processing should be recursive or not.

Methods	
public String EA_Connect(Repository repository)	Enterprise Architect event that enables the Add-In to identify their type and to respond to Enterprise Architect start up. This event occurs when Enterprise Architect first loads your Add-In. Enterprise Architect itself is loading at this time so that while a Repository object is supplied, there is limited information that you can extract from it. The main use is to initialize the global Add-In data. Parameters: • repository – Enterprise Architect Repository object representing the currently open Enterprise Architect model. Return Value: • A string identifying a non-specialized Add-In.
public void EA_Disconnect()	Enterprise Architect event that enables the Add-In to respond to user requests to disconnect the model branch from an external project. called when the Enterprise Architect closes. If you have stored references to Enterprise Architect objects (not particularly recommended anyway), you must release them here.

	Note: .NET users must call memory management functions.
public pbject EA_GetMenuItems(Repository repository, String menuLocation, String menuName)	Enterprise Architect event that enables the Add-In to provide the Enterprise Architect user interface with additional Add-In menu options in various context and main menus. When a user selects an Add-In menu option, an event is raised and passed back to the Add-In that originally defined that menu option. This event is raised just before Enterprise Architect shows menu options to the user. Parameters: • repository – Enterprise Architect Repository object representing the currently open Enterprise Architect model. • menuLocation – String representing the part of the user interface that brought up the menu. This can be TreeView, MainMenu, or Diagram. • menuName - Name of the parent menu for which sub-items are to be defined. In the case of the top-level menu this is an empty string. Return Value: One of the following types: • string indicating the label for a single menu option. • array of strings indicating a multiple menu option. • null to indicate that no menu should be displayed. In the case of the top-level menu it should be a single string or an array containing only one item, or empty/null.
public Boolean IsProjectOpen(Repository repository)	Tests to determine if the Enterprise Architect Repository object representing the currently open Enterprise Architect model is accessible. Parameters: • repository – Enterprise Architect

	Repository object representing the currently open Enterprise Architect model. Return Value: • Boolean to indicate whether the Enterprise Architect model is accessible. A value of true is returned if the model is accessible.
public void EA_GetMenuState(Repository repository, String menuLocation, String menuName, String menuItemName, ref Boolean isMenuEnabled, ref Boolean isMenuChecked)	Enterprise Architect event that enables the Add-In to set a menu option to either enabled or disabled. This is useful when dealing with locked packages and other situations where it is convenient to show a menu option, but not enable it for use. This event is raised just before Enterprise Architect shows menu options to the user. Parameters: • repository – Enterprise Architect Repository object representing the currently open Enterprise Architect model. • menuLocation – String representing the part of the user interface that brought up the menu. This can be TreeView, MainMenu, or Diagram. • menuName - Name of the parent menu for which sub-items are to be defined. In the case of the top-level menu this is an empty string. • menuItemName – String containing the menu option clicked. • isMenuEnabled – Boolean when set to false indicates to disable the menu option. • isMenuChecked – Boolean when set to true indicates to check the menu option.
public void EA_MenuClick(Repository repository, String menuLocation, String menuName, String menuItemName)	Enterprise Architect event that is received by the Add-In in response to user selection of a menu option. This event is raised just before Enterprise Architect shows menu options to the user. Parameters:

	• repository – Enterprise Architect Repository object representing the currently open Enterprise Architect model. • menuLocation – String representing the part of the user interface that brought up the menu. This can be TreeView, MainMenu, or Diagram. • menuName - Name of the parent menu for which sub-items are to be defined. In the case of the top-level menu this is an empty string. • menuItemName – String containing the menu option clicked.
internal static void populateTreeView(OpenFMBMessageProfileSelector profileSelector)	Populates the OpenFMBMessageProfileSelector tree view with the relevant Open FMB packages and classes from the model. Parameters: • profileSelector – Reference to the OpenFMBMessageProfileSelector class. This class provides access to the tree view to be populated.
private static void populateTreeNodes(TreeNode parentNode, Package parentPackage, Boolean isParentPackage)	Populates the tree nodes in the OpenFMBMessageProfileSelector tree view with the relevant Open FMB packages from the model. Parameters: • parentNode – TreeNode containing the current parent tree node that the child tree nodes will be added to. • currentPackage – Enterprise Architect Package containing the current UML package that is being processed. • isParentPackage – Boolean to indicate that the currentPackage is the top-level UML package containing the Open FMB UML packages (Open FMB Parent Package). This is only set to true the first time PopulateTreeNodes is called.
private static void checkParentNode(TreeNode currentNode)	Sets the check box of all parent nodes of the current node in the. Parameters: • currentNode – TreeNode containing the current tree node.

private static void populatePackageClassesAndEnumerations(TreeNode parentNode, Package currentPackage)	Populates the tree nodes in the OpenFMBMessageProfileSelector tree view with the relevant Open FMB classes and enumerations from the model. Parameters: • parentNode – TreeNode containing the current parent tree node that the child tree nodes will be added to. • currentPackage – Enterprise Architect Package containing the current UML package that is being processed.
private static Boolean isClass(element element)	Determines if the current UML element being processed in PopulatePackageClassesAndEnumerations is a class. Parameters: • element – Enterprise Architect Element that is currently being processed. Return Value: • Boolean to indicate whether the Enterprise Architect Element is a class. A value of true is returned if the Enterprise Architect Element is a class.
private static Boolean isEnumeration(Element element)	Determines if the current UML element being processed in PopulatePackageClassesAndEnumerations is an enumeration. Parameters: • element – Enterprise Architect Element that is currently being processed. Return Value: • Boolean to indicate whether the Enterprise Architect Element is an enumeration. A value of true is returned if the Enterprise Architect Element is an enumeration.
private static Boolean isUMLDiagram(Element element)	Determines if the current UML element being processed in PopulatePackageClassesAndEnumerations is a UML diagram. Parameters:

	• element – Enterprise Architect Element that is currently being processed. Return Value: • Boolean to indicate whether the Enterprise Architect Element is a UML diagram. A value of true is returned if the Enterprise Architect Element is a UML diagram.
private static void removeFromHideCheckBoxList(String nodeName)	Determines if the current node name is in the HideCheckBoxList. If it is then it will be removed. Parameters: • nodeName – Node name which is to be removed from the HideCheckBoxList. This node name is typically a package name that contains an Open FMB mdule.
private static void initializeGenerateProto3Variables()	Initializes the class variables that are specific to the generation of the protobuf files. This is called every time the "Generate Protobuf" button is clicked to clear out the information from a previous generate process. This allows the user to fix any issues encountered during the generate process without closing and reopening the Add-In.
internal static void generateProto3()	Control point for the generation of the protobuf proto3 files.
private static void processPackage (Package package)	Processes the UML package passed in. Determines if the package is part of the Open FMB Model that will be converted to the Protobuf format. Loops through all elements and child packages belonging to the package passed in and calls the appropriate methods for further processing. This method is called recursively for each package. Parameters: • package – Enterprise Architect Package containing the current UML package that is being processed.
private static void processPackageTaggedValues(Package package)	Processes the Tagged Values of the UML package passed in. The Tagged Values at the UML package level are used to populate the Proto3ModuleInfo class. Parameters:

	• package – Enterprise Architect Package containing the current UML package that is being processed.
private static Proto3MessageEnumeration processGlobalPackageElements(Package package)	Processes the UML Elements of the UML package passed in (Open FMB Parent Package). The Elements at the Open FMB Parent Package level are used to populate the Proto3MessageEnumeration class within the Proto3ModuleInfo class. Parameters: • package – Enterprise Architect Package containing the current UML package that is being processed.
private static void processPackageElement(Element element, Proto3File proto3File)	For the UML Element passed in, calls the BuildProto3MessageEnumeration method if the processing is recursive or if the element is a top level element. Parameters: • element – reference to the Enterprise Architect Element containing the current UML element that is being processed. • proto3File – reference to the Proto3File class containing the current Proto3File class. This class is the top level container for all protobuf proto3 files except for the uml.proto file.
private static Proto3File buildProto3File(Package package)	Populates the Proto3File class with the package name and the parent package name. Parameters: • package – Enterprise Architect Package containing the current UML package that is being processed. Return Value: • Proto3File class containing the UML package information to be converted to the proto3 file header information.
private static Proto3MessageEnumeration buildProto3MessageEnumeration(Element element, Proto3File proto3File)	Processes the UML Element passed in and populates the Proto3MessageEnumeration class. Parameters: • element – reference to the Enterprise Architect Element containing the current UML element that is being processed.

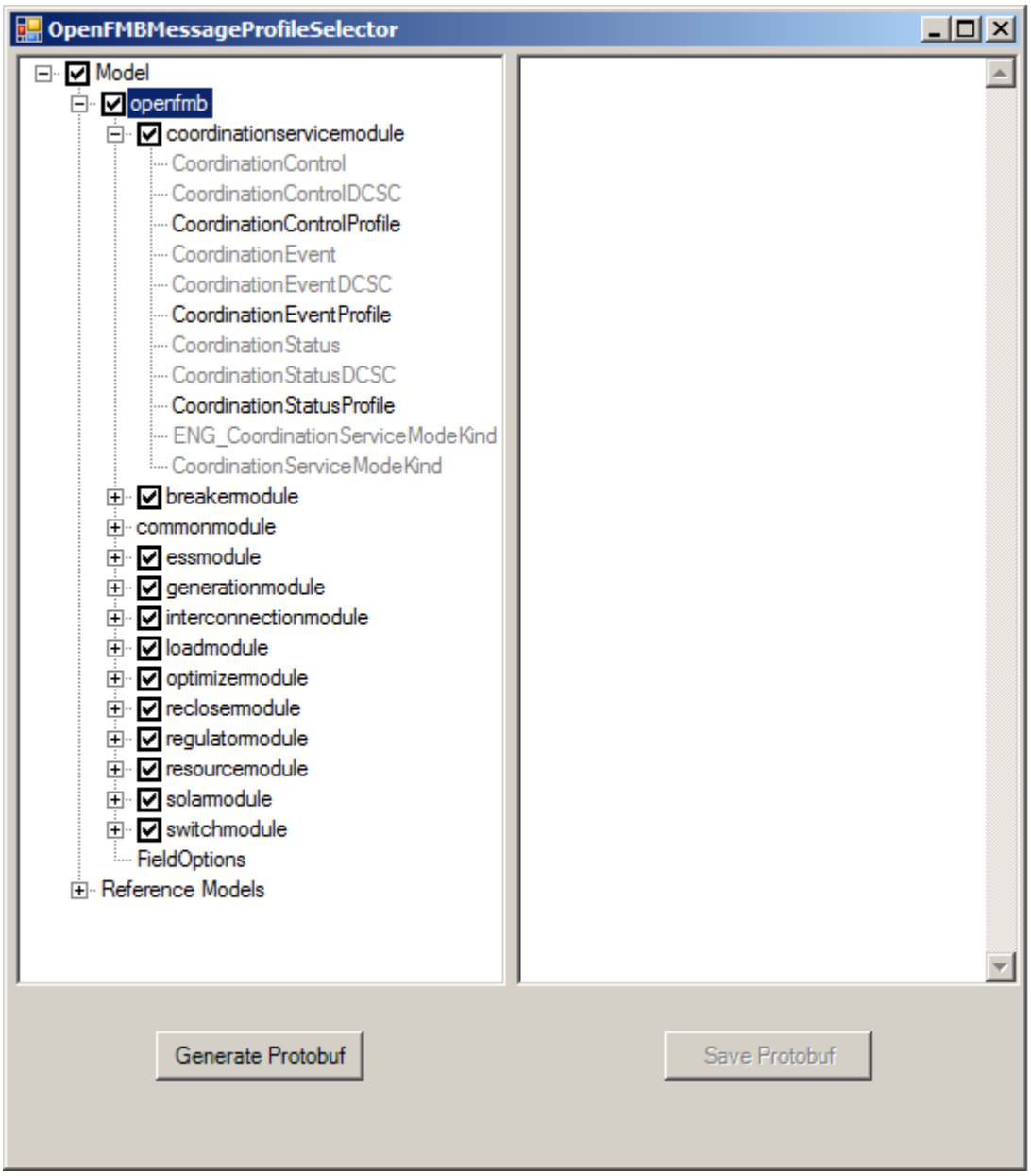
	• proto3File – reference to the Proto3File class containing the current Proto3File class. This class is the top level container for all protobuf proto3 files except for the uml.proto file.
private static List<Proto3Field> buildProto3Fields(Element element, Proto3File proto3File)	Processes the Attributes of the UML Element passed in and populates the Proto3Field class for each Attribute. Parameters: • element – reference to the Enterprise Architect Element containing the current UML element that is being processed. • proto3File – reference to the Proto3File class containing the current Proto3File class. This class is the top level container for all protobuf proto3 files except for the uml.proto file.
private static Proto3Field processElementAttributes(Element element, EA.Attribute attribute, Proto3File proto3File)	Processes the Attributes of the UML Attribute passed in and populates the Proto3Field class. These Attributes contain information for multiplicity and default values of the field as well as other Protobuf decorations. Parameters: • element – reference to the Enterprise Architect Element containing the current UML element that is being processed. • attribute – reference to the Enterprise Architect Attribute containing the current UML element attribute that is being processed. • proto3File – reference to the Proto3File class containing the current Proto3File class. This class is the top level container for all protobuf proto3 files except for the uml.proto file.
private static String getBaseDataType(Element element)	Determines the base data type class of the UML Element passed in. Parameters: • element – reference to the Enterprise Architect Element containing the current UML element that is being processed.
private static Proto3Field processConnector(Connector connector, Proto3File proto3File)	Processes the UML Connector passed in, determines whether it is an Association or Generalization, calls the appropriate method for

	further processing and calls the method BuildProtosMessageEnumeration. Parameters: • connector – reference to the Enterprise Architect Connector containing the current UML element that is being processed. • proto3File – reference to the Proto3File class containing the current Proto3File class. This class is the top level container for all protobuf proto3 files except for the uml.proto file.
private static Proto3Field processAssociation(Package package, Element element, Connector connector)	Processes the UML Connector passed in and populates the Proto3Field class. Parameters: • package – reference to the Enterprise Architect Package of the target Enterprise Architect Element of the association that is being processed. • element - reference to the target Enterprise Architect Element of the association that is being processed. • connector – reference to the Enterprise Architect Connector containing the current UML element that is being processed.
private static Proto3Field processGeneralization(Package package, element element, Connector connector)	Processes the UML Connector passed in and populates the Proto3Field class. Parameters: • package – reference to the Enterprise Architect Package of the target Enterprise Architect Element of the association that is being processed. • element - reference to the target Enterprise Architect Element of the association that is being processed. • connector – reference to the Enterprise Architect Connector containing the current UML element that is being processed.
Private static void outputProcessingMessages(List<String> messages)	Loops through the list of processing messages (informational, warning, and error) and writes each message to the Text Box Output.

	Parameters: • messages – reference to the list of processing messages (informational, warning, and error).
internal static void saveProto3(String selectedPath)	This is the control point for writing all Protobuf files. Parameters: • selectedPath – the path name selected by the user when the "Save Protobuf" button is pressed.
private static Boolean isPackageSelected(String packageName)	Loops through the TreeNode to determine which nodes (package name) have their check boxes checked. Each node that has their check box checked is written. Parameters: • packageName – the package name from each Proto3File class used to identify the nodes in the TreNode.

2.3 OpenFMBMessageProfileSelector

This is the high-level Windows Form for the User Interface with a TreeView in the left pane and a TextBox in the right pane.



Class Variables	
int	TVIF_STATE Mask to indicate the state and stateMask members are valid. Default value is 0x80.
Int	TVIS_STATEIMAGEMASK

	Mask to set the state image by isolating bits 12 through 15 of the state member. Default value is 0xF000.
Int	TV_FIRST Mask to indicate the start of the TreeView messages. Default value is 0x1100.
Int	TVM_SETITEM Mask to indicate the attributes to set. Default value is TV_FIRST + 63.

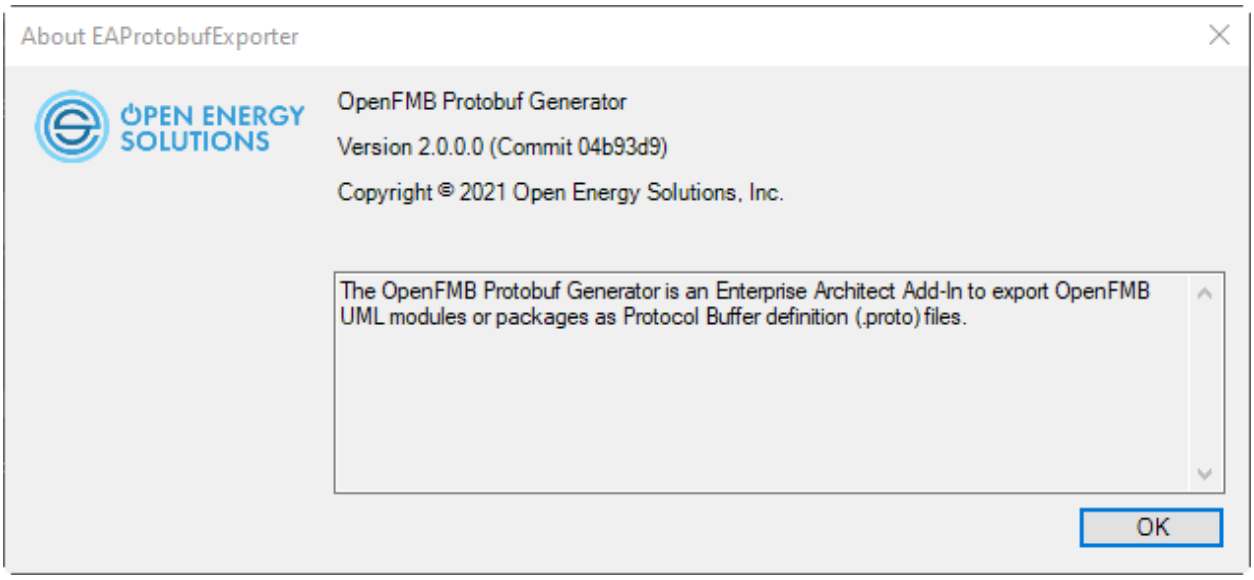
Structure	
int	mask
IntPtr	hItem
int	State
int	stateMask

Methods	
public void hideCheckBox(TreeNode treeNode)	Hides the TreeNode check box of the node passed in. Parameters: • treeNode – reference to the TreeNode who's check box will be hidden.
Public void setCheckBoxOfAllChildNodes(TreeNode treeNode, bool nodeChecked)	Checks or unchecks all child nodes of the tree node passed in. Parameters: • treeNode – reference to the TreeNode whose check box will be hidden. • nodeChecked – Boolean to indicate whether to check (true) or uncheck (false) the tree node's check box.
Public void setCheckBoxOfAllParentNodes(TreeNode treeNode, bool nodeChecked)	Checks or unchecks all parent nodes of the tree node passed in. Parameters: • treeNode – reference to the TreeNode whose check box will be hidden. • nodeChecked – Boolean to indicate whether to check (true) or uncheck (false) the tree node's check box.
Private void node_AfterCheck(object sender,	Called when a node is checked in the user

TreeViewEventArgs eventArgs)	interface. Sets the check boxes of parent and child nodes accordingly. Parameters: • sender – reference to the control or object that raised the event. • eventArgs – reference to the TreeViewEventArgs class.
Public void whatNodesAreChecked()	Loops through the tree nodes to determine which ones are checked.
Private void processTreeNodes(TreeNodeCollection treeNodes, int depth)	Loops through the lower level tree nodes to determine which ones are checked and writes the names of the checked nodes to the TreeView in the user interface.
Public void disableGenerateButton()	Disables the Generate Proto button.

2.4 AboutBox

This is a Windows Form containing information about the application.



2.5 UserAction

This is the Windows Form inherited class to process user actions.

Methods	
public void OnGenerateAction()	Processes the "Generate Protobuf" button click.
Public void OnSaveAction()	Processes the "Save Protobuf" button click.

2.6 DataTypeConversion

Converts from UML datatypes (Key) to Protobuf datatypes (Value). The class contains a Dictionary of key/value pairs and a supporting method to convert from the UML datatypes to the Protobuf datatypes.

Class Variables	
Dictionary<String, String>	invalidDataTypeCheckDictionary Dictionary of Key/Value pairs to correspond the UML datatypes (Key) to invalid Protobuf datatypes (Value).
Dictionary<String, String>	dataTypeDictionary Dictionary of Key/Value pairs to correspond the UML datatypes (Key) to Protobuf datatypes (Value).
Dictionary<String, String>	baseDataTypeDictionary Dictionary of Key/Value pairs to correspond the UML base datatypes (Key) to Protobuf datatypes (Value). The UML base datatypes are determined from base data type classes.

invalidDataTypeDictionary

Key	Value
bool	bool – Invalid data type in Open FMB Model
byte	int32 – Invalid data type in Open FMB Model
float	float – Invalid data type in Open FMB Model
int	int32 – Invalid data type in Open FMB Model
integer	int32 – Invalid data type in Open FMB Model
INT8	int32 – Invalid data type in Open FMB Model
INT16	int32 – Invalid data type in Open FMB Model
INT8U	uint32 – Invalid data type in Open FMB Model
INT16U	uint32 – Invalid data type in Open FMB Model
negativeInteger	sint32 – Invalid data type in Open FMB Model
nonPositveInteger	sint64 – Invalid data type in Open FMB Model
short	int32 – Invalid data type in Open FMB Model
long	int64 – Invalid data type in Open FMB Model
double	double – Invalid data type in Open FMB Model

decimal	double – Invalid data type in Open FMB Model
hexBinary	bytes – Invalid data type in Open FMB Model
nonNegativeInteger	int32 – Invalid data type in Open FMB Model
normalizedString	string – Invalid data type in Open FMB Model
positiveInteger	int32 – Invalid data type in Open FMB Model
unsignedByte	uint32 – Invalid data type in Open FMB Model
unsignedInt	uint32 – Invalid data type in Open FMB Model
unsignedShort	uint32 – Invalid data type in Open FMB Model
unsignedLong	uint64 – Invalid data type in Open FMB Model
ObjRef	Invalid data type in Open FMB Model

dataTypeDictionary

Key	Value
boolean	bool
string	string
float	float
FLOAT32	float
INT32	int32
INT64	int64
INT32U	uint32
INT64U	uint64
dateTime	int64
uuidType	uuidType

baseDataTypeDictionary

Key	Value
boolean	bool
bool	bool
string	string
byte	int32
float	float
int	int32

integer	int32
negativeInteger	sint32
nonPositveInteger	sint64
short	int32
long	int64
double	double
decimal	double
hexBinary	bytes
nonNegativeInteger	int32
normalizedString	string
positiveInteger	int32
unsignedByte	uint32
unsignedInt	uint32
unsignedShort	uint32
unsignedLong	uint64
dateTime	int64

Methods	
public static String checkInvalidDataType(String umlDataType)	Determines if the Key (string) passed in has a corresponding Value. If not, then a null is returned. Parameters: • umlDataType – UML datatype (Key) of the UML element to be converted. Return Value: • String containing the corresponding invalid Protobuf datatype (Value) of the UML element.
Public static String getProto3DataType(String umlDataType)	Determines if the Key (string) passed in has a corresponding Value. If not, then a null is returned. Parameters: • umlDataType – UML datatype (Key) of the UML element to be converted. Return Value:

	String containing the corresponding Protobuf datatype (Value) of the UML element.
Public static String getBaseDataType(String umlDataType)	Determines if the Key (string) passed in has a corresponding Value. If not, then a null is returned. Parameters: • umlDataType – UML base datatype (Key) of the UML element to be converted. Return Value: String containing the corresponding Protobuf datatype (Value) of the UML element.

2.7 PrimitiveDataTypeWrappers

Converts from Protobuf datatypes (Key) to Protobuf primitive datatype wrappers (Value). The class contains a Dictionary of Key/Value pairs and a supporting method to convert from the Protobuf datatypes to the Protobuf primitive datatype wrappers.

Note: The Protobuf Primitive Datatype Wrappers allow the Protobuf users to determine if a field with a type of one of the primitive datatypes is optional.

Class Variables	
Dictionary<String, String>	primitiveDataTypeWrapperDictionary Dictionary of Key/Value pairs to correspond the Protobuf datatypes (Key) to Protobuf primitive datatype wrappers (Value).

dataTypeDictionary

Key	Value
double	DoubleValue
float	FloatValue
int64	Int64Value
uint64	UInt64Value
int32	Int32Value
uint32	Uint32Value
bool	BoolValue
string	StringValue
bytes	BytesValue

Methods	
public static String getWrapperDataType(String umlDataType)	Determines if the Key (string) passed in has a corresponding Value. If not, then a null is returned. Parameters: • primitiveDataType – Protobuf datatype (Key) of the UML element to be converted. Return Value: • String containing the corresponding Protobuf primitive datatype wrapper (Value) of the UML element.

2.8 Proto3ModuleInfo

Contains the module header information for all protobuf proto3 files.

Class Variables	
String	packageName Top-level UML package containing the Open FMB UML packages (Open FMB Parent Package).
String	comment UML user comments on the Open FMB Parent Package.
String	stereotype Stereotype assigned to the FieldOptions class under the Open FMB Parent Package.
List<Proto3MessageEnumeration>	proto3MessageEnumerations Reference to the Proto3MessageEnumeration class. This is used to store the FieldOptions and MessageOptions class information.
String	selectedPath User selected absolute path where the protobuf files will be written.
WriteProto3File	writeProto3File Reference to the WriteProto3File class. This is used to write out the global protobuf file (uml.proto).
String	IMPORT_PACKAGE_NAME Constant string "descriptor.proto" defining the file name for the import package in the global protobuf file.
String	GO_PACKAGE_OPTION_NAME Constant string "ProtobufTag_go_package" defining the name portion of the name/value pair for the protobuf go_package option. One of the custom Tagged Values of the top-level Open FMB package.
String	goPackageOptionValue The value portion of the name/value pair for the protobuf go_package option. One of the custom Tagged Values of the top-level Open FMB package.
String	JAVA_PACKAGE_OPTION_NAME

	Constant string "ProtobufTag_java_package" defining the name portion of the name/value pair for the protobuf java_package option. One of the custom Tagged Values of the top-level Open FMB package.
String	javaPackageOptionValue The value portion of the name/value pair for the protobuf java_package option. One of the custom Tagged Values of the top-level Open FMB package.
String	JAVA_MULTIPLE_FILES_OPTION_NAME Constant string "ProtobufTag_java_multiple_files" defining the name portion of the name/value pair for the protobuf java_multiple_files option. One of the custom Tagged Values of the top-level Open FMB package.
String	javaMultipleFilesOptionValue The value portion of the name/value pair for the protobuf java_multiple_files option. One of the custom Tagged Values of the top-level Open FMB package.
String	CSHARP_NAMESPANCE_OPTION_NAME Constant string "ProtobufTag_csharp_namespace" defining the name portion of the name/value pair for the protobuf csharp_namespace option. One of the custom Tagged Values of the top-level Open FMB package.
String	cSharpNamespaceOtionValue The value portion of the name/value pair for the protobuf csharp_namespace option. One of the custom Tagged Values of the top-level Open FMB package.
String	IMPORT_NAME Constant string "ProtobufTag_package" defining the name portion of the name/value pair for the top-level package name. This is used to build an import statement in all protobuf files. One of the custom Tagged Values of the top-level Open FMB package.
String	importValue

	The value portion of the name/value pair for the top-level package name. This is used to build an import statement in all protobuf files. One of the custom Tagged Values of the top-level Open FMB package.

Constructor	
public Proto3GlobalInfo(String packageName)	Parameters: • packageName – package name of the global proto3 file.

Methods	
public Boolean isValid()	Checks all required class variables, to determine if any are null or blank. If a null or blank required class variable is found, then an error is returned. Return Value: • A Boolean to indicate if the global information is valid or not. Returns true if no errors are detected. Returns false if any error is detected.
Public void write()	Writes the uml.proto file to the path constructed from the class variables selectedPath and packageName.
Public void print(int indent)	Writes the values of the class variables to the text box and calls the Print method of the associated Proto3MessageEnumeration class. Parameters: • indent – number of indents to prepend to the line to write.

2.9 Proto3File

Contains the UML package information to be converted to the proto3 file header information.

Class Variables	
String	parentPackageName Top-level UML package containing the Open FMB UML packages (Open FMB Parent Package).
String	packageName Name of the UML package containing an Open FMB module. This package represents one protobuf file.
String	comment UML user comments on the Open FMB module.
List<String>	importPackageNames A list of package names referenced within the current UML package (Open FMB module). These packages are external to the current package and are imported during compilation of the protobuf files.
List<Proto3MessageEnumeration>	proto3MessageEnumerations List of Proto3MessageEnumerations contained within the current package. The Proto3MessageEnumerations is UML element information to be converted to the proto3 message/enumeration information.
Boolean	selectedPackage Determines if the current package has been selected to be saved.
String	selectedPath User selected absolute path where the protobuf files will be written.
WriteProto3File	writeProto3File Reference to the WriteProto3File class. This is used to write out the protobuf files.
Boolean	primitiveDataTypeWrapperSet Specifies whether one of the Proto3MessageEnumeration classes has a

	Proto3Field class with a datatype of the one of the Protobuf PrimitiveDataTypeWrappers. If set to true, an import statement will be written to the file specifying the wrapper file (google/protobuf/wrappers.proto).

Constructor	
public Proto3File(String parentPackageName, String packageName)	Parameters: • parentPackageName – package name of the top-level pacakage (Open FMB Parent Package). • packageName – package name of the global proto3 file.

Methods	
public Proto3MessageEnumeration findProto3MessageEnuemeration(String elementName)	Searches the current list of Proto3MessageEnumeration classes in the Proto3File class to see if the Proto3MessageEnumeration class exists. Parameters: • elementName – name of the UML element that is currently being processed. This name is used to search for a corresponding Proto3MessageEnumeration class. Return Value: • A reference to the Proto3MessageEnumeration class found. A null value will be returned if a corresponding class is not found.
Public void write(Proto3GlobalInfo proto3GlobalInfo)	Writes the protobuf file associated with the packageName to the path constructed from the class variables selectedPath, parentPackageName, and packageName calls the Write method of each the associated Proto3MessageEnumeration class in the list of Proto3MessageEnumeration classes. Parameters: • proto3GlobalInfo – reference to the protobuf proto3 global information.
Public void print(int indent)	Writes the values of the class variables to the text box and calls the Print method of each the associated Proto3MessageEnumeration class in the list of Proto3MessageEnumeration classes.

	Parameters: • indent – number of indents to prepend to the line to write.

2.10 Proto3MessageEnumeration

Contains the UML element information to be converted to the proto3 message/enumeration information.

Class Variables	
String	type Specifies the type of information contained in the Proto3MessageEnumeration class. The valid types: message; enum; extend.
String	name Name of the UML element represented by the Proto3MessageEnumeration class.
String	comment UML user comments on the UML element.
Boolean	writeEnumeration Specifies whether to write the list of Proto3Field classes as enumerations. This parameter is set to true when the type class variable is set to "enum".
String	fieldVariableNamePrefix Prefix to be added to the variable names in the list of Proto3Field classes when the type class variable is set to "enum".
List<String>	proto3Fields List of Proto3Fields representing the UML element attributes associated with the Proto3MessageEnumeration class information (UML element).
Boolean	primitiveDataTypeWrapperSet Specifies whether one of the Proto3Fields has a datatype of the one of the Protobuf PrimitiveDataTypeWrappers.

Constructor	
public Proto3MessageEnumeration(String type, String name)	Parameters: • type – type of information contained in the Proto3MessageEnumeration class. The valid types: message; enum; extend. • name – name of the UML element

	represented by the Proto3MessageEnumeration class.

Methods	
public void write(WriteProto3File writeProto3File, Boolean writeComment)	Writes the Proto3MessageEnumeration information to the protobuf file specified by the writeProto3File parameter and calls the Write method of each the associated Proto3Field class in the list of Proto3Field classes. Parameters: • writeProto3File – reference to the protobuf file to be written to. • writeComment – controls whether or not to write the contents of the comments class variable to the protobuf file. This parameter is false when called from the Proto3GlobalInfo class and true when called from the Proto3File class.
Public Boolean sortProto3Fields()	Sorts the list of Proto3Fields based on the default value in each Proto3Field from the list. If the type class variable is "message" then the private method sortProtobufTags is called. If the type class variable is "enum" then the list is sorted irrespective of the default values being consecutive numbers.
Private Boolean sortProtobufTags()	Sorts the list of Proto3Fields based on the default value in each Proto3Field from the list. The default values represent the ProtobufTag name/value pair of the Tagged Values of the UML element attribute. The default values of the UML element attributes for a parent UML element must be consecutive numbers starting at 1.
Public void print(int indent)	Writes the values of the class variables to the text box and calls the Print method of each the associated Proto3Field class in the list of Proto3Field classes. Parameters: • indent – number of indents to prepend to the line to write.

2.11 Proto3Field

Contains the UML attribute information to be converted to the proto3 field information.

Class Variables	
String	variableType UML class associated with the variable or field represented by the Proto3Field class.
String	variableName UML attribute name associated with the variable or field represented by the Proto3Field class.
String	comment UML user comments on the UML element attribute.
String	proto3FieldName Protobuf proto3 datatype corresponding to the UML attribute datatype.
String	externalPackageName UML package name containing the UML attribute represented by the Proto3Field class. The externalPackageName corresponds to the imported packages specified in the protobuf file.
Int	defaultValue Default value of the variable or field represented by the Proto3Field class. If the type of associated Proto3MessageEnumeration is "message" then the ProtobufTag on the attribute is used as the default value. If the type of associated Proto3MessageEnumeration is "enum" or "extend" then the Initial Values on the attribute is used as the default value.
Boolean	parentMessage Specifies that the UML element attribute is a Generalization or inherited class
int	minMultiplicity Specifies the minimum multiplicity of the UML element attribute to the parent UML element.
Int	maxMultiplicity

	Specifies the maximum multiplicity of the UML element attribute to the parent UML element.
Boolean	infiniteMaxMultiplicity Specifies that the maximum multiplicity of the UML element attribute to the parent UML element is infinite.
Boolean	uuid Specifies whether the UML element attribute has a special meaning for the string datatype. If the UML element attribute contains a Tagged Value with the name of ProtobufTag_UUID and value of TRUE, then the uuid is set to true.
Boolean	key Specifies whether the UML element attribute has a special meaning for publish/subscribe protocols. If the UML element attribute contains a Tagged Value with the name of ProtobufTag_Key and value of TRUE, then the key is set to true.
Boolean	primitiveDataTypeWrapperSet Specifies whether the datatype of the field is one of the Protobuf PrimitiveDataTypeWrappers. If set to true, the field type will be replaced with the corresponding wrapper value from the PrimitiveDataTypeWrappers class.
Boolean	optionalEnumerationWrapperSet Specifies whether the field is an optional enumeration. If set to true, the field type will be prepended with "Optional_".

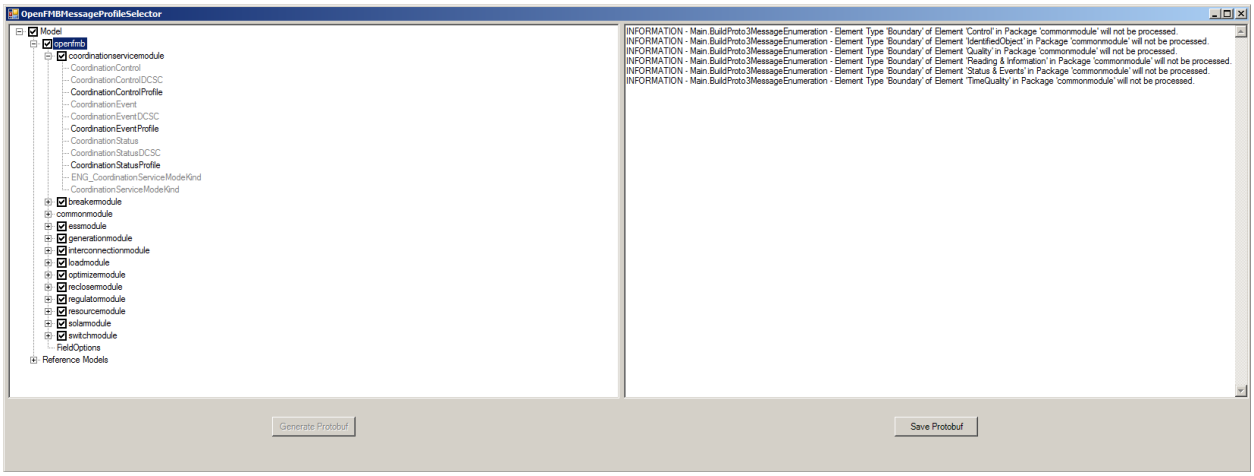
Constructor	
public Proto3Field(String variableType, String variableName)	Parameters: • variableType – UML class associated with the variable or field represented by the Proto3Field class. • variableName – UML attribute name associated with the variable or field represented by the Proto3Field class.

Methods	
public void write(WriteProto3File writeProto3File,	Writes the Proto3Field information to the protobuf

Boolean writeComment, Boolean writeEnumeration, String fieldVariableNamePrefix)	file specified by the writeProto3File parameter. Parameters: • writeProto3File – reference to the protobuf file to be written to. • writeComment – controls whether to write the contents of the comments class variable to the protobuf file. This parameter is false when called from the Proto3GlobalInfo class and true when called from the Proto3File class. • writeEnumeration – specifies whether the Proto3Field information is for an enumeration. This parameter is false if the Proto3Field information is for a message and true if the Proto3Field information is for an enumeration. • fieldVariableNamePrefix – prefix added to the variable name if the Proto3Field information is for an enumeration. This parameter will be null if the writeEnumeration parameter is false.
Public void print(int indent)	Writes the values of the class variables to the text box. Parameters: • indent – number of indents to prepend to the line to write.

2.12 TextBoxOutput

Writes formatted messages to the TextBox (right pane) of the OpenFMBMessageProfileSelector User Interface and writes out the formatted messages to a log file.



Class Variables	
TextBox	40extbox Reference to the Windows Forms TextBox class. This is used to write information to the TextBox (right pane) of the OpenFMBMessageProfileSelector User Interface.
WriteProto3File	writeProto3File Reference to the WriteProto3File class. This is used to write out the log file.

Constructor	
public TextBoxOuput (TextBox 40extbox)	Parameters: • textbox – text box area in the OpenFMBMessageProfileSelector User Interface to be written too.

Methods	
public void clear()	Clears the contents of the TextBox.
Public String indent(int depth)	Prepends the line to write by 4 spaces (indent) times the depth passed in. Parameters: • depth – number of indents to prepend to the line to write.

	Return Value: • A blank string containing the depth of the indent to prepend to the line to write.
Public void outputText(String text)	Appends the text string passed in, without a new line character, to the list of lines to write. Parameters: • text – line to write.
Public void outputText(int depth, String text)	Appends the text string passed in, without a new line character, and indent with the depth passed in to the list of lines to write. Parameters: • depth – number of indents to prepend to the line to write. • text – line to write.
Public void outputTextLine()	Appends a blank line to the list of lines to write.
Public void outputTextLine(String text)	Appends the text string passed in, with a new line character, to the list of lines to write. Parameters: • text – line to write.
Public void outputTextLine(int depth, String text)	Appends the text string passed in, with a new line character, and indent with the depth passed in to the list of lines to write. Parameters: • depth – number of indents to prepend to the line to write. • text – line to write.
Public void writeLogFile(String logFileName)	Writes the list of lines to write to the log file. Parameters: • logFileName – name of the log file to be written. This name includes the full path for the file.

2.13 WriteProto3File

Writes out the proto3 files.

Class Variables	
String	outputFilename Name of the output file name to be written. This name includes the full path for the file.
List<String>	linesToWrite List of lines to write to the file specified by the outputFileName.

Methods	
public void clear()	Clears the contents of the list of lines to write.
public void append(int depth, String text)	Appends the text string passed in and indent with the depth passed in to the list of lines to write. Parameters: • depth – number of indents to prepend to the line to write. • text – line to write.
public String indent(int depth)	Prepends the line to write by 4 spaces (indent) times the depth passed in. Parameters: • depth – number of indents to prepend to the line to write. Return Value: • A blank string containing the depth of the indent to prepend to the line to write.
public void writeComment(int depth, String comment)	Appends the comment string passed in and indent with the depth passed in to the list of lines to write. Parameters: • depth – number of indents to prepend to the line to write. • comment – comment to write.
public void writeFile()	Writes the list of lines to write to the output file name contained in the class variable outputFileName.