

The FileImport Class Library

FileImport is an Integra runtime object class designed for Integra's profile processing. The actual object at runtime is called *import*.

Though the class has the same name and has the same structure for the three Integra profile types (Word, Excel and Text), the availability (usability) of the properties and methods differ.

Integra's Advanced tab allows definition of event driven Callback script. The runtime object *import* can be used within the Callback Script or within an Add-in Script Library, thus giving control over Integra's behaviour during its runtime events.

Below is a list of all the properties and methods of the FileImport class. They are explained in detail throughout this section and availability will be indicated.

Note: ALL properties allow Read/Write access, however, it is indicated if Read Only is Recommended. For the latter you should not modify the value of the properties unless you are very confident in the use of Integra's runtime object classes.

1 FileImport.Continue [Class Property]

Version: 2.x to 4.7
Availability: Word, Excel, Text, Import

This property is used to control Integra's process flow. If import.Continue is True, Integra will continue processing importingdata. If after any event the import.Continue property is set to False, the event or the entire import process will be aborted:

The following table shows the effect of setting import.Continue = False for each of the Integra events (import only):

Event	Effect of setting import.Continue = False	Since Version
CB_INITIALISE	Abort process, jump to CB_TERMINATE event	2.0
CB_BFORIMPORT	Abort process, jump to CB_TERMINATE event	3.0
CB_BFORREADCOM	Skip to next Excel or Text row, or Word document	2.0
CB_BFORWRITENOTES	Skip to next Excel or Text row, or Word document	2.0
CB_TERMINATE	No effect	2.0

If any error exception is triggered, import.Continue is set to False, as well as stats.procstat is set to 0. Both settings effectively abort the entire import process. The profile developer can create a Lotus Script Error Handler in which he can set both process flow properties in order to tell Integra to e.g. ignore the error and to continue processing.

2 FileImport.CopyAndPasteWord [Class Method (Procedure)]

Version: 2.x to 4.7
Availability: Word, Import

CopyAndPasteWord(stats As staticVars, rtfname As String)

Main procedure called for importing Word documents into Notes rich text fields. Since the introduction of the API import as the default method with Integra for Notes version 4, this procedure simply calls the FileImport.ApiImportRTF() method. Unless of course advanced script is used to switch back to the copy and paste method.

Essentially both methods achieve more or less the same results. However, there can be minor differences in the fidelity of the Word to Notes rich text conversion. Sometimes one achieves better results with the Copy & Paste method, rectifying the loss in performance.

Parameters:

stats (StaticVars)

Handle to Integra's runtime object

rtfname (String)

Name of the rich text field in which the Word document content is to be imported to.

3 FileImport.EndOfFile [Class Property]

Version: 2.x to 4.7
Availability: Text, Import

This property indicates that Integra reached the end of the to be imported CSV, Text or XML file. This property is of no use to the developer.

4 FileImport.ErrorText [Class Property]

Version: 2.x to 4.7
Availability: Word, Excel, Text, Import

If an error exception occurs, either because of an internal error of the Integra script libraries or because of an error in any advanced Notes Formula or Lotus Script, the *ErrorText* property contains the error message including error code and line number of occurrence. In the case of an internal error, the text also shows the name of the method which failed and the class it belongs to. At the same time the property is set to True.

Using Lotus Script in the Script Error Handler (Integra profile) the profile developer can analyse the error message for specific errors. If the error handler can correct a particular error, the *ErrorText* property should be blanked ("") and the [import.IsError](#) property set to False.

5 FileImport.Inputline [Class Property]

Version: 2.x to 4.7
Availability: Text, Import

This property is used to temporary store the next text row imported from a CSV or XML file. This property is of no use to the developer.

6 FileImport.IsError [Class Property]

Version: 2.x to 4.7
Availability: Word, Excel, Text, Import

If an error exception occurs, either because of an internal error of the Integra script libraries or because of an error in any advanced Notes Formula or Lotus Script, the *IsError* property is set to True. At the same time the *property* contains the error message.

Using Lotus Script in the Script Error Handler (Integra profile) the profile developer can analyse the error message contained in [import.ErrorText](#) for specific errors. If the error handler can correct a particular error, the [import.ErrorText](#) property should be blanked ("") and the *IsError* property set to False.

7 FileImport.IsExecuteCallBack [Class Method (Function)]

Version: 2.x to 4.7
Availability: Word, Excel, Text, Import

IsExecuteCallBack (stats As StaticVars, status As Integer)

This *IsExecuteCallBack* method is called from various places within the runtime libraries to execute the advanced callback formula and script as defined in the Integra profile. The sequence is first Notes Formula then Lotus Script.

If any formula or script error occurs, the [Default advanced script structure explained](#) method is called.

Parameters:

stats (StaticVars)

Handle to Integra's runtime object [stats](#)

status (Integer)

Integra event with which the callback formula and or script is called with.

Return Value:

Returns the value of the `export.continue` property or `False` if any formula or script error occurred and no error callback script is defined.

8 FileImport.IsExecuteErrorHandle [Class Method (Function)]

Version: 2.x to 4.7
Availability: Word, Excel, Text, Import

IsExecuteErrorHandle (stats As StaticVars)

The *IsExecuteErrorHandle* method is called from the [Default advanced script structure explained](#) method if a formula or script error was triggered during the execution of the advanced formula and or script.

Parameters:

stats (StaticVars)

Handle to Integra's runtime object [stats](#)

Return Value:

Returns the value of the `export.continue` property, which is set to `False` before the execution of the error handle script.

9 FileImport.QuitWord [Class Method (Procedure)]

Version: 2.x to 4.7
Availability: Word, Import

This method removes the Word COM objects from memory after each Word document import is completed (closing any eventually still open Word files).

The method re-sets all Word environment variables to its former state if changes were required for Integra's processing.

10 FileImport.ReadExcelProperties [Class Method (Procedure)]

Version: 3.x to 4.7
Availability: Excel, Import

ReadExcelProperties (stats As staticvars)

This method reads values of the Excel properties into the internal value structure (`.`). Integra evaluates all formula fields as listed in the Selected Field List of the Integra profile and if the return value is of type string and the format of the string is `EXCEL.PROPERTIES.property`, it will write the Excel property value into the field's `import.record.value` index. Later on when Integra is writing the field values into the created Notes document, the values (Excel properties) will be written into the respective field names (as listed in the Selected Field List).

Please refer to the Integra profile to see the list of supported Excel properties.

Parameters:

stats (StaticVars)

Handle to Integra's runtime object [stats](#)

11 FileImport.ReadNext [Class Method (Function)]

Version: 2.x to 4.7
Availability: Word, Excel, Text, Import

ReadNext (stats As StaticVars)

Word

This method is called for every Word document that has been selected for import. The method creates the required Word COM objects and opens the to be imported Word document. It is then calling the `read` method to read in any Word property or Word field as defined in the Integra profile.

Excel

This method is called for every row in the to be imported Excel spreadsheet. The method will read in the values of each column cell in the current Excel row either by

- referring to the columns tag name (which is the field name as listed in the Selected Field List). This requires the Integra profiles *Use Tags* option to be set to *Yes*.
-
- or by calculating the column number using the index of the fields in the Selected Field List adding the *Column Offset* as defined in the Integra profile.
-
- **Note:** for the latter it is important that ALL fields in the Selected Field List which refer to an Excel property (EXCEL.PROPERTY.*property*) are listed after ALL column fields.

Text

This method is called for every record in the to be imported text file. A record is determined by the Line Separator field in the Integra profile. For CSV formatted files this is the carriage return character.

Note: XML import is not supported

Parameters:

stats (StaticVars)

Handle to Integra's runtime object [stats](#)

Return Value:

Returns *True* if no error exceptions occurred (e.g. triggered by field formulae), otherwise *False*

12 FileImport.ReadWordProperties [Class Method (Procedure)]

Version: 3.x to 4.7
Availability: Word, Import

ReadWordProperties (stats As staticvars)

This method reads values of the Word properties into the internal value structure (). Integra evaluates all formula fields as listed in the Selected Field List of the Integra profile and if the return value is of type string and the format of the string is WORD.PROPERTIES.*property*, it will write the Word property value into the field's import.record.value index. Later on when Integra is writing the field values into the created Notes document, the values (Word properties) will be written into the respective field names (as listed in the Selected Field List).

Please refer to the Integra profile to see the list of supported Word properties.

Release 4.0 +

Since release 4.0 this method also reads in values of Word field of type mailmerge (wdFieldMergeField). Same applies as for the above explained Word properties, however, the field formula must return a string in the format WORD.FIELD.*fieldname*(where *fieldname* refers to a Word mailmerge field).

Note: only formula (computed) fields are allowed in the context of Word import profiles. Any return value (of a listed formula field) that does not conform to the above described formats, will be written to the respective Notes field as is. Hence, formula fields can be used to create Notes fields with values not referring to the imported Word document.

Parameters:

stats (StaticVars)

Handle to Integra's runtime object [stats](#)

13 FileImport.Record [Class Property]

Version: 2.x to 4.7
Availability: Word, Excel, Text, Import

The Record property contains an object handle to the class object.

14 FileImport.SaveAndUpdateEmbedded [Class Method (Procedure)]

Version: 2.x to 4.7
Availability: Word, Import

SaveAndUpdateEmbedded(stats As StaticVars)

If the option (save as) Embedded Object is selected in the Integra profile, the embedded object requires some additional settings. This method sets the following two properties of the embedded object:

FitBelowFields = False
FitToWindow = True

Parameters:

stats (StaticVars)

Handle to Integra's runtime object

15 FileImport.TimeStamp [Class Property]

Version: 2.x to 4.7
Availability: Word, Excel, Text, Import

The TimeStamp property contains the data and time from when the Integra profile was initiated. The TimeStamp property can be used to flag imported Notes documents that are processed during as single profile execution with a common date-time stamp.

16 FileImport.WriteToDoc [Class Method (Procedure)]

Version: 2.x to 4.7
Availability: Word, Excel, Text, Import

ReadNext (stats As StaticVars)

Word

This method is called for every Word document that has been selected for import. The method writes all read in Word properties and fields into the (new) Notes document and then searches all formula fields in the Selected Field List for the reserved macro "WORD.RICHTEXT". If found, Integra will depending on the profile settings attach or embed the Word document into the respective field of the Notes document (containing the reserved macro) or import the Word content as rich text. For the latter Integra provides two methods: using the undocumented Notes API (default) or a UI driven Copy & Paste method ().

Excel

For Excel profiles the method simply writes all read-in values in the respective Notes document fields.

Text

For Text profiles the method simply writes all read-in values in the respective Notes document fields.

Parameters:

stats (StaticVars)

Handle to Integra's runtime object *stats*

doc (NotesDocument)

Handle to the Notes document that has been created for the currently imported Word document, Excel row or Text record. Integra passes on the document handle stored in *stats.ImpDoc*.