


FlatFile

Stereotype Attributes

Flat File Adapter

Attribute	Description	Allowed Values
alias	Specify the File alias the adapter should use to establish the connection.	any valid File alias
action	<div> The adapter action derives from the used operation. Do not configure this.</div>	

Flat File (Class)

A class having this stereotype is the root record of the flat file definition. This class cannot have any properties, only associations to **FlatFileRecord** classes and **FlatFileGroup** classes are allowed.

Attribute	Description	Allowed Values			Example
recordSeparator	Separator of the different records, normally line feed and carriage return. For serialized files, any other character can be defined.	any character or one of		C S y n t a x	<newline>
		<esc>		\ x 1B	
		<newline>	Parses correctly on Unix and Windows platforms.	\n	10
		<space>			32
		<tab>		\t	9
		<tabulator>			
		<unixnewline>	Composes newline.	\n	10
escapeCharacter	Defines the character used for escaping when a reserved character is used within a field value.	any character			/
quoteCharacter	The quoteCharacter will be ignored by reading field value.	any character			/
fillCharacter	Defines a dummy character to fill non-existent values (results in NULL). Used for fixed property layout only.	any character			0
reservedCharacters	Defines a list of characters to be escaped automatically when the file is composed.	any character			{ " / " , " % " , " & " , " (" , ") " }

On this Page:

- Stereotype Attributes
 - Flat File Adapter
 - Flat File (Class)
 - Flat File Complex Attribute (Class)
 - Flat File Group (Class)
 - Flat File Record (Class)
 - Flat File Attribute (Property)
 - Fixed Layout
 - Separated Layout
 - Flat File Sub Record (Property)
- Flat File Adapter Operations
 - compose Operation
 - parse Operation
- Parameter Types

FlatFileAdapter_ProductExport_Example



Click the icon to download a simple example model that shows the usage of the Flat File adapter in **Scheer PAS Designer**.

FlatFileAdapter_Hierarchical_Example



Click the icon to download a simple example model that shows how to compose a hierarchical flat file with the Flat File adapter in **Scheer PAS Designer**.

FlatFileAdapter_Quotes_Example



Click the icon to download a simple example model that shows the usage of a quote character with the Flat File adapter in **Scheer PAS Designer**.

composeEmptyAttributes	Controls whether empty trailing attributes of data segments will be written during composition.	true/false	
-------------------------------	---	------------	--

Related Pages:

- [Charset Definitions](#)
- [Number Formatting](#)
- [Date and Time Formatting Patterns](#)

Flat File Complex Attribute (Class)

This class is used to divide fields into sub-fields. You can think of this like a FlatFileRecord placed within a single field.

Attribute	Description	Allowed Values		Example
attributeLayout	<p>Defines the attribute layout (fixed or separated) of the complex attribute.</p> <ul style="list-style-type: none"> • Fixed: For property values that are shorter than the maximum length of the field, the field is filled with a fill character. As per default, this is blank space, but you can change the the fill character in attribute fill Character of the FlatFile class (see Attributes of Class FlatFile). • Separated: If separated is used, specify the separator using attributeSeparator (see below). 	fixed	Fixed property layout.	
		separated	Separated property layout.	
attributePattern	A RegEx pattern to parse the field content into a complex structure using capture groups.	any valid regular expression	<code>^(.(?)(: /))([A-Za-z0-9.])(: [0-9])(/.)\$</code>	
attributeSeparator	Defines the property/field separator.	any character	Use the specified character as property separator.	
		<Tab>	Use tabulator as property separator.	
		comma (,)	Use the comma (,) as property separator.	
suppressEscaping	Boolean value to suppress escaping.	true	Property values of this property will not be un-escaped (parser) or escaped (composer) (default).	
		false	Escaping/un-escaping is not suppressed.	
composeMacro	A macro that is executed while parsing/composing a file or complex field.	any valid macro expression (see Using Macro Expressions on Parsing or Composing a Flat File)		GetCounter(0)
parseMacro	<p>This macro can contain multiple commands separated by commas or spaces. Macros on classes are executed before the processing of its properties or associations. The ID represents a counter.</p> <p>The following counters are available:</p> <ul style="list-style-type: none"> • eight automatic counters with ID AUTO0 .. AUTO7 • two automatic line counters with ID LINE0 and LINE1 (parsing only) • unlimited custom counters with ID CUSTOM0 .. CUSTOMx <p>Automatic counters are increased by 1 for each processed record. Custom counters have to be increased manually using the increase macro. All counters have the initial value of 0 when they process the first record.</p> <p>For more details on macro commands see Using Macro Expressions on Parsing or Composing a Flat File.</p>			

Flat File Group (Class)


This class is used to group multiple records into one (virtual) structure. A group does not have a representation in the flat file itself.

Attribute	Description	Allowed Values	Example
composeMacro	A macro that is executed while parsing/composing a file or complex field.	any valid macro expression (see Using Macro Expressions on Parsing or Composing a Flat File)	GetCounter(AUTO0)
parseMacro	<p>This macro can contain multiple commands separated by commas or spaces. Macros on classes are executed before the processing of its properties or associations. The ID represents a counter.</p> <p>The following counters are available:</p> <ul style="list-style-type: none"> eight automatic counters with ID AUTO0 .. AUTO7 two automatic line counters with ID LINE0 and LINE1 (parsing only) unlimited custom counters with ID CUSTOM0 .. CUSTOMx <p>Automatic counters are increased by 1 for each processed record. Custom counters have to be increased manually using the increase macro. All counters have the initial value of 0 when they process the first record.</p> <p>For more details on macro commands see Using Macro Expressions on Parsing or Composing a Flat File.</p>		
evaluationOrder	Defines the order in which the association of the classes starting on same parent class must be processed.	any integer	
pattern	A pattern to identify the record. The pattern is checked before the fields are separated. If no pattern is defined, all records will be parsed.	any character	^Pattern.*
		a valid regular expression	

Flat File Record (Class)

A class having this stereotype describes the properties of a flat file record. A record is one "line" in a file and can consist of multiple properties.

Attribute	Description	Allowed Values		Example
pattern	A pattern to identify the record. The pattern is checked before the fields are separated. If no pattern is defined, all records will be parsed.	a valid regular expression		^Pattern.*
attributeLayout	Defines the property layout of the flat file record.	fixed	Fixed property layout. For property values that are shorter than the maximum length of the field, the field is filled with a fill character. As per default, this is blank space, but you can change the fill character in attribute fillCharacter of the FlatFile class (see the Attributes of the FlatFile class above).	
		separated	Separated property layout. Specify the separator using attributeSeparator (see below).	
		pattern	Property layout is specified by a RegEx pattern in attributePattern (see below).	
attributePattern	A RegEx pattern to parse the record content into the properties using capture groups.	a valid regular expression		^(.?) (:/) ([A-Za-z0-9.]) ([0-9]) (/.)\$
attributeSeparator	Defines the property/field separator.	any character	Use the specified character as property separator.	

		C o m m a (,)	Use the comma (,) as property separator.	
		< T a b>	Use tabulator as property separator.	
ignoreEmptyRecords	Boolean value for ignoring empty records. If set to true, no item will be generated, if none of the defined properties or sub records have any content. <div> Note, that a record containing only empty Strings is not empty – in opposition to a record composed from NULLS. See ignoreEmptyStrings below to skip processing of records containing only empty Strings.</div>	true	Ignore empty records.	
		false	Process empty records.	
suppressEscaping	Boolean value to suppress escaping. If suppressEscaping on a FlatFileRecord is true, FlatFileComplexAttribute that are part of this record will inherit this setting.	true	Property values of this record will not be un-escaped (parser) or escaped (composer) (default).	
		false	Escaping/un-escaping is not suppressed.	
composeMacro	A macro that is executed while parsing /composing a file or complex field.		any valid macro expression (see Using Macro Expressions on Parsing or Composing a Flat File)	GetCounter(AUTO0)
parseMacro	This macro can contain multiple commands separated by commas or spaces. Macros on classes are executed before the processing of its properties or associations. The ID represents a counter. The following counters are available: <ul style="list-style-type: none"> • eight automatic counters with ID AUTO0 .. AUTO7 • two automatic line counters with ID LINE0 and LINE1 (parsing only) • unlimited custom counters with ID CUSTOM0 .. CUSTOMx Automatic counters are increased by 1 for each processed record. Custom counters have to be increased manually using the increase macro. All counters have the initial value of 0 when they process the first record. For more details on macro commands see Using Macro Expressions on Parsing or Composing a Flat File .			
lineNumber	Specifies the number of a record in the file. The first record is lineNumber=1, the second lineNumber=2, etc.		any integer	
evaluationOrder	Defines the order in which the association of the classes starting on same parent class must be processed.		any integer	
ignoreEmptyStrings	Boolean value for ignoring empty string properties. If set to true , empty string values will be processed to NULL . Use this attribute in combination with ignoreEmptyRecords to skip processing of records containing only empty Strings .	true	Ignore empty string values.	
		false	Preserve empty string values.	

Flat File Attribute (Property)

Fixed Layout

Attribute	Description	Allowed Values	Example
-----------	-------------	----------------	---------

suppressEscaping	Boolean value to suppress escaping.	true	Property values of this property will not be un-escaped (parser) or escaped (composer) (default).	
		false	Escaping/un-escaping is not suppressed.	
parseMacro composeMacro	<p>A macro that is executed while parsing/composing a file or complex field.</p> <p>This macro can contain multiple commands separated by commas or spaces. Macros on classes are executed before the processing of its properties or associations. The ID represents a counter.</p> <p>The following counters are available:</p> <ul style="list-style-type: none"> • eight automatic counters with ID AUTO0 .. AUTO7 • two automatic line counters with ID LINE0 and LINE1 (parsing only) • unlimited custom counters with ID CUSTOM0 .. CUSTOMx <p>Automatic counters are increased by 1 for each processed record. Custom counters have to be increased manually using the increase macro. All counters have the initial value of 0 when they process the first record.</p> <p>For more details on macro commands see Using Macro Expressions on Parsing or Composing a Flat File.</p>	any valid macro expression (see Using Macro Expressions on Parsing or Composing a Flat File)		GetCounter(0)
padding	Defines the padding rule for the field from the left or right side.	<p>left ("<any character")</p> <p>right ("<any character")</p>	<ul style="list-style-type: none"> • Parsing: Ignore the specified character from the left/right side to the first different character. • Composing: Fill the property from the left /right side to the first different character using the specified character. 	<p>left("0")</p> <p>right(" ")</p>
format	Pattern for formatting numeric and date & time values. For details see Number Formatting respectively Date and Time Formatting Patterns .	any valid number or dateTime pattern		<p>S9G999G990D00</p> <p>%Y.%m.%d-%H:%M:%S</p>
order	The evaluation order of the properties. If offset is not used, order reflects the field number within the record.	any integer		
offset	The character position of this field within the record.	any integer		
externalLength	Number of characters of the field (only for fixed length records relevant).	any integer		

Separated Layout

Attribute	Description	Allowed Values	Example
suppressEscaping	Boolean value to suppress escaping.		
parseMacro	<p>A macro that is executed while parsing/composing a file or complex field.</p> <p>This macro can contain multiple commands separated by commas or spaces. Macros on classes are executed before the processing of its properties or associations. The ID represents a counter.</p> <p>The following counters are available:</p> <ul style="list-style-type: none"> • eight automatic counters with ID AUTO0 .. AUTO7 • two automatic line counters with ID LINE0 and LINE1 (parsing only) • unlimited custom counters with ID CUSTOM0 .. CUSTOMx <p>Automatic counters are increased by 1 for each processed record. Custom counters have to be increased manually using the increase macro. All counters have the initial value of 0 when they process the first record.</p>	any valid macro expression (see Using Macro Expressions on Parsing or Composing a Flat File)	GetCounter(0)

composeMacro	For more details on macro commands see Using Macro Expressions on Parsing or Composing a Flat File .		
format	Pattern for formatting numeric and date & time values. For details see Number Formatting respectively Date and Time Formatting Patterns .		
order	The evaluation order of the properties. If offset is not used, order reflects the field number within the record.	any integer	
offset	The relative position of the field in respect of the other fields in the record, e.g. field number 3 has offset = 2.	any integer	

Flat File Sub Record (Property)

Attribute	Description	Allowed Values	Example
condition	A condition that must evaluate <i>true</i> if the record exists. The condition can refer to a self object which represents the current state of the parent.	any valid conditional expression	<code>self.UNS.exists()</code>
evaluation Order	Defines the order in which the associations starting on same parent class must be processed.	any integer	
offset	Define the position of this record in the flat file, starting with 0 for the first record and always relative to the parent element.	any integer	

Flat File Adapter Operations

compose Operation

Name	Type	Direction	Mandatory	Description	Allowed Values	Example
anyObjectFlow	Any with Flat File class stereotype	in	✓	Provide an object containing the flat file data. The class defining the type of this object should have stereotype FlatFile and should depict the structure of the file.		
encoding	String	in		Provide the encoding of the file to be composed as specified on Charset Definitions .	any valid encoding (see Charset Definitions)	UTF-8
					default ISO-8859-1 (Latin1)	
locale	Number sLocale	in		Specify how number values will be treated, when written to the flat file (decimal point, currency symbol, ...). You can overwrite the system locales here. Refer to Number Formatting for more information.	default system locales	
name	String	in	✓	Specify a full path to the flat file, if you want to write the FlatFile object to the file system. Alternatively, you can compose the flat file to a Blob object (see parameter data). Note, that the name parameter takes priority over data .		tmp/myFlatFile.txt
data	Blob	out	✓	If you want to compose the FlatFile object to a Blob object, use this parameter as output of the compose action. Alternatively, you can write the composed flat file directly to the file system (see parameter name). Note, that the name parameter takes priority over data .		



If you provide both parameters, **name** and **data**, the Flat File object will be written to the file system.

parse Operation

Name	Type	Direction	Mandatory	Description	Allowed Values	Example
data	Blob	in	(✓)	Provide the flat file data to be parsed. Alternatively, you can specify a path to a flat file in the file system (see parameter name). Note, that the name parameter takes priority over data .		
encoding	String	in		Provide the encoding of the file to be parsed as specified on Charset Definitions .	any valid encoding (see Chars et Definitions) d e f a u l t ISO-8859-1 (Latin1)	UTF-8
locale	NumbersLocale	in		Specify how number values will be treated, when parsed from the flat file (decimal point, currency symbol, ...). You can overwrite the system locales here, if the file was written with divergent locales. Refer to Number Formatting for more information.		
name	String	in	(✓)	Specify a full path to the flat file to be parsed. Alternatively, you can parse the flat file from a Blob object (see parameter data). Note, that the name parameter takes priority over data .		tmp/myFile.txt
anyObjectFlow	Any with FlatFile class stereotype	out	(✓)	The adapter returns a parsed flat file object. The class defining the type of this object should have stereotype FlatFile and should depict the structure of the file.		



If you provide both parameters, **name** and **data**, the Flat File object will be parsed from the file system.

Parameter Types

Class	Attribute	Type	Description
NumbersLocale	negativeSign	String	Characters used to signify negative values. Usually '-'.
	positiveSign	String	Characters used to signify positive values. Usually '+'.
	thousandsSeparator	String	Characters used to separate units of thousand, e.g. ','.
	decimalPoint	String	e.g. '.'
	currencySymbol	String	e.g. '\$'