

MiscMergeTemplate

Inherits From: Object
Declared In: misckit/MiscMergeTemplate.h

Class Description

This class conatins the template that is used by a merge engine. It performs two functions: (1) parse a string or text file into the commands required by a merge ending and (2) act as a container for the commands once they have been parsed, providing them to a merge engine as needed.

Typically, MiscMergeTemplate objects are used in a very simple way: they are instantiated, given the ASCII text or string to parse, and then passed to MiscMergeEngine instances as needed. That's it!

It should be noted that template text which is simply copied from the template into the merged output (ie. an text outside of a merge command) is actually turned into a special `^copy^` command by the parsing algorithm. This allows the merge engine to deal exclusively with MiscMergeCommand subclasses to perform a merge. This implementation detail should not affect anything that would normally be done with this object, but it is important to understand this fact if attempting to understand the data structure created by the parsing routines.

Instance Variables

List ***commands**;

commands

Storage for parsed MiscMergeCommands.

Method Types

Handling delimiters

- + startFieldDelimiter
- + endFieldDelimiter
- + setFieldDelimitersStart:end:

Finding MiscMergeCommands

- + getClassForCommand:

Initializing a MiscTemplate

- init
- initWithString:
- initFromFile:
- initFromFileNamed:

Loading the template

- parseFromString:
- parseFromFile:
- parseFromFileNamed:

Accessing parsed template

- commands
- commandAt:

Class Methods

endFieldDelimiter

- + (char)**endFieldDelimiter**

Returns the character used to end a merge command.

getClassForCommand:

- + **getClassForCommand:**(MiscString *)*aCommand*

Given the command string *aCommand*, this method determines which MiscMergeCommand subclass implements the merge command. It returns the class object needed to create instances of the MiscMergeCommand subclass.

This method works by asking the runtime if it can find Objective-C classes with specific names. The name that is looked up is build from the first word found in *aCommand*. The first word is turned to all lower case, with the first letter upper case, and then sandwiched between `^Merge^` and `^Command^`. For example, the merge command `^«If xxx = y»^` has the word `^If^` as the first word. Thus, the class `^MergelfCommand^` will be searched for. If the desired class cannot be found, then it is assumed that the merge command is giving the name of a field which should be inserted into the output document.

To avoid name space conflicts, all internal merge commands actually use a slightly different name. Thus, there really is no `^MergelfCommand^` to be found. This method, when it doesn't find the `^MergelfCommand^` class, will search for another class, with a private name. That class will be found. (If it weren't found, then the default `^field^` command class would be returned.) This allows a programmer to override any built in command. To override the `^if^` command, simply create a `^MergelfCommand^` class and it will be found before the built in class. If a programmer wishes to make a particular command, such as `^omit^`, inoperative, this technique may be used to override with a MiscMergeCommand subclass that does nothing.'

Note that if the command string *aCommand* contains merge commands inside itself, then a special `^delayed command^` class will be returned. That class will, during a merge, create an engine, perform a merge on its text, and then parse itself into the correct type of command. This allows merges to contain commands that change depending upon the data records.

Note: This method should really be in the MiscMergeCommand class and not here. Sorry! I may move it when I have the time. **-don**

setFieldDelimitersStart:end:

+ **setFieldDelimitersStart:**(char)*aChar1* **end:**(char)*aChar2*

Sets the characters used to start and end a merge command. Returns **self**. It is highly recommended that *aChar1* and *aChar2* be different characters.

startFieldDelimiter

+ (char)**startFieldDelimiter**

Returns the character used to start a merge command.

Instance Methods

commandAt:

- **commandAt:**(unsigned)*index*

Returns the *ith* merge command to be executed by the merge engine.

commands

- **commands**

Returns a List object containing, in order, all the merge commands to be executed by the merge engine.

init

- **init**

Initializes the MiscMergeTemplate instance and returns **self**. This is the designated initializer.

initWithFile:

- **initWithFile:**(MiscFile *)*aFile*

Initializes the MiscMergeTemplate instance, parses the template from the file pointed to by *aFile*, and returns **self**. This method is provided for convenience.

initWithFileName:

- **initWithFileName:**(MiscString *)*aFileName*

Initializes the MiscMergeTemplate instance, parses the template from the file named *aFileName*, and returns **self**. This method is provided for convenience.

initWithString:

- **initWithString:**(MiscString *)*aString*

Initializes the MiscMergeTemplate instance, parses the template from *aString*, and returns **self**. This method is provided for convenience.

parseFromFile:

- **parseFromFile:**(MiscFile *)*aFile*

Parses the MiscMergeTemplate from the file pointed to by *aFileName*. Returns **self**.

parseFromFileNamed:

- **parseFromFileNamed:**(MiscString *)*aFileName*

Parses the MiscMergeTemplate from the file named *aFileName*. Returns **self**.

parseFromString:

- **parseFromString:**(MiscString *)*aString*

Parses the MiscMergeTemplate from *aString*. Returns **self**.