MiscMergeCommand

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

Class Description

The MiscMergeCommand class implements a merge command. Since the MiscKit merge engine can dynamically add new commands, it is possible to create custom subclasses of MiscMergeCommand to implement new functionality in the engine. The merge engine already implements most of the commands a user would want, but certain applications may wish to override those commands, replace them, or add new commands.

To create a subclass, two methods, -**parseFromString:** and -**executeForMerge:** need to be implemented. The first is expected to break up a text string into whatever arguments a particular MiscMergeCommand subclass needs in order to function. The second performs, during a merge, whatever special task the command is designed to do.

The other methods in this object may be used by subclasses to aid in parsing the command string. They can grab key words, conditional operators, an arguments (single word or quoted string). A special kind of argument, promptable, is also supported. A promptable argument is expected to have its actual value determined at run time.

When implementing commands, the full API of the MiscMergeEngine object is available. This allows the programmer to store information in the engine, manipulate the symbol tables used for resolving fields, and alter the output being created by the merge.

The MiscKit source code is a good place to look for examples of how to implement various MiscMergeCommand subclasses.

Method Types

Initializing	- initFrom:
Subclass responsibilities	- parseFromString: - executeForMerge:
Methods to aid in parsing	- eatKeyWord:from:isOptional: - getArgumentStringFrom:toEnd: - getPromptFrom:toEnd: - getPromptableArgumentStringFrom:wasPrompt:toEnd: - getConditionalFrom:
Errors subclasses may trap	- error_conditional: - error_keyword: - error_noprompt - error_closequote
Finding conditional results	+ evaluateConditionWith:for:

Class Methods

evaluateConditionWith:for:

+ (BOOL)**evaluateConditionWith:**(MiscMergeEngine *)*anEngine* **for:***aCommand*

Evaluates the condition in *aCommand* in the context of the merge in progress in *anEngine*. Returns YES if the condition evaluated true and NO if not. If *aCommand* doesn't implement the MiscMergeCondCallback protocol, you'll get a NO back automatically. This method should be used by MiscMergeCommand subclasses that need to evaluate conditionals as part of their task, such as the "if" command.

Instance Methods

eatKeyWord:from:isOptional:

- (BOOL)**eatKeyWord:**(MiscString *)*aKeyWord* **from:**(MiscString *)*aString* **isOptional:**(BOOL)*flag*

Attempts to remove the contents of *aKeyWord* from *aString*. If *flag* is YES, then no complaint will be made if *aKeyWord* is missing. YES or NO is returned to tell the caller if the required key word was found or not, no matter what the value of *flag* was. Note that it *aKeyWord* is optional, then it will only be found if *aString* has no whitespace at the beginning. This is a design decision, since the merger has a few pseudo-commands, such as copy, which could give a less desireable behavior if the whitespace were trimmed from the start of *aString*. Since user-implemented MiscMergeCommand subclasses should pass YES for *flag* this subtle design decision should not have a major impact.

error_closequote

- (void)**error_closequote**

This method is called if, while parsing, it is discovered that quotations are not matched up properly. Prints the name of the merge command class and the text ªClosing quote missing or spurious extra argument added.º to the console.

error_conditional:

- (void)**error_conditional:**(MiscString *)*theCond*

This method is called if, while parsing, it is discovered that a conditional is unrecognized. Prints the name of the merge command class, the text ªUnrecognized conditional:º, and the -**stringValue** of *theCond* to the console.

error_keyword:

- (void)**error_keyword:**(MiscString *)*aKeyWord*

This method is called if, while parsing, it is discovered that a required key word is missing. Prints the name of the merge command class, the text ªMissing key word:º, and the -**stringValue** of *aKeyWord* to the console.

error_noprompt

- (void)**error_noprompt**

This method is called if, while parsing, it is discovered that the required prompt is missing. (Referring to arguments that are promptable.) Prints the name of the merge command class and the text ªMissing prompt.º to the console.

executeForMerge:

- **executeForMerge:**(MiscMergeEngine *)*aMerger*

This method is called by the merge engine while it is performing a merge. The command is expected to perform it's specified function when this call is received. Returns **self**. The return value is currently ignored by the caller, which is usually, but does not have to be, *aMerger*.

getArgumentStringFrom:toEnd:

- **getArgumentStringFrom:**(MiscString *)*aString* **toEnd:**(BOOL)*endFlag*

Attempts to parse an argument from *aString*. If *endFlag* is set, then whatever is found to the end of *aString* will be assumend to be the required argument. Otherwise, if an argument contains whitespace, it should be surrounded by quotation marks ª"º. The parsed argument will be removed from *aString*.

getConditionalFrom:

- (MISC_Merge_Cond_Op)**getConditionalFrom:**(MiscString *)*aString*

Attempts to parse a conditional from *aString*. Currently recognized conditionals are: <>, ><, !=, <=, =<, >=, =>, <, >, ==, =. Returns the type of conditional found or MISC_MCO_NONE if an unrecognized conditional is found. Removes the parsed conditional from *aString*.

getPromptFrom:toEnd:

- **getPromptFrom:**(MiscString *)*aString* **toEnd:**(BOOL)*endFlag*

Attempts to parse a promptable argument from *aString*. If the argument begins with a "?" then the argument is a "prompt". Removes the parsed argument from *aString* and returns it. Returns **nil** if the wrong kind of argument was found. It is expected that a promptable argument's value will be determined at run time by asking the user for the value that should be stored for it.

getPromptableArgumentStringFrom:wasPrompt:toEnd:

- **getPromptableArgumentStringFrom:**(MiscString *)*aString* **wasPrompt:**(BOOL *)*prompt* **toEnd:**
  (BOOL)*endFlag*

Attempts to parse an argument, which could be promptable, from *aString*. If the argument begins with a "?" then the argument is a "*prompt*". Otherwise, a regular argument is parsed. Removes the parsed argument from *aString* and returns it. *prompt* is set to YES or NO depending upon what was parsed.

initFrom:

- **initFrom:**(MiscString *)*aString*

Initializes a new instance of MiscMergeCommand, based upon the text of *aString*. Actual parsing is performed by the -**parseFromString:** method. If parsing fails and returns **nil**, then this method will free the receiving instance and return **nil**. Returns **self** otherwise.

parseFromString:

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

This method is called while parsing a merge template. The text of the full merge command is contained in *aString*. This method should break *aString* up into keywords, conditionals, and arguments as needed and store the results in instance variables for later use during merges. Note that returning **self** tells the template parsing machinery that all is well. Return **nil** if there is an error or the command cannot be properly initialized. (But do *not* call -**free** on **self** if **nil** is returned!)