NSScanner (MiscRegex)

Declared In: <misckit/NSScanner+MiscRegex.h>

Category Description

This category extends the NSScanner class to provide support for regular expressions (using NSString's MiscRegex category).

As in the regular NSScanner methods, the starting scan location for these methods is the location after any characters in the skip set have been skipped over.

Like the NSString+MiscRegex category, all methods that take an NSString regex argument have a parallel version that takes a precompiled regex structure instead. This way, if several searches need to be done with the same regular expression, the expression only needs to be compiled once. Use the MiscNewRegexStruct() function to create the structure (see the documentation for the NSString+MiscRegex category).

NOTE: This category depends on the ranges returned by methods in the NSString+MiscRegex category, which operate on the -cString of the given string. Unfortunately, due to character translations and/or Unicode composed character sequences, these ranges might not match up with the characters they were intended to. If that happens, the scanner could get out of sync (the scanLocation and returned strings could be off) or exceptions could be raised (if the returned ranges are out of bounds for the normal string). Therefore, these methods should only be used on strings that have a 1-1 character mapping between itself and its -cString. See the documentation for the NSString+MiscRegex category for more information.

Method Types

Scanning with regular expressions:	- scanRegex:intoString: - scanRegexStruct:intoString: - scanUpToRegex:intoString: - scanUpToRegexStruct:intoString:

Instance Methods

scanRegex:intoString:

- (BOOL)**scanRegex:**(NSString *)*regex* **intoString:**(NSString **)*value*

scanRegexStruct:intoString:

- (BOOL)**scanRegexStruct:**(regexp_t)*pattern* **intoString:**(NSString **)*value*

Scans for a match of the expression *regex*, and if a match is found returns (by reference) in *value* a string object equal to the matched portion. If *regex* matches the characters at the scan location, returns YES; otherwise returns NO. Note that *value* might be empty when YES is returned -- that means that the expression matched a string of length zero at the scan location.

This method may be invoked with **NULL** as *value* to simply scan past a given regular expression.

See also: -scanUpToRegex:intoString:

scanUpToRegex:intoString:

- (BOOL)**scanUpToRegex:**(NSString *)*regex* **intoString:**(NSString **)*value*

scanUpToRegexStruct:intoString:

- (BOOL)**scanUpToRegexStruct:**(regexp_t)*pattern* **intoString:**(NSString **)*value*

Scans the string until the expression *regex* is matched, accumulating characters encountered into a string that's returned by reference in *value*.   The receiver's scan location will then be at the position of the match (or at the end of the string being scanned if *regex* isn't found).   If any characters are scanned, returns YES; otherwise returns NO.

This method may be invoked with **NULL** as *value* to simply scan up to a given regular expression.

See also: -scanRegex:intoString: