WebSphere Product Center

# Scripting Reference Guide

***Version 5.1***

Note!

Before using this information and the product it supports, read the information in “Notices” at the end of this document.

**24 September2004**

This edition of this document applies to WebSphere Product Center (5724-I68), version 5.1, and to all subsequent releases and modifications until otherwise indicated in new editions.

# Table of Contents



 

 

# Introduction

The following pages list the script operations that are included in WebSphere Product Center. A description and prototype is provided for each operation.

The prototypes use the following:

Return type - Object - Method - (Parameters)

**NOTE: The script operations listed in this document are for reference only and are subject to change. Refer to the Script Sandbox in the WebSphere Product Center product for more up-to-date information.**

This document covers the following sections:

- **WebSphere Product Center script operations** - The first section includes list of script operations that have been included in the WebSphere Product Center 5.0 Script Sandbox.
- **Updates to Script Operations** - The section includes additional updates to the script operations, which are not identified in the section "WebSphere Product Center Script Operatons". Therefore, it is important to reference both sections.
- **Script Types** – This section defines the script types that are available in WebSphere Product Center.

# WebSphere Product Center script operations

## Basic

## Array

### add

| | |
|---|---|
| Description | To add elements to an Array |
| Prototype | void Array::add(elements) |

### remove

| | |
|---|---|
| Description | To remove the element at the specified position |
| Prototype | void Array::remove(int i) |

### sort

| | |
|---|---|
| Description | Return the array sorted |
| Prototype | Array Array::sort() |

## Date

### addDate

| | |
|---|---|
| Description | Add the integer value given to the field specified. Allowed field values are : YEAR MONTH DATE HOUR MINUTE |
| Prototype | Date Date::addDate(String field, Integer value) |

### formatDate

| | |
|---|---|
| Description | Use to format a date as a human readable format. The newFormat string is a pattern whose format is identical to the format used by Java |
| Prototype | String Date::formatDate(String newFormat) |

### getDateField

| | |
|---|---|
| Description | Get the value of the field specified. Allowed field values are : YEAR MONTH DATE HOUR_OF_DAY MINUTE SECOND |

| | |
|---|---|
| Prototype | Integer Date::getDateField(String field) |

### getTime

| | |
|---|---|
| Description | Returns the number of seconds since January 1, 1970, 00:00:00 GMT represented by this Date object |
| Prototype | Integer Date::getTime() |

### isDateAfter

| | |
|---|---|
| Description | Returns true if and only if this date is after otherDate |
| Prototype | Boolean Date::isDateAfter(Date otherDate) |

### isDateBefore

| | |
|---|---|
| Description | Returns true if and only if this date is before otherDate |
| Prototype | Boolean Date::isDateBefore(Date otherDate) |

### **new**$Date

| | |
|---|---|
| Description | Builds a Date object from a String given a date format |
| Prototype | new Date(String sFormat, String sDate) |

### reformatDate

| | |
|---|---|
| Description | Takes a date string which is assumed to be formatted according to the pattern indicated by currentDateFormat, and returns a new string formatted according to the newDateFormat provided. If no newDateFormat is given, the WPC dateFormat is used. |
| Prototype | String reformatDate (String formattedDateString, String currentDateFormat [, String newDateFormat]) |

### setDateField

| | |
|---|---|
| Description | Return a Date equal to the input Date, except that the specified field is set to the given value. Allowed field values are : YEAR MONTH DATE HOUR_OF_DAY MINUTE SECOND |
| Prototype | Date Date::setDateField(String field, Integer value) |

| | |
|---|---|
| | value) |

### today

| | |
|---|---|
| Description | Returns the current date and time |
| Prototype | Date today () |

## HashMap

### containsKey

| | |
|---|---|
| Description | Returns true if key exists. |
| Prototype | Boolean HashMap::containsKey(Object key) |

### containsValue

| | |
|---|---|
| Description | Returns true if value exists. |
| Prototype | Boolean HashMap::containsValue(Object val) |

### forEachHmElement

| | |
|---|---|
| Description | Executes the statements for each (oKey, oValue) map in hm |
| Prototype | forEachHmElement(HashMap hm, Object oKey, Object oValue) { statements } |

### intersectValues

| | |
|---|---|
| Description | Return the set-intersection of hm1, hm2, ... (only values are considered) |
| Prototype | HashMap intersectValues(HashMap hm1, HashMap hm2, ...) |

### keyForValue

| | |
|---|---|
| Description | Returns a key mapped to valueToSearch in hm or null |
| Prototype | Object HashMap::keyForValue(Object valueToSearch) |

### mergeValues

| | |
|---|---|
| Description | Return the set-union of hm1, hm2, ... (only values are considered) |
| Prototype | HashMap mergeValues(HashMap hm1, HashMap hm2, ...) |

| | |
|---|---|
| | HashMap hm2, ...) |

### size

| | |
|---|---|
| Description | Returns the size of a HashMap or any array |
| Prototype | Integer HashMap::size () |

## LanguageConstruct

### break-continue

| | |
|---|---|
| Description | To break/continue from a loop |
| Prototype | [break\|continue] |

### catchError

| | |
|---|---|
| Description | Analagous to a try-catch in Java, all statements are executed and errMsg is set to null in the absence of errors |
| Prototype | catchError(String errMsg) { statements } |

### escapeForCSV

| | |
|---|---|
| Description | Escape for CSV |
| Prototype | String escapeForCSV(String s) |

### escapeForHTML

| | |
|---|---|
| Description | Escape for HTML |
| Prototype | String escapeForHTML(String s) |

### escapeForJS

| | |
|---|---|
| Description | Escape for JavaScript |
| Prototype | String escapeForJS(String s) |

### for

| | |
|---|---|
| Description | Equivalent to doing init-statement; while(cond) {t-statements; each-statement;} |
| Prototype | for(init-statement; cond; each-statement) { t-statements } |

### if-else

| | |
|---|---|
| Description | If cond is true t-statements are executed, otherwise f-statements are executed |
| Prototype | if(Boolean cond) { t-statements } [else { f-statements }] |

### logDebug

| | |
|---|---|
| Description | Logs the debug message with the debug log that is available from the schedule profile details screens. Use with caution because the debug log is maintained in memory. |
| Prototype | void logDebug (String message) |

### logError

| | |
|---|---|
| Description | Logs the error message with the corresponding item id to the location specified in the context |
| Prototype | void logError(String itemId, String message) |

### logWarning

| | |
|---|---|
| Description | Logs the warning message with the corresponding item id to the location specified in the context |
| Prototype | void logWarning(String itemId, String message) |

### return

| | |
|---|---|
| Description | Used in functions: returns e to the caller |
| Prototype | return e |

### throwError

| | |
|---|---|
| Description | Use to throw a Java-like exception. This operation is usually used in conjunction with the catchError operation |
| Prototype | void throwError (String rejectionCause) |

### useTransaction

| | |
|---|---|
| Description | Executes the statements in a transaction, rollback takes place if an error occurs |
| Prototype | useTransaction { statements } |

### while

| | |
|---|---|
| Description | As long as cond is true, t-statements are executed |
| Prototype | while(Boolean cond) { t-statements } |

## Numeric

### checkDouble

| | |
|---|---|
| Description | If the input string is null or empty, the default value is returned. Otherwise the original value parsed as an Double is returned. |
| Prototype | Double checkDouble(String str, Double defaultValue) |

### checkInt

| | |
|---|---|
| Description | If the input string is null or empty, the default value is returned. Otherwise the original value parsed as an Integer is returned. |
| Prototype | Integer checkInt(String str, Integer defaultValue) |

### max

| | |
|---|---|
| Description | Return the max |
| Prototype | Number max(Number a, Number b) |

### min

| | |
|---|---|
| Description | Return the min |
| Prototype | Number min(Number a, Number b) |

### rand

| | |
|---|---|
| Description | Returns a random integer that is between 0 and max |
| Prototype | Integer rand(Integer max) |

### reformatDouble

| | |
|---|---|
| Description | Returns a new String representing the number, reformatted to fit the criteria specified by minDigitsBeforeDecPoint and maxDigitsAfterDecPoint |
| Prototype | String reformatDouble (Double origDouble, Integer minDigitsBeforeDecPoint, Integer |

| | |
|---|---|
| | Integer minDigitsBeforeDecPoint, Integer maxDigitsAfterDecPoint |

### toDouble

| | |
|---|---|
| Description | Parses str as a Double |
| Prototype | Double toDouble(String str) |

### toInteger

| | |
|---|---|
| Description | Parses str as an Integer |
| Prototype | Integer toInteger(String str) |

## RegularExpression

### buildRE

| | |
|---|---|
| Description | Returns a regular expression corresponding to the given pattern. Match flags are 0=caseSensitive, 1=ignoreCase, 2=matchMultiline (new lines match as ^ and $, 4=matchSingleLine (treat multiple lines as one line). Flags are additive. |
| Prototype | new RE(String pattern, Integer matchFlags |

### match

| | |
|---|---|
| Description | Return the contents of the parenthesized subexpressions after a successful match |
| Prototype | String[] RE::match(String str) |

### new$RE

| | |
|---|---|
| Description | Returns a regular expression corresponding to the given pattern. Optional match flags are 0=caseSensitive, 1=ignoreCase, 2=matchMultiline (new lines match as ^ and $, 4=matchSingleLine (treat multiple lines as one line). Flags are additive. |
| Prototype | new RE(String pattern, Integer matchFlags ) |

### substitute

| | |
|---|---|
| Description | Return substituteIn with zero or more occurences of the regular expression specified in the RE |

| | |
|---|---|
| | object replaced with the substitution string |
| Prototype | String[] RE::substitute(String substituteIn, String substitution) |

# ScriptObject

### getFunctionByName

| | |
|---|---|
| Description | Build the function object for the function sFunctionName in this script object |
| Prototype | FunctionObject ScriptObject::getFunctionByName(String sFunctionName) |

### getScriptByPath

| | |
|---|---|
| Description | Build the script object for the script stored at sScriptPath |
| Prototype | ScriptObject getScriptByPath(String sScriptPath) |

### getScriptContextValue

| | |
|---|---|
| Description | Return the value of the variable named sVariableName |
| Prototype | Object getScriptContextValue(String sVariableName) |

### invoke

| | |
|---|---|
| Description | Invoke this function object with the arguments arg1, arg2, etc |
| Prototype | Object FunctionObject::invoke(Object arg1, Object arg2, etc) |

### invokeSoapServer

| | |
|---|---|
| Description | Invoke a soap server. SURL is the URL of the service. SMETHODNAME is the name of the operation called. APARAMVALUES is an array containing the request parameters. APARAMNAMES is an optional array containing the names of the paramters. Returns the return value of the SOAP operation call. |
| Prototype | Object invokeSoapServer(String sURL, String sMethodName, Object[] aParamValues [,String[] |

| | |
|---|---|
| | aParamNames] ) |

### runScript

| | |
|---|---|
| Description | Run this script |
| Prototype | void ScriptObject::runScript(HashMap hmContext) |

### setScriptContextValue

| | |
|---|---|
| Description | Set the value of the variable named sVariableName |
| Prototype | void setScriptContextValue(String sVariableName, Object oVariableValue) |

## Scripting

### setScriptProgress

| | |
|---|---|
| Description | Sets the percentage completed value in the context of a running script |
| Prototype | setScriptProgress(number percent) |

### setScriptStatsDeletedCnt

| | |
|---|---|
| Description | Sets the count of items deleted in the context of a running script |
| Prototype | setScriptStatsDeletedCnt(number count) |

## String

### buildCSV

| | |
|---|---|
| Description | Takes a variable number of arguments, and returns a string with the arguments concatenated in csv format |
| Prototype | String buildCSV (String str1, String str2, ..., String strN) |

### buildDelim

| | |
|---|---|
| Description | Takes a variable number of arguments, and returns a string with the arguments concatenated in delim format, using the qualifier to enclose |

| | |
|---|---|
| | strings that contain the delimiter |
| Prototype | String buildDelim (String delimiter, String qualifier, String str1, String str2, ..., String strN) |

### buildFixedWidth

| | |
|---|---|
| Description | Takes a variable number of arguments, and returns a string with the arguments concatenated in fixed width format. |
| Prototype | String buildFixedWidth (String str1, Integer len1, String strN, Integer lenN) |

### checkString

| | |
|---|---|
| Description | If the input string is null or empty, the default value is returned. Otherwise the original value itself is returned. |
| Prototype | String checkString (String str, String defaultValue) |

### concat

| | |
|---|---|
| Description | Takes a variable number of arguments, and returns a string with the arguments concatenated in the order given |
| Prototype | String concat (String str1, String str2, ..., String strN) |

### contains

| | |
|---|---|
| Description | Tests if this string contains an occurence of the match substring |
| Prototype | Boolean String::contains (String match) |

### containsUsingLookupTable

| | |
|---|---|
| Description | Return true if and only if the string contains at least one of the keys from the lookup table |
| Prototype | Boolean String::containsUsingLookupTable(LookupTable lkp) |

### endsWith

| | |
|---|---|
| Description | Tests if this string ends with an occurence of the match substring |

| | |
|---|---|
| Prototype | Boolean String::endsWith (String match) |

### escapeWithHTMLEntities

| | |
|---|---|
| Description | Translates all character with HTML character codes less than beg or greater than end to HTML character codes |
| Prototype | String escapeWithHTMLEntities(String str, Integer beg, Integer end) |

### getLoginString

| | |
|---|---|
| Description | Returns the url string needed for login automatically to the given url as the current user. If you are an admin, you can generate a login string for another user by passing the username as an extra parameter. Note that the url should not include the server name/port and should start with '/'. If an error occurs, a null string is returned. |
| Prototype | String getLoginString(String sUrl, Date dExpirationDate, [String sUserName]) |

### getNameFromPath

| | |
|---|---|
| Description | if str contains / returns the substring of str after the last / char exclusively, otherwise returns the original string |
| Prototype | String getNameFromPath(String str) |

### getPageURL

| | |
|---|---|
| Description | Return the URL for the page requested given the required object. The required objects are either: an Item, ItemList, or Category |
| Prototype | String getPageURL(requiredObject) |

### getParentPath

| | |
|---|---|
| Description | if str contains / returns the substring of str up to the last / char exclusively, otherwise returns the empty string |
| Prototype | String getParentPath(String str) |

### getRidOfRootName

| | |
|---|---|
| Description | If str contains / gets rid of all preceding first / inclusive |

| | |
|---|---|
| | inclusive |
| Prototype | String getRidOfRootName(String str) |

### getMemorySummary

| | |
|---|---|
| Description | Invokes the garbage collector, sleeps for 5 seconds and then returns a string summarizing memory usage. |
| Prototype | String getMemorySummary() |

### indexOf

| | |
|---|---|
| Description | Returns the index within this string of the first occurrence of the specified match substring |
| Prototype | Integer String::indexOf (String match) |

### isLowerCase

| | |
|---|---|
| Description | Checks if all the characters in this string are lower case using the rules of the default locale |
| Prototype | Boolean String::isLowerCase () |

### isStringSingleByte

| | |
|---|---|
| Description | Returns true if the string is made of single byte characters only, false otherwise |
| Prototype | Boolean isStringSingleByte(String s) |

### isUpperCase

| | |
|---|---|
| Description | Checks if all the characters in this string are upper case using the rules of the default locale |
| Prototype | Boolean String::isUpperCase () |

### lastIndexOf

| | |
|---|---|
| Description | Returns the index within this string of the rightmost occurrence of the specified match substring |
| Prototype | Integer String::lastIndexOf (String match) |

### length

| | |
|---|---|
| Description | Returns the length of this string |
| Prototype | Integer String::length () |

### newLookupTable

| | |
|---|---|
| Description | Returns a new lookup table with the given spec and name. |
| Prototype | newLookupTable(Spec spec, String name) |

### parseCSV

| | |
|---|---|
| Description | Returns an array of each token, as parsed by the CSV parser. If a field number is provided, just the corresponding token substring is returned. |
| Prototype | String[] String::parseCSV () \| String String::parserCSV(Integer field) |

### parseDelim

| | |
|---|---|
| Description | Returns an array of each token, as parsed by the Delim parser. If a field number is provided, just the corresponding token substring is returned. |
| Prototype | String[] String::parseDelim (String delimeter) \| String String::parseDelim (String delimeter, Integer iField) |

### parseFixedWidth

| | |
|---|---|
| Description | Returns the corresponding token substring between the two indexes |
| Prototype | String String::parseFixedWidth (Integer beginIndex, Integer endIndex) |

### removeHTML

| | |
|---|---|
| Description | Returns a new string resulting from removing all html tags from the original string |
| Prototype | String removeHTML (String str) |

### replace

| | |
|---|---|
| Description | Returns a new string resulting from replacing all occurrences of the match substring in this string with the replacement substring |
| Prototype | String replaceString (String str, String match, String replacement) |

### replaceCharsNotInDecRangeWithHex

| | |
|---|---|
| Description | Does the replace where iStartDecRange and iEndDecRange are inclusive |

| | |
|---|---|
| | iEndDecRange are inclusive |
| Prototype | String replaceCharsNotInDecRangeWithHex (String str, Integer iStartDecRange, Integer iEndDecRange, String sEncoding, String sQualifier) |

### replaceString

| | |
|---|---|
| Description | Returns a new string resulting from replacing all occurrences of the match substring in this string with the replacement substring |
| Prototype | String replace (String str, String match, String replacement) |

### replaceUsingLookupTable

| | |
|---|---|
| Description | Return a string in which any substring matching a key in the lookup table is replace by the corresponding value |
| Prototype | String String::replaceUsingLookupTable(LookupTable lkp) |

### resizeString

| | |
|---|---|
| Description | Use to increase the size of a string to the finalLength by applying the appropriate padding to the left or right of the string with the given padChar. |
| Prototype | String resizeString (String str, Integer finalLength, Character padChar, Boolean padToTheRight) |

### splitLine

| | |
|---|---|
| Description | Returns an array of tokens obtained by breaking the line using this parser (e.g. CSV parser, fixed width parser) |
| Prototype | String[] IParser::splitLine() |

### startsWith

| | |
|---|---|
| Description | Tests if this string begins with an occurence of the match substring |
| Prototype | Boolean String::startsWith (String match) |

### stripOutNonASCII

| | |
|---|---|
| Description | Returns a new string resulting from removing all non-ASCII characters in this string |
| Prototype | String stripOutNonASCII (String str) |

### substitute

| | |
|---|---|
| Description | Substitutes a string for this regular expression in another string. This method works like the Perl function |
| Prototype | String RE::substitute(String substituteIn, String substitution) |

### substring

| | |
|---|---|
| Description | Returns a new string that is a substring of this string. The beginIndex is inclusive but endIndex is not. |
| Prototype | String substring (String str, Integer beginIndex [, Integer endIndex]) |

### toLowerCase

| | |
|---|---|
| Description | Converts all of the characters in this string to lower case using the rules of the default locale |
| Prototype | String toLowerCase (String str) |

### toTitleCase

| | |
|---|---|
| Description | Converts the first alphabet of all the words in a string to upper case |
| Prototype | String toTitleCase (String str) |

### toUpperCase

| | |
|---|---|
| Description | Converts all of the characters in this string to upper case using the rules of the default locale |
| Prototype | String toUpperCase (String str) |

### trim

| | |
|---|---|
| Description | Removes white space from both ends of this string |
| Prototype | String trim (String str) |

### unescapeHTMLEntities

| | |
|---|---|
| Description | Translates all character escaped with HTML character codes to corresponding characters |
| Prototype | String urlEncode(String str) |

### urlEncode

| | |
|---|---|
| Description | Translates a string into x-www-form-urlencoded format |
| Prototype | String urlEncode(String str) |

## Zip

### unzip

| | |
|---|---|
| Description | Unzip zip file given by srcPath into directory given by dstPath |
| Prototype | Boolean unzip(String srcPath, String dstPath) |

### zip

| | |
|---|---|
| Description | Zips files under directory given by srcPath and creates zip file given by dstPath |
| Prototype | Boolean zip(String srcPath, String dstPath) |

# Container

## Attribute Groups

### addAttributeToAttrGroup

| | |
|---|---|
| Description | Adds an attribute to the attribute collection. |
| Prototype | void AttrGroup::addAttributeToAttrGroup(String attrPath) |

### addLocalesToAttrGroup

| | |
|---|---|
| Description | Adds the locales to the Attribute Collection |
| Prototype | void AttrGroup::addLocalesToAttrGroup(String localesCSVString) |

### addLocalizedNodeToAttrGroup

| | |
|---|---|
| Description | Associates this localized node with this attribute collection |
| Prototype | void AttrGroup::addLocalizedNodeToAttrGroup(Node node) |

### addSpecToAttrGroup

| | |
|---|---|
| Description | Associates all the nodes of the spec with this attribute collection. If the bDynamic flag is set to true then any additional nodes added to the spec, after the spec has been associated to the Attribute Collection, will become part of the Attribute Collection. If the bDynamic flag is set to false then only the nodes that are part of the spec at this time only will be added to the Attribute Collection. |
| Prototype | void AttrGroup::addSpecToAttrGroup(Spec spec, [boolean bDynamic]) |

### deleteAttrGroup

| | |
|---|---|
| Description | Deletes this attribute collection |
| Prototype | void AttrGroup::deleteAttrGroup() |

### getAllAttrGroupsForAttribute

| | |
|---|---|
| Description | Returns an array of AttrGroups where the attrPath is included. Return null if attrPath is not included in any Attribute Group. |
| Prototype | AttrGroup[] getAllAttrGroupsForAttribute(String attrPath) |

### getAllAttributePathsFromAttrGroup

| | |
|---|---|
| Description | Returns all the attribute paths associated with this attribute collection |
| Prototype | String[] AttrGroup::getAllAttributePathsFromAttrGroup() |

### getAttrGroupByName

| | |
|---|---|
| Description | Returns the attribute collection with the given name. Otherwise it becomes null. |

| | |
|---|---|
| Prototype | AttrGroup getAttrGroupByName(String name) |

### getAttrGroupName

| | |
|---|---|
| Description | Returns the name of this attribute collection |
| Prototype | String AttrGroup::getAttrGroupName() |

### getAttrGroupType

| | |
|---|---|
| Description | Returns the type of this attribute collection. Type can only be GENERAL or INHERITANCE. |
| Prototype | String AttrGroup::getAttrGroupType() |

### new$AttrGroup

| | |
|---|---|
| Description | Returns a new attribute collection with the given name, type and description. Type can either be GENERAL or INHERITANCE |
| Prototype | new AttrGroup(String name, String type, [String desc]) |

### removeAttributeFromAttrGroup

| | |
|---|---|
| Description | Removes the attribute from the attribute collection. |
| Prototype | void AttrGroup::removeAttributeFromAttrGroup(String attrPath) |

### removeLocalesFromAttrGroup

| | |
|---|---|
| Description | Removes the locales from the Attribute Collection |
| Prototype | void AttrGroup::removeLocalesFromAttrGroup(String localesCSVString) |

### removeSpecFromAttrGroup

| | |
|---|---|
| Description | Disassociates all the nodes of the spec from this attribute collection |
| Prototype | void AttrGroup::removeSpecFromAttrGroup(Spec spec) |

# Catalog

### buildTestCatalogData

| | |
|---|---|
| Description | Create a document at sDocStorePath for the file specification fileSpec with nbRows of random data, with the primary key starting at firstSku |
| Prototype | buildTestCatalogData(Spec fileSpec, String sDocStorePath, Integer nbRows [, Integer firstSku]) |

### containsByPrimaryKey

| | |
|---|---|
| Description | Returns true if the catalog or item set contains an item with the primary key sPrimaryKey |
| Prototype | boolean Catalog::containsByPrimaryKey(String sPrimaryKey) - boolean ItemSet::containsByPrimaryKey(String sPrimaryKey) |

### deleteCatalog

| | |
|---|---|
| Description | (deprecated) |
| Prototype | |

### disableInheritance

| | |
|---|---|
| Description | Will not retrieve Inherited data for the container from the database |
| Prototype | void Container::disableInheritance() |

### exportCatalog

| | |
|---|---|
| Description | Use to syndicate a catalog using mktplaceSpec and specMap |
| Prototype | void Catalog::exportCatalog(Spec mktplaceSpec, SpecMap specMap) |

### getCatalogAccessControlGroupName

| | |
|---|---|
| Description | Returns the Access Control Group for this catalog. |
| Prototype | String Catalog::getCatalogAccessControlGroupName() |

### getCatalogAttribute

| | |
|---|---|
| Description | Returns a list of values for the attribute sAttribName |

| | |
|---|---|
| Prototype | String[] Catalog::getCatalogAttribute(String sAttribName) |

### getCatalogAttributes

| | |
|---|---|
| Description | Returns a HashMap mapping attributes to their respective values |
| Prototype | HashMap Catalog::getCatalogAttributes() |

### getCatalogCategoryTrees

| | |
|---|---|
| Description | Return an array with category trees of this catalog |
| Prototype | HashMap Catalog::getCatalogCategoryTrees() |

### getCatalogId

| | |
|---|---|
| Description | returns the id of this catalog. |
| Prototype | Integer Catalog::getCatalogId() |

### getCatalogItemCountInVersion

| | |
|---|---|
| Description | Returns the number of items in the specified version of this catalog |
| Prototype | Integer Catalog::getCatalogItemCountInVersion(Version version) |

### getCatalogNamesList

| | |
|---|---|
| Description | Return the list of names of available catalogs filtered by catalog privileges LIST (list catalog), VIEW_ITEMS (view items in catalog), MODIFY_ITEMS (modify items in catalog). By default the catalog names for the catalogs with LIST privilege access are returned. |
| Prototype | String[] getCatalogNamesList([String filterByPrivilege] |

### getCatalogsByAttributeValue

| | |
|---|---|
| Description | Returns all catalogs that have the provided value for the attribute |
| Prototype | Catalog[] getCatalogsByAttributeValue(String attribute_name, String value) |

### getCatalogSpec

| | |
|---|---|
| Description | Returns the spec this catalog. If the optional boolean bGetImmutableSpec is set to true, an immutable spec is returned. |
| Prototype | Spec Catalog::getCatalogSpec([Boolean bGetImmutableSpec]) |

### getCatalogVersion

| | |
|---|---|
| Description | Returns the version of this catalog. |
| Prototype | Version Catalog::getCatalogVersion() |

### getCatalogVersionSummary

| | |
|---|---|
| Description | Return an array with versions of this catalog - most recent first |
| Prototype | Versions[] Catalog::getCatalogVersionSummary() |

### getCategorizedItemCountInVersion

| | |
|---|---|
| Description | Returns the number of items categorized in the specified category tree for the specified version of this catalog |
| Prototype | Integer Catalog::getCategorizedItemCountInVersion(Version version, CategoryTree ctgtree) |

### getContainerId

| | |
|---|---|
| Description | Returns the id of this container. |
| Prototype | Integer Container::getContainerId() |

### getContainerLocalesForRole

| | |
|---|---|
| Description | Gets the locales that are allowed for this container specifically for the particular role. |
| Prototype | String Container::getContainerLocalesForRole(Role role) |

### getCtgByName

| | |
|---|---|
| Description | Returns the catalog object with the corresponding name. If no name is provided, return the default catalog (if defined). |

| | |
|---|---|
| Prototype | Catalog getCtgByName([String name] |

### getCtgCategorySpecs

| | |
|---|---|
| Description | Returns the category specs for this catalog |
| Prototype | HashMap Catalog::getCtgCategorySpecs() |

### getCtgFileDiffStatus

| | |
|---|---|
| Description | Returns true or false to indicate whether or not the file was modified between the two versions selected for differences syndication |
| Prototype | Boolean getCtgFileDiffStatus(String sFileName) |

### getCtgFileExists

| | |
|---|---|
| Description | Returns true or false to indicate whether the physical file really exists |
| Prototype | Boolean getCtgFileExists(String sFileName) |

### getCtgItemByAttributeValue

| | |
|---|---|
| Description | Returns ItemSet of items from the catalog that have the provided value for the attribute (currently implemented only for nodes in the catalog spec) |
| Prototype | ItemSet Catalog::getCtgItemByAttributeValue(String node_path, String value) |

### getCtgItemByPrimaryKey

| | |
|---|---|
| Description | Method deprecated. Use Container::getEntryByPrimaryKey. Returns the item from the catalog with the given primary key - this method cannot be used to retrieve newly created items that have not been saved yet. |
| Prototype | Item Catalog::getCtgItemByPrimaryKey(String sPrimaryKey) |

### getCtgItemIdByPrimaryKey

| | |
|---|---|
| Description | Returns an item id by primary key |
| Prototype | Integer Catalog::getCtgItemIdByPrimaryKey(String sPrimaryKey) |

### getCtgName

| | |
|---|---|
| Description | Returns the name of this catalog |
| Prototype | String Catalog::getCtgName() |

### getCtgSpec

| | |
|---|---|
| Description | Returns the spec this catalog. If the optional boolean bGetImmutableSpec is set to true, an immutable spec is retrieved (you can not modify the spec, but it is faster to retrieve). By default you get a mutable spec. |
| Prototype | Spec Catalog::getCtgSpec([Boolean bGetImmutableSpec]) |

### getDefaultCatalogName

| | |
|---|---|
| Description | See getCtgByName(). Returns the name of the catalog being used for an aggregation/syndication. |
| Prototype | (deprecated) String getDefaultCatalogName() |

### getItemBySku

| | |
|---|---|
| Description | (deprecated) see getCtgItemByPrimaryKey |
| Prototype | (deprecated) see getCtgItemByPrimaryKey |

### getItemSetForCatalog

| | |
|---|---|
| Description | returns an ItemSet of the items in this catalog |
| Prototype | ItemSet Catalog::getItemSetForCatalog() |

### getItemSetForPrimaryKeys

| | |
|---|---|
| Description | Returns an ItemSet of the items in this catalog for the given primary keys - set bOptimize to true if you don't plan on changing the items, the item set is then optimized but these items don't keep track of changed attributes |
| Prototype | ItemSet Catalog::getItemSetForPrimaryKeys(Array pkeys, Boolean bOptimize) |

### getItemsInCategory

| | |
|---|---|
| Description | Returns an array of the items in this category. The option Boolean 'ordered' being set to true |

| | |
|---|---|
| | makes the operation return the ordered children of this category if the catalog is set up to use ordering. |
| Prototype | Item[] Catalog::getItemsInCategory(Category, [Boolean ordered]) |

### getPrimaryCategoryTree

| | |
|---|---|
| Description | Returns the primary category tree of this catalog |
| Prototype | CategoryTree Catalog::getPrimaryCategoryTree() |

### hasCtgListPermission

| | |
|---|---|
| Description | Returns true if the current user has permission to list this catalog, false otherwise |
| Prototype | Boolean Catalog::hasCtgListPermission() |

### insertNewVersion

| | |
|---|---|
| Description | Add a version called sName on this catalog |
| Prototype | Version Catalog::insertNewVersion(String sName) |

### loadCatalog

| | |
|---|---|
| Description | Use to aggregate a file into the catalog using the given fileSpec and the given specmap. |
| Prototype | void Catalog::loadCatalog(String docStorePathForFileToLoad, Spec fileSpec, SpecMap specMap, String feedType [itm\|icm\|ctr]) |

### new Catalog

| | |
|---|---|
| Description | Returns a new catalog with the given spec and name. Pass optional args in the map with these keys "useInheritance" (default is false), "displayAttribute" (path of node), "accessControlGroup" (pass the ACG object), "isLookupTable" (default is false--set to true to create a lookup table and the Default Lookup Table Hierarchy is used as the category tree). If the displayAttribute is not set, the pk attribute is used. |
| Prototype | new Catalog(Spec catalogSpec, String name, CategoryTree categoryTree [,Hashmap |

| | |
|---|---|
| | optionalArgs] |

### setCatalogAccessControlGroupName

| | |
|---|---|
| Description | Sets the Access Control Group to the given name for this catalog. |
| Prototype | void Catalog::setCatalogAccessControlGroupName(String acgName) |

### setContainerProperties

| | |
|---|---|
| Description | The properties specified in the PROPERTIES hashmap are set for the container in question. The hashmap keys can be one of SCRIPT_NAME PRE_SCRIPT_NAME PREVIEW_SCRIPT_NAME ENTRY_BUILD_SCRIPT. The values are required to be string names. |
| Prototype | |

### setDefaultCtgView

| | |
|---|---|
| Description | Sets the ctgview as the default catalog view. |
| Prototype | void Catalog::setDefaultCtgView(CtgView ctgView) |

## Category

### addChildCategory

| | |
|---|---|
| Description | Adds childCategory as a child of this category |
| Prototype | Boolean Category::addChildCategory(Category childCategory) |

### addItemSecondarySpecToCategory

| | |
|---|---|
| Description | Associates a secondary spec defining this categories attrs. The optional parameters allows for the Spec to be associated with the category but does not build out the EntryNode structure, useful to improve performance on imports |
| Prototype | void Category::addSecondarySpecToCategory(String sSpecName, [Boolean bAdd] |

### addSecondarySpecToCategory

| | |
|---|---|
| Description | Associates a secondary spec defining this categories attrs. |
| Prototype | void Category::addSecondarySpecToCategory(String sSpecName) |

### deleteCategory

| | |
|---|---|
| Description | Delete the category |
| Prototype | void Category::deleteCategory() |

### getCategoryAttrib

| | |
|---|---|
| Description | Returns the value of the attribute sAttribPath (spec_name/attribute_name) of this category |
| Prototype | Object Category::getCategoryAttrib(String sAttribPath) |

### getCategoryChildren

| | |
|---|---|
| Description | Returns the categories immediately below this category. The option Boolean 'ordered' being set to true makes the operation return the ordered children of this category if the catalog (if not specified, the default catalog) is set up to use ordering. The option restrictToSubtreeWithItems being set to true only returns categories that have items in their sub-trees. |
| Prototype | Category[] Category::getCategoryChildren([Boolean ordered, Catalog catalog, Boolean restrictToSubtreeWithItems]) |

### getCategoryHasChildren

| | |
|---|---|
| Description | Returns true if the category has children. |
| Prototype | Boolean Category::getCategoryHasChildren() |

### getCategoryLevels

| | |
|---|---|
| Description | Returns the levels of this category in an array of Integers. |
| Prototype | Integer[] Category::getCategoryLevels() |

### getCategoryOrganizations

| | |
|---|---|
| Description | Return the all organizations this category is mapped to |
| Prototype | Organization[] Category::getCategoryOrganizations() |

### getCategoryParent

| | |
|---|---|
| Description | Returns this category's parent. If there are multiple parents, only the first one is returned. |
| Prototype | Category Category::getCategoryParent ([CategoryCache cat_cache]) |

### getCategoryParents

| | |
|---|---|
| Description | Returns the parent categories of this Category |
| Prototype | Category[] Category::getCategoryParents () |

### getCategoryTree

| | |
|---|---|
| Description | Returns the category tree object this category belongs to. Use getCategoryTreeByName() to get the category tree being used for an aggregation/syndication. |
| Prototype | CategoryTree Category::getCategoryTree() |

### getFullPaths

| | |
|---|---|
| Description | Returns the full name paths of this Category, using the sDelimiter as the delimiter if provided. The full path returned includes the root categories name if bWithRootName is true. |
| Prototype | String[] Category::getFullPaths ([String sDelimiter], [boolean bWithRootName] |

### getItemSecondarySpecsForCategory

| | |
|---|---|
| Description | Returns the item secondary specs associated with this category |
| Prototype | Spec[] Category::getItemSecondarySpecsForCategory([Catalog ctg]) |

### getItemSetForCategory

| | |
|---|---|
| Description | Returns an ItemSet of the items in this category. The option Boolean 'ordered' being set to true |

| | |
|---|---|
| | The option Boolean 'ordered' being set to true makes the operation return the ordered children of this category if the catalog is set up to use ordering. |
| Prototype | ItemSet Category::getItemSetForCategory(Catalog ctg, [Boolean ordered]) |

### getMappedCategories

| | |
|---|---|
| Description | Returns the categories in ctr (if any) to which this category is mapped |
| Prototype | Category[] Category::getMappedCategories(CategoryTree ctr) |

### getSecondarySpecsForCategory

| | |
|---|---|
| Description | Returns the secondary specs defining this categories attrs |
| Prototype | Spec[] Category::getSecondarySpecsForCategory() |

### mapCategoryToOrganizations

| | |
|---|---|
| Description | Maps the category to all the organizations provided. If bAdd is true, the old mappings are added to otherwise they are overwritten to be the new set of organizations |
| Prototype | void Category::mapCategoryToOrganizations(Category[] categories [, boolean bAdd]) |

### removeChildCategory

| | |
|---|---|
| Description | Remove childCategory from this category's children. Only allowed if childCategory has at least another parent. |
| Prototype | void Category::removeChildCategory(String categoryName) |

### removeItemSecondarySpecFromCategory

| | |
|---|---|
| Description | Disassociates a secondary item spec to from this Category. |
| Prototype | void |

| | |
|---|---|
| | Category::removeItemSecondarySpecFromCategory(String sSpecName)) |

### removeSecondarySpecFromCategory

| | |
|---|---|
| Description | Disassociates a secondary spec defining this categories attrs. |
| Prototype | void Category::removeSecondarySpecFromCategory(String sSpecName)) |

### setCategoryAttrib

| | |
|---|---|
| Description | Sets the attribute sAttribPath (spec_name/attribute_name) of this category to sValue |
| Prototype | void Category::setCategoryAttrib(String sAttribPath, Object sValue) |

## CategorySet

### forEachCategorySetElement

| | |
|---|---|
| Description | Executes the statements for each (oCategory) in the categorySet |
| Prototype | forEachCategorySetElement(CategorySet categorySet, Object oCategory) { statements } |

### getCategorySetSize

| | |
|---|---|
| Description | Returns the number of categories in a category set |
| Prototype | Integer CategorySet::getCategorySetSize() |

## CategoryTree

### buildCategory

| | |
|---|---|
| Description | Returns a new category object when given the complete path of the new category and the delimiter that separates the categories in the path. If the primary key is not specified, then it should either be automatically set via a sequence or value rule, or it should be set after creation. The |

| | |
|---|---|
| | final path part will be initially assigned to the pk, if the pk is not supplied. |
| Prototype | Category CategoryTree::buildCategory(String path, [String delimiter], [String primaryKey] |

### deleteCategoryTree

| | |
|---|---|
| Description | Delete the category tree ctr. Returns Validation Error array if any validation errors occured. Null if successful |
| Prototype | ValidationError[] deleteCategoryTree(CategoryTree ctr) |

### getCategoryByPath

| | |
|---|---|
| Description | Returns the category with a full name path equivalent to sNamePath. sNamePath is expected to be delimited by sDelim. sNamePath should not contain the name of the root category, since we are already restricted to a spcific category tree. If bLight is true, not all data for the category is retrieved. If bReadOnly is true, a read only copy of the category is retrieved - bReadOnly should be used in exports, for example |
| Prototype | Category CategoryTree::getCategoryByPath (String sNamePath, String sDelim, [boolean bUsingCode]) |

### getCategoryByPathNoCfp

| | |
|---|---|
| Description | Returns the category with a full name path equivalent to sNamePath. sNamePath is expected to be delimited by sDelim. sNamePath should not contain the name of the root category, since we are already restricted to a spcific category tree. If bLight is true, not all data for the category is retrieved. If bReadOnly is true, a read only copy of the category is retrieved - bReadOnly should be used in exports, for example |
| Prototype | Category CategoryTree::getCategoryByPathNoCfp (String sNamePath, String sDelim [, boolean bLight, boolean bReadOnly]) |

### getCategoryCache

| | |
|---|---|
| Description | Returns a CategoryCache for this CategoryTree. The cache will be empty if get_all_categories is false and the size will be the greater of the the given size or 100. If get_all_categories is true then the cache will contain all the categories for the given category tree and the size arguments will be ignored. The size of the cache in the latter case will be the greater of the number of categories in the tree or 100 |
| Prototype | CategoryCache CategoryTree::getCategoryCache(Integer size, Boolean get_all_categories) |

### getCategorySet

| | |
|---|---|
| Description | Returns a CategorySet for this CategoryTree |
| Prototype | CategorySet CategoryTree::getCategorySet([Boolean bReadonly]) |

### getCategorySetByAttributeValue

| | |
|---|---|
| Description | Returns a CategorySet with all categories in the category tree which have the given AttribName and AttribValue |
| Prototype | CategorySet CategoryTree::getCategorySetByAttributeValue(String attribName, Object attribValue, [Boolean bReadOnly] |

### getCategorySetByFullNamePath

| | |
|---|---|
| Description | Returns an CategorySet of the categories in the category tree from the given full name paths. Do not include the category tree name in the full name paths |
| Prototype | CategorySet CategoryTree::getCategorySetByFullNamePath(String[] fullNamePaths, String delimiter ) |

### getCategorySetByItemSecondarySpec

| | |
|---|---|
| Description | Returns an CategorySet that is a subset of the categories of this tree having the specified spec in their item secondary spec list] |

| | |
|---|---|
| Prototype | CategorySet CategoryTree::getCategorySetByItemSecondarySpec(String specName) |

### getCategorySetByLevel

| | |
|---|---|
| Description | Returns an CategorySet of the categories in the category tree at a particular level |
| Prototype | CategorySet CategoryTree::getCategorySetByLevel(Integer level, [Boolean bReadOnly]) |

### getCategorySetByPrimaryKey

| | |
|---|---|
| Description | Returns a CategorySet with the categories in the category tree which have match the primary key |
| Prototype | CategorySet CategoryTree::getCategorySetByPrimaryKey(String primaryKey, [Boolean bReadOnly]) |

### getCategorySetByStandAloneSpec

| | |
|---|---|
| Description | Returns an CategorySet that is a subset of the categories of this tree having the specified spec in their stand alone spec list |
| Prototype | CategorySet CategoryTree::getCategorySetByStandAloneSpec (String specName) |

### getCategoryTreeByName

| | |
|---|---|
| Description | Returns the category tree object with the corresponding name. If name is not provided, return the category tree being used for the aggregation/syndication. |
| Prototype | CategoryTree getCategoryTreeByName([String name]) |

### getCategoryTreeName

| | |
|---|---|
| Description | returns the name of this categoryTree. |
| Prototype | String CategoryTree::getCategoryTreeName() |

### getCategoryTreeNamesList

| | |
|---|---|
| Description | Return the list of names of available category trees filtered by category tree privileges LIST (list |

| | |
|---|---|
| | category tree), VIEW_ITEMS (view items in category tree), MODIFY_CATEGORY_ATTRIBUTES (modify category attributes in category tree). By default the category tree names for the category tree with LIST privilege access are returned. |
| Prototype | String[] getCategoryTreeNamesList([String filterByPrivilege]) |

### getCategoryTreeSpec

| | |
|---|---|
| Description | Returns the spec of this category tree |
| Prototype | Spec CategoryTree::getCategoryTreeSpec() |

### getDefaultCategoryTreeName

| | |
|---|---|
| Description | See getCategoryTreeByName(). Returns the name of the category tree being used for an aggregation/syndication. Use getCategoryTreeByName() to get the category tree being used for the aggregation/syndication. |
| Prototype | (deprecated) String getDefaultCategoryTreeName() |

### getDefaultCtrViewName

| | |
|---|---|
| Description | Returns name of default category tree view. |
| Prototype | String CategoryTree::getDefaultCtrViewName() |

### hasCtrListPermission

| | |
|---|---|
| Description | Returns true if the current user has permission to list this category tree, false otherwise |
| Prototype | Boolean CategoryTree::hasCtrListPermission() |

### new$Category

| | |
|---|---|
| Description | Returns a new category object when given the complete path of the new category and the delimiter that separates the categories in the path. If the primary key is not specified, then it should either be automatically set via a sequence or value rule, or it should be set after creation. The final path part will be initially assigned to the pk, if the pk is not supplied. |
| Prototype | new Category(CategoryTree ctr, String path, [String delimiter], [String primaryKey] |

| | |
|---|---|
| | [String delimiter], [String primaryKey] |

### new CategoryTree

| | |
|---|---|
| Description | Returns a new category tree with the given primary spec and name. Pass optional args in the map with these keys "useInheritance" (default is false), "displayAttribute" (path of node), "pathAttribute" (path of node), "accessControlGroup" (pass the ACG object), "isOrganizationTree" (default is false--set to true to create an organization tree). If the pathAttribute is not set, the primary key will be used. If the displayAttribute is not set, the pathAttribute is used. |
| Prototype | new CategoryTree(Spec spec, String name [,HashMap optionalArgs]] |

### setDefaultCtrView

| | |
|---|---|
| Description | Sets the ctrview as the default category tree view. |
| Prototype | void CategoryTree::setDefaultCtrView(CtgView ctrView) |

### saveCategoryTree

| | |
|---|---|
| Description | Saves this category tree. DO NOT USE in AGGREGATION if you are in a item-to-category feed or a category tree feed. The category tree you are aggregating to gets saved automatically at the end of an aggregation. However, if you side affect another category tree, then call this operation to capture the changes you made. Returns Validation Error array if any validation errors occured. Null if successful |
| Prototype | ValidationError[] CategoryTree::saveCategoryTree () |

### setCategoryCacheFetchSize

| | |
|---|---|
| Description | Sets the category cache fetch size (i.e. the number of categories gotten in bulk each time). This is only applicable if the category cache is associated with an ItemSet. |
| Prototype | void CategoryCache::setCategoryCacheFetchSize(Inte |

ger i)

# CategoryTreeMap

### addCategoryTreeMapping

| | |
|---|---|
| Description | Add a map between the two categories cat1 and cat2 |
| Prototype | void CategoryTreeMap::addCategoryTreeMapping(Category cat1, Category cat2) |

### getCategoryTreeMap

| | |
|---|---|
| Description | Returns the category tree map between the two category trees ctr1 and ctr2 |
| Prototype | CategoryTreeMap getCategoryTreeMap(CategoryTree ctr1, CategoryTree ctr2) |

### removeCategoryTreeMapping

| | |
|---|---|
| Description | Remove a map between the two categories cat1 and cat2 |
| Prototype | void CategoryTreeMap::removeCategoryTreeMapping(Category cat1, Category cat2) |

### saveCategoryTreeMap

| | |
|---|---|
| Description | Save this category tree map |
| Prototype | void CategoryTreeMap::saveCategoryTreeMap() |

# Item

### buildCtgItem

| | |
|---|---|
| Description | |
| Prototype | (deprecated) see new$CtgItem |

### cloneItem

| | |
|---|---|
| Description | Create and return a clone of this item. |
| Prototype | Item Item::cloneItem() |

### deleteCtgItem

| | |
|---|---|
| Description | Delete the catalog item itm |
| Prototype | void deleteCtgItem(Item itm) |

### displayCtgItemAttrib

| | |
|---|---|
| Description | Returns the html string for displaying item attribute specified by attribute path |
| Prototype | String Item::displayCtgItemAttrib(String sAttribPath) |

### forEachCtgItem

| | |
|---|---|
| Description | Executes the statements for each item in the catalog called sCatalogName |
| Prototype | forEachCtgItem([String sCatalogName, ], Item item) { statements } |

### getCatalog

| | |
|---|---|
| Description | Returns the catalog for this item. |
| Prototype | Catalog Item::getCatalog() |

### getChangedAttributes

| | |
|---|---|
| Description | Returns an array of changed attribute paths |
| Prototype | String[] Entry::getChangedAttributes(Entry secondEntry) |

### getCtgItemAllCategories

| | |
|---|---|
| Description | (Deprecated) See getCtgItemCategories. Return the all categories this item is mapped to, |
| Prototype | Category[] Item::getCtgItemCategories() |

### getCtgItemAtOldVersion

| | |
|---|---|
| Description | Returns the old version of the item in the differences syndication. |
| Prototype | Item Item::getCtgItemAtOldVersion() |

### getCtgItemAttribByPk

| | |
|---|---|
| Description | Returns the value of the attribute sAttribPath (spec_name/attribute_name) of this item |

| | |
|---|---|
| Prototype | Object Catalog::getCtgItemAttribByPk(String pk, String sAttribPath) |

### getCtgItemOrganizations

| | |
|---|---|
| Description | Return the all organizations this item is mapped to |
| Prototype | Organization[] Item::getCtgItemOrganizations() |

### getCtgItemAttrib

| | |
|---|---|
| Description | Returns the value of the attribute sAttribPath (spec_name/attribute_name) of this item |
| Prototype | Object Item::getCtgItemAttrib(String sAttribPath) |

### getCtgItemAttribNamesList

| | |
|---|---|
| Description | Returns an array of String containing the attribute name of all the attributes of this item (optional parameter allows option exclude categorySpecificAttribute - true by default) |
| Prototype | String[] Item::getCtgItemAttribNamesList([Boolean bAllAttributes]) |

### getCtgItemAttribs

| | |
|---|---|
| Description | Returns a HashMap mapping the paths (spec_name/attribute_name) of attributes to their respective values |
| Prototype | HashMap Item::getCtgItemAttribs() |

### getCtgItemAttribsList

| | |
|---|---|
| Description | Returns an array of String containing the paths (spec_name/attribute_name) of all the attributes of this item |
| Prototype | String[] Item::getCtgItemAttribsList() |

### getCtgItemAttributeNewValue

| | |
|---|---|
| Description | |
| Prototype | (deprecated) use Item::getCtgItemAttrib() |

### getCtgItemAttributeOldValue

(deprecated) use Item::getCtgItemAtOldVersion()

### getCtgItemAttributesStatus

| | |
|---|---|
| Description | Returns a HashMap of {attributePath}->{the difference status (A, M, D, U)} for each attribute of the item. |
| Prototype | HashMap Item::getCtgItemAttributesStatus() |

### getCtgItemCategories

| | |
|---|---|
| Description | Return the categories this item is mapped to. If catTreeName is given, returns the categories within that ctr only (use the default category tree if no category tree is passed). Also, can use an optional CategoryCache passed in catCache |
| Prototype | Category[] Item::getCtgItemCategories([String catTreeName] [, CategoryCache catCache]) |

### getCtgItemCategoryPaths

| | |
|---|---|
| Description | Returns an array of delimited strings of the category paths this item belongs to. If ctr is given, returns the paths of the categories within that ctr only. |
| Prototype | String[] Item::getCtgItemCategoryPaths(String sPathDelimiter, [Boolean bWithRoot], [CategoryTree ctr]) |

### getCtgItemCatSpecificAttribsList

| | |
|---|---|
| Description | Returns an array of String containing the paths (spec_name/attribute_name) of all the category specific attributes of this item |
| Prototype | String[] Item::getCtgItemCatSpecificAttribsList() |

### getCtgItemDiffStatus

| | |
|---|---|
| Description | For content difference syndications, returns this item's difference status (A, M, D, U) |
| Prototype | String Item::getCtgItemDiffStatus() |

### getCtgItemId

| | |
|---|---|
| Description | Returns this item's Id |
| Prototype | Integer Item::getCtgItemId() |

### getCtgItemMappedAttrib

| | |
|---|---|
| Description | Returns the value of the attribute mapped to/from sAttribMappedPath (mapped_spec_name/attribute_name) of this item |
| Prototype | String Item::getCtgItemMappedAttrib(String sAttribMappedPath) |

### getCtgItemMappedAttribs

| | |
|---|---|
| Description | Returns a HashMap with the mapped attributes values indexed by their path (mapped_spec_name/attribute_name) of this item |
| Prototype | HashMap Item::getCtgItemMappedAttribs() |

### getCtgItemMappedAttribsList

| | |
|---|---|
| Description | Returns an array of String containing the paths (mapped_spec_name/attribute_name) of all the mapped attributes of this item |
| Prototype | String[] Item::getCtgItemMappedAttribsList() |

### getCtgItemPrimaryKey

| | |
|---|---|
| Description | Returns this item's primary key value |
| Prototype | String Item::getCtgItemPrimaryKey() |

### getCtgItemRelatedItemInfo

| | |
|---|---|
| Description | Returns an array of length 2 containing: [0]=Related Item's Catalog's Name, [1]=Related Item's Primary Key, for the related item represented by the given internal unique item id, at the browsing version of the catalog of the given item |
| Prototype | String[] Item::getCtgItemRelatedItemInfo(Integer iItemId) |

### getLinkedItemForNode

| | |
|---|---|
| Description | Returns the linked item associated with the specified node. |
| Prototype | Item Item::getLinkedItemForNode(String node_path) |

### getOriginalItem

| | |
|---|---|
| Description | Returns the original picture of the item before modification. Deprecated. Please use Entry::getOriginalEntry |
| Prototype | Item Item::getOriginalItem() |

### getRootItemNode

| | |
|---|---|
| Description | Return the root item node for this item |
| Prototype | EntryNode Item::getRootItemNode() |

### mapCtgItemToCategory

| | |
|---|---|
| Description | Map this item to this category. If optional boolean ADDTOPICTURE is false, the secondary specs will not be associated and cannot be set; useful for performance. If optional boolean VALIDATECATEGORY is true and the category's hierarchy does not have the VALIDATION_RULES option disabled, the mapping will only occur if the category passes validation. |
| Prototype | void Item::mapCtgItemToCategory(Category category, [Boolean addToPicture], [Boolean validateCategory]) |

### mapCtgItemToOrganizations

| | |
|---|---|
| Description | Maps the item to all the organizations provided. If bAdd is true, the old mappings are added to otherwise they are overwritten to be the new set of organizations. Deprecated--Call moveCtgItemToCategories |
| Prototype | void Item::mapCtgItemToOrganizations(Category[] organizations [, boolean bAdd]) |

### moveCtgItemToCategories

| | |
|---|---|
| Description | Move item from existing categories to new set of categories, if bAdd is true, then category mappings will be added. |
| Prototype | void Item::moveCtgItemToCategories(Category[] categories), [, boolean bAdd]) |

### new$CtgItem

| | |
|---|---|
| Description | Returns a new item object. The argument can be a catalog name or a catalog object. The argument being a catalog object allows the propagation of attribute collections to process settings etc. to new items being built with this operation. If no catalog name/object is provided, then the default catalog from the current script context is used. bRunEntryBuildScript or bBuildNonPersisted should be set to false to disable the default behavior of this script operation to run the entry build script or build the non-persisted attributes respectively for this new item. |
| Prototype | new CtgItem([String sCtgName/Catalog ctg], [Boolean bRunEntryBuildScript], [Boolean bBuildNonPersisted]) |

### removeCtgItemFromCategory

| | |
|---|---|
| Description | Remove mapping from this item to this category, if the mapping exists. |
| Prototype | void Item::removeCtgItemFromCategory(Category category) |

### saveCtgItem

| | |
|---|---|
| Description | Save the item. When called outside of an import script, returns null if the save was successful, otherwise returns an array of ValidationError's. When called in an import, returns null. |
| Prototype | ValidationError[] Item::saveCtgItem() |

### setCtgItemAttrib

| | |
|---|---|
| Description | Sets the attribute sAttribPath (spec_name/attribute_name) of this item to sValue |
| Prototype | void Item::setCtgItemAttrib(String sAttribPath, Object sValue) |

### setCtgItemMappedAttrib

| | |
|---|---|
| Description | Sets the attribute mapped to/from sAttribMappedPath (mapped_spec_name/attribute_name) of this item |

| | |
|---|---|
| | to sValue |
| Prototype | void Item::setCtgItemMappedAttrib(String sAttribPath, Object oValue) |

### setCtgItemMappedAttribs

| | |
|---|---|
| Description | Set the attributes of this item: hmPathValue should contain (path_y, value_x)'s; the item attribute path_x receives value_x if path_y is mapped to path_x in specmap - if no spec map is specified, the specmap of the import is being used. |
| Prototype | void Item::setCtgItemMappedAttribs(HashMap hmPathValue, [SpecMap specmap]) |

### setCtgItemPrimaryKey

| | |
|---|---|
| Description | Sets this item's primary key value |
| Prototype | void Item::setCtgItemPrimaryKey(String pk) |

### setCtgItemRelationshipAttrib

| | |
|---|---|
| Description | Sets the attribute sAttribPath (spec_name/attribute_path) of type RELATIONSHIP of this item to the related item represented by the given catalog and primary key |
| Prototype | void Item::setCtgItemRelationshipAttrib(String sAttribPath, Catalog relatedItemCtg, String sRelatedItemPrimaryKey) |

### setCtgItemRelationshipAttribUsingItem

| | |
|---|---|
| Description | Sets the attribute sAttribPath (spec_name/attribute_path) of type RELATIONSHIP of this item to the related item given |
| Prototype | void Item::setCtgItemRelationshipAttribUsingItem(String sAttribPath, Item relatedItem) |

### setExitValue

| | |
|---|---|
| Description | Set the exit value of an entry in a workflow step. Assumed to be called from an IN(), OUT(), or TIMEOUT() step script function. |

| | |
|---|---|
| Prototype | Entry::setExitValue(String exitValue) |

### setIgnoreCategorySpecificAttributes

| | |
|---|---|
| Description | Set whether or not category specific attributes should be processed for the item |
| Prototype | void Item::setIgnoreCategorySpecificAttributes(Boolean bIgnore) |

### validateMappedAttribs

| | |
|---|---|
| Description | Validate a set of attribute values indexed by their mapped path against the destination spec |
| Prototype | HashMap validateMappedAttribs(HashMap hmPathValue, [SpecMap specmap]) |

## ItemNode

### getItemNode

| | |
|---|---|
| Description | Return the first item node matching the path sPath |
| Prototype | ItemNode ItemNode::getItemNode(String sPath) |

### getItemNodeChildren

| | |
|---|---|
| Description | Return the children of this ItemNode |
| Prototype | ItemNode[] ItemNode::getItemNodeChildren() |

### getItemNodePath

| | |
|---|---|
| Description | Return the path of this item node |
| Prototype | String ItemNode::getItemNodePath() |

### getItemNodes

| | |
|---|---|
| Description | Return the item nodes matching the path sPath |
| Prototype | ItemNode[] ItemNode::getItemNodes(String sPath) |

### getItemNodeValue

| | |
|---|---|
| Description | Return the value of this ItemNode |
| Prototype | String ItemNode::getItemNodeValue() |

### setItemNode

| | |
|---|---|
| Description | Return the itemNode with path sPath (building any node needed along the path) or null if the path is invalid |
| Prototype | ItemNode ItemNode::setItemNode(String sPath) |

### setItemNodeRelationshipValue

| | |
|---|---|
| Description | Set the value of this ItemNode of type RELATIONSHIP to the related item represented by the given catalog and primary key |
| Prototype | void ItemNode::setItemNodeRelationshipValue(Catalog relatedItemCtg, String sRelatedItemPrimaryKey) |

### setItemNodeRelationshipValueUsingItem

| | |
|---|---|
| Description | Set the value of this ItemNode of type RELATIONSHIP to the related item given |
| Prototype | void ItemNode::setItemNodeRelationshipValueUsingItem(Item relatedItem) |

### setItemNodeValue

| | |
|---|---|
| Description | Set the value of this ItemNode and return it |
| Prototype | Object ItemNode::setItemNodeValue(Object value) |

## ItemSet

### associateCategoryCacheToItemSet

| | |
|---|---|
| Description | Associates the CategoryCache to the ItemSet so that when items are fetched, the corresponding categories are also fetched in bulk |
| Prototype | void ItemSet::associateCategoryCacheToItemSet(CategoryCache catCache) |

### forEachItemSetElement

| | |
|---|---|
| Description | Executes the statements for each (oItem) map in the ItemSet |

| | |
|---|---|
| | the ItemSet |
| Prototype | forEachItemSetElement(ItemSet is, Object oItem) { statements } |

### getItemSetSize

| | |
|---|---|
| Description | Returns the number of items in an item set |
| Prototype | Integer ItemSet::getItemSetSize() |

### setItemSetFetchCategorySpecificAttributes

| | |
|---|---|
| Description | Sets the item set to fetch or not fetch category specific attributes |
| Prototype | void ItemSet::setItemSetFetchCategorySpecificAttributes(Boolean b) |

### setItemSetFetchLinkedItems

| | |
|---|---|
| Description | Sets the item set to fetch or not fetch master linked items |
| Prototype | void ItemSet::setItemSetFetchLinkedItems(Boolean b) |

### setItemSetFetchSize

| | |
|---|---|
| Description | Sets the item set fetch size (i.e. the number of items gotten in bulk each time) |
| Prototype | void ItemSet::setItemSetFetchSize(Integer i) |

### sortItemSet

| | |
|---|---|
| Description | Sorts the ItemSet for performance |
| Prototype | void ItemSet::sortItemSet() |

## LookupTable

### addRow

| | |
|---|---|
| Description | Add a new row to this lookup table - with value(s) sValue/asValues for the key sKey. Returns TRUE if and only if the add was successful. |
| Prototype | Boolean LookupTable::addRow(String sKey, String sValue), Boolean |

| | |
|---|---|
| | LookupTable::addRow(String sKey, String[] asValues) |

### getKeysFromValues

| | |
|---|---|
| Description | Reverse lookup of keys using values from the lookup table. The values can either be Paths in the Spec or the column number of the lookup table starting from 0 and not including the Key column. |
| Prototype | String[] LookupTable::getKeysFromValues(String[] values) |

### getLkpByName

| | |
|---|---|
| Description | Returns the lookup table object with the corresponding name. By default the lookup table is read-only, but can be made mutable by setting the isReadOnly parameter to false. |
| Prototype | LookupTable getLkpByName(String name, [Boolean isReadOnly] |

### getLkpId

| | |
|---|---|
| Description | Return the id of this lookup table. |
| Prototype | Integer LookupTable::getLkpId() |

### getLkpKeys

| | |
|---|---|
| Description | Return the keys of this lookup table |
| Prototype | String[] LookupTable::getLkpKeys() |

### lookup

| | |
|---|---|
| Description | Returns the sSecKey-th value for sKey in the lookup table sLookupTableName or lkp |
| Prototype | String lookup(String sLookupTableName, String sKey [, String sSecKey]), String lookup(LookupTable lkp, String sKey [, String sSecKey]) |

### lookupValues

| | |
|---|---|
| Description | Returns values for sKey in the lookup table lkp |
| Prototype | String[] lookupValues(LookupTable lkp, String sKey) |

| | |
|---|---|
| | sKey) |

### put

| | |
|---|---|
| Description | Put a new row in the lookup table sLkpTableName |
| Prototype | void put(String sLkpTableName, String sStartKey, [String sEndKey,] String sValue), void put(String sLkpTableName, String sStartKey, [String sEndKey,] String[] asValues) |

## Queues

### createQueue

| | |
|---|---|
| Description | Creates a new queue with the given parameters. |
| Prototype | IMsgQueue createQueue (String queueName, String queueDesc, MsgQueueProtocolEnum protocol, String syncScriptPath)) |

### getMessageFromQueue

| | |
|---|---|
| Description | Gets the indexth oldest message from the given queue. For example, getMessageFromQueue("Queue1", 2) would return the 2nd oldest message from the queue with name "Queue1". If there is no such message or queue, returns null. |
| Prototype | Message getMessageFromQueue (String queueName, Integer index) |

### getMsgAppResponse

| | |
|---|---|
| Description | Initiates the request for response for a message. |
| Prototype | Void Message::getMsgAppResponse() |

### getMsgAppResponseDoc

| | |
|---|---|
| Description | Returns the Doc object for the message. |
| Prototype | Doc Message::getMsgAppResponseDoc() |

### getMsgAttachments

| | |
|---|---|
| Description | Returns a HashMap of attachment names to attachments for the given message. |

| | |
|---|---|
| Prototype | HashMap Message::getMsgAttachments () |

### getMsgByMsgId

| | |
|---|---|
| Description | Returns the message object with the message id msgId null otherwise. |
| Prototype | Message getMsgByMsgId(String msgId) |

### getMsgDoc

| | |
|---|---|
| Description | Returns the Doc object for the message. |
| Prototype | Doc Message::getMsgDoc() |

### getMsgId

| | |
|---|---|
| Description | Returns the generated unique id for the message. |
| Prototype | String Message::getMsgId() |

### getMsgProtocolResponseDoc

| | |
|---|---|
| Description | Returns the Doc object for the message. |
| Prototype | Doc Message::getMsgProtocolResponseDoc() |

### getMsgQueue

| | |
|---|---|
| Description | Returns the MsgQueue object for the message. |
| Prototype | MsgQueue Message::getMsgQueue() |

### qmgrGetMsgQueueByName

| | |
|---|---|
| Description | Returns the queue if present in the system. |
| Prototype | MsgQueue qmgrGetMsgQueueByName(String queueName) |

### sendMsg

| | |
|---|---|
| Description | Sends the message. If successful, will return a message object. If it fails it will return null. |
| Prototype | Message MsgQueue::sendMsg(Doc doc) |

### setMsgDoc

| | |
|---|---|
| Description | Sets the Doc object for the message. |
| Prototype | void Message::setMsgDoc(IDoc doc) |

## Selection

### addEntryToSelection

| | |
|---|---|
| Description | Add the entry to the basic selection - the entry can be an item or a hierarchy node (does nothing for advanced selection). |
| Prototype | void Selection::addEntryToSelection(Entry entry) |

### deleteSelection

| | |
|---|---|
| Description | Delete the selection. Return true if the deletion occured, false if selection was in use. |
| Prototype | boolean Selection::deleteSelection() |

### getHierarchyNodeSetForSelection

| | |
|---|---|
| Description | Return the hierarchy nodes in that selection as a HierarchyNodeSet |
| Prototype | HierarchyNodeSet Selection::getHierarchyNodeSetForSelection() |

### getItemSetForSelection

| | |
|---|---|
| Description | Return the items in that selection as a ItemSet |
| Prototype | ItemSet Selection::getItemSetForSelection() |

### getSelectionAccessControlGroupName

| | |
|---|---|
| Description | Returns the Access Control Group for this selection. |
| Prototype | String Selection::getSelectionAccessControlGroupName () |

### getSelectionByName

| | |
|---|---|
| Description | Return the selection called sName |
| Prototype | Selection getSelectionByName(String sName) |

### getSelectionCatalog

| | |
|---|---|
| Description | Returns the selection's catalog |
| Prototype | Catalog Selection::getSelectionCatalog() |

### getSelectionHierarchy

| | |
|---|---|
| Description | Returns the selection's hierarchy. |

| | |
|---|---|
| Prototype | Hierarchy Selection::getSelectionHierarchy() |

### getSelectionHierarchyNodeCount

| | |
|---|---|
| Description | Returns the number of hierarchy nodes in a selection - returns 0 for advanced selections. |
| Prototype | Integer Selection::getSelectionHierarchyNodeCount() |

### getSelectionItemCount

| | |
|---|---|
| Description | Return count of the items in the selection |
| Prototype | Integer Selection::getSelectionItemCount() |

### getSelectionName

| | |
|---|---|
| Description | Returns the name of this selection |
| Prototype | String Selection::getSelectionName() |

### getSelectionNamesList

| | |
|---|---|
| Description | Return the list of names of available selections for catalog |
| Prototype | String[] getSelectionNamesList(Catalog catalog) |

### new$AdvancedSelection

| | |
|---|---|
| Description | Builds a new advanced selection (Selection) with the given name/catalog and returns it. Call saveSelection to save it. |
| Prototype | new AdvancedSelection(Catalog catalog, String name, String expression) |

### new$BasicSelection

| | |
|---|---|
| Description | Returns an empty basic selection (Selection) on catalog |
| Prototype | new BasicSelection(Catalog catalog, String name) |

### saveSelection

| | |
|---|---|
| Description | Save the basic or advanced selection to the database |
| Prototype | void Selection::saveSelection() |

### setSelectionAccessControlGroupName

| | |
|---|---|
| Description | Sets the Access Control Group to the given name for this selection. |
| Prototype | void Selection::setSelectionAccessControlGroupName( String acgName) |

### setSelectionName

| | |
|---|---|
| Description | Returns the name of this selection |
| Prototype | void Selection::setSelectionName(String name) |

## Version

### getVersionDate

| | |
|---|---|
| Description | Returns the date of this version |
| Prototype | Date Version::getVersionDate() |

### getVersionName

| | |
|---|---|
| Description | Returns the type of this version |
| Prototype | String Version::getVersionName() |

### getVersionType

| | |
|---|---|
| Description | Returns the type of this version |
| Prototype | String Version::getVersionType() |

## Views

### Sample script for creating views

### addCtgTab

| | |
|---|---|
| Description | Add container tab object to the catalog view. The tab is added to the end of the list of tabs already defined for the container ctg view. |
| Prototype | void CtgView::addCtgTab(CtgTab tab) |

### getCtgTabAttrGroupsList

| | |
|---|---|
| Description | Returns a list of ordered attribute collections for the catalog view tab |

| | |
|---|---|
| | the catalog view tab |
| Prototype | String[] CtgTab::getCtgTabAttrGroupsList() |

### getCtgTabRow

| | |
|---|---|
| Description | Get the set of rows in the current container tab object |
| Prototype | CtgTabRow[] CtgTab::getCtgTabRow() |

### getCtgTabs

| | |
|---|---|
| Description | Gets an ordered array of container tab objects for the particular container view |
| Prototype | CtgTab[] CtgView::getCtgTabs() |

### getCtgViewAttrGroupsList

| | |
|---|---|
| Description | Returns list of ordered attribute groups for the catalog view. |
| Prototype | String[] CtgView::getCtgViewAttrGroupsList() |

### getCtgViewAttribsList

| | |
|---|---|
| Description | Returns list of ordered attribute paths for the catalog view. |
| Prototype | String[] CtgView::getCtgViewAttribsList() |

### getCtgViewByName

| | |
|---|---|
| Description | Returns the view with the corresponding name. If no name is specified, returns the default view. Use '[System Default]' to refer to the default view. The viewType can be 'ITEM_LIST', 'ITEM_POPUP', 'BULK_EDIT', 'ITEM_EDIT', 'CATEGORY_EDIT' or 'CATEGORY_BULK_EDIT'. By default ITEM_EDIT/CATEGORY_EDIT is used. If the view is not found, it returns null. |
| Prototype | CtgView Container::getCtgViewByName([String viewName, String viewType] |

### getCtgViewPermission

| | |
|---|---|
| Description | Returns the permission [E-editable\|V-viewable] for the node specified by the path in the current view. |

| | |
|---|---|
| Prototype | String CtgView::getCtgViewPermission(String attrGroupName) |

### getCurrentCtgViewName

| | |
|---|---|
| Description | Returns name of current catalog view (only in Data Entry scripts). Returns an empty string outside of the Data Entry scripts. |
| Prototype | String getCurrentCtgViewName() |

### getDefaultCtgViewName

| | |
|---|---|
| Description | Returns name of default catalog view. |
| Prototype | String Catalog::getDefaultCtgViewName() |

### getListOfCtgViewNames

| | |
|---|---|
| Description | Returns an array of view names available for this catalog. An entry with '[System Default]' is always included as the first entry. |
| Prototype | String[] Catalog::getListOfViewNames() |

### getNewCtgTab

| | |
|---|---|
| Description | Builds a new container tab object with the given name and returns it. The tab needs to be added to the catalog view in order to save it. |
| Prototype | CtgTab CtgView::getNewCtgTab(String name, AttrGroup[] rows) |

### getTabRowPath

| | |
|---|---|
| Description | Returns the attribute path for this tab row. |
| Prototype | String CtgTabRow::getTabRowPath() |

### insertCtgTabAt

| | |
|---|---|
| Description | Insert container tab object to the catalog view at the index position(zero base). If index is invalid, tab is added to the end of the list. |
| Prototype | void CtgView::insertCtgTabAt(CtgTab tab, Integer index) |

### new$CtgTabRow

| | |
|---|---|
| Description | Builds a new container tab row object for the node specified by the path. |

| | |
|---|---|
| Prototype | CtgTabRow CtgTabRow(String path) |

### new$CtgView

| | |
|---|---|
| Description | Builds a new Ctg View |
| Prototype | new CtgView(Container container, String name) |

### removeCtgTabAt

| | |
|---|---|
| Description | Remove container tab object to the catalog view at the index position (zero base) |
| Prototype | void CtgView::removeCtgTabAt(Integer index) |

### saveCtgTabs

| | |
|---|---|
| Description | Save the container tab objects that were new/modified in the container view |
| Prototype | void CtgView::saveCtgTabs() |

### saveCtgView

| | |
|---|---|
| Description | Saves the current ctgview to the database |
| Prototype | Boolean CtgView::saveCtgView() |

### setCtgTabRow

| | |
|---|---|
| Description | Sets the current container tab object to the new set of rows |
| Prototype | void CtgTab::setCtgTabRow(CtgTabRow[] rows) |

### setCtgView

| | |
|---|---|
| Description | Sets the container view object with the given name/catalog and returns it. The viewType can be 'ITEM_LIST', 'ITEM_POPUP', 'BULK_EDIT' or 'ITEM_EDIT'. By default ITEM_EDIT is used. Permissions are [V\|E] |
| Prototype | CtgView CtgView::setCtgView(String viewType, String[] paths, String[] permissions) |

### setTabular

| | |
|---|---|
| Description | Sets the CtgTabRow object (for grouping node only) to display the children in tabular format. string sets to 'T' or 'F' |
| Prototype | void CtgTabRow::setDisplayTabular(String str) |

### Sample Script for Creating Views

```
ctg = getCtgByName("Ctg1");
defCtgViewName = ctg.getDefaultCtgViewName();
editCtgView = ctg.getCtgViewByName(defCtgViewName);

mypath = [];
mypath.add("Ctg1/Key");
mypath.add("Ctg1/Group/EAN/en_MY");
mypath.add("Ctg1/Group/EAN/zh_MY");
mypath.add("Ctg1/Group/EAN/ms_MY");

//out.writeln(mypath);
myper = [];
myper.add("E");
myper.add("V");
myper.add("E");
myper.add("V");

newCtgView = newCtgView.setCtgView("ITEM_EDIT", mypath, myper);
newCtgView.saveCtgView();
newCtgView = newCtgView.setCtgView("BULK_EDIT", mypath, myper);
newCtgView.saveCtgView();
```

(given a catalog view, set the ctg view into different subview with different permissions and save them individually)

# DocStore

## Doc

### copyDoc

| | |
|---|---|
| Description | Copy this document to the specified sPath in the docstore. If the path ends with a '/' it is assumed that the doc needs to be copied to the specified directory with its current name |
| Prototype | Doc Doc::copyDoc(sPath) |

### deleteDoc

| | |
|---|---|
| Description | Delete this document from the docstore |
| Prototype | void Doc::deleteDoc() |

### forEachDocument

| | |
|---|---|
| Description | Executes the statements for each document (used in distribution scripts). If the optional docs_list |

| | |
|---|---|
| | parameter is provided, however, the statements are executed for each element of docs_list |
| Prototype | forEachDocument([Doc[] docs_list, ], Doc doc) { statements } |

### getDocAttribute

| | |
|---|---|
| Description | Return the attribute sAttributeName from this document |
| Prototype | String Doc::getDocAttribute(String sAttributeName) |

### getDocAttributes

| | |
|---|---|
| Description | Return the attributes of this document |
| Prototype | HashMap Doc::getDocAttributes() |

### getDocByPath

| | |
|---|---|
| Description | Return the document with path sPath |
| Prototype | Doc getDocByPath(String sPath) |

### getDocContentAsString

| | |
|---|---|
| Description | Return the content of this document as a string. WARNING - this means that the entire content of the document, however big, will be returned in a string so the uer needs to make sure that any call of this operation is not going to be used in a situation where the content of the document is too big (too big being defined by the amount of memory available to the process this operation is running in). |
| Prototype | String Doc::getDocContentAsString() |

### getDocLastModifiedTimeStamp

| | |
|---|---|
| Description | Returns the date/time this document was last modified |
| Prototype | Date Doc::getDocLastModifiedTimeStamp() |

### getDocLength

| | |
|---|---|
| Description | Returns the length of the document in kilo bytes. If bBytes is true, value is returned in bytes instead of Kb. Important when smaller files are |

| | |
|---|---|
| | concerned |
| Prototype | Integer Doc::getDocLength([Boolean bBytes |

### getDocListByPaths

| | |
|---|---|
| Description | Return the document at each path specified in sPaths |
| Prototype | Doc[] getDocListByPaths(String[] sPaths) |

### getDocPath

| | |
|---|---|
| Description | Return this document path |
| Prototype | String Doc::getDocPath() |

### getHrefForDocPath

| | |
|---|---|
| Description | Return a absolute path for the document with path sDocPath. This can be used in an HTML reference to provide a link to the document. |
| Prototype | String getHrefForDocPath(String sDocPath) |

### moveDoc

| | |
|---|---|
| Description | Move this document to the specified sPath in the docstore. If the path ends with a '/' it is assumed that the doc needs to be moved to the specified directory with the same doc name as the source |
| Prototype | Doc Doc::moveDoc(sPath) |

### saveMultipartRequestData

| | |
|---|---|
| Description | Saves the documents sent through multipart post in the docstore at location:/archives/uploaded/multipart/saveDir/. If a charset is given, that is used. Otherwise, the default_charset_value as specified in austin.properties is used. |
| Prototype | Doc[] saveMultipartRequestData(String saveDir, [String charset]) |

### setDocAttribute

| | |
|---|---|
| Description | Set the attribute sAttributeName to sAttributeValue for this document |
| Prototype | void Doc::setDocAttribute(String sAttributeName, String sAttributeValue) |

# DocStore

### getDocStoreDirectoriesInDirectory

| | |
|---|---|
| Description | Return the list of paths of directories under the directory sPath |
| Prototype | String[] getDocStoreDirectoriesInDirectory(String sPath) |

### getDocStoreFilesInDirectory

| | |
|---|---|
| Description | Return the list of paths of documents under the directory sPath |
| Prototype | String[] getDocStoreFilesInDirectory(String sPath) |

### getDocStoreSubtreeList

| | |
|---|---|
| Description | Return the list of files under sPath |
| Prototype | String[] getDocStoreSubtreeList(String sPath) |

# XML Document

### Sample Script for new$XMLDocument

### new XMLDocument

| | |
|---|---|
| Description | Creates an XmlDocument from a docstore Doc instance or a string. |
| Prototype | new XmlDocument(Doc doc/String str) |

### validateXML

| | |
|---|---|
| Description | Validates an XmlDocument from a docstore Doc instance |
| Prototype | String validateXML(String docPath) |

# Sample Script for new$XMLDocument

```
var files = getDocStoreFilesInDirectory("dms/files");
 var i;
```

```
var len = files.size();
 for( i = 0; i < len; i++)
 {
 var doc = getDocByPath(files[i]);
 var xmlDoc = new XmlDocument(doc);
 var action =  parseXMLNode("action");
 forEachXMLNode("properties/property")
 {
 ...
 }
 }
```

# Entry

# Entry

### displayEntryAttrib

| | |
|---|---|
| Description | Returns the html string for displaying entry attribute specified by attribute path |
| Prototype | String Entry::displayEntryAttrib(String sAttribPath) |

### getChangedAttributesForMultiOccurrence

| | |
|---|---|
| Description | Returns a HashMap that contains 4 String[] mapped to keys DELETED_OLD, ADDED_NEW, MODIFIED_OLD, MODIFIED_NEW. Used the XXX_OLD on oldEntry and the XXX_NEW on this entry (the new entry). This method determines the differences between the attributes of another ENTRY for multi-occurrence(grouping and non-grouping) ENTRIES. ADDED_NEW and DELETED_OLD will only include multi-occurrence attributes (groupings and non-groupings). Note on multi-occurrence non-groupings: The MODIFIED_NEW and _MODIFIED_OLD_lists will never include any multi-occurrence non-groupings, as multi-occurrence for non-groupings will only show up as Deleted or Added. Please consult documentation for more details. |
| Prototype | HashMap Entry::getChangedAttributesForMultiOccurrence (Entry oldEntry) |

### getDestinationEntrySetForRelatedEntries

| | |
|---|---|
| Description | Returns EntrySet with all entries this entry is related to filtering by container if filter Container is provided. |
| Prototype | EntrySet Entry::getDestinationEntrySetForRelatedEntries( Container filterContainer) |

### getDisplayValue

| | |
|---|---|
| Description | Returns the display value for the entry. If no display value is available then the primary key value is returned. |
| Prototype | String Entry::getDisplayValue(Locale locale) |

### getEntryAttrib

| | |
|---|---|
| Description | Returns the value of the attribute sAttribPath (spec_name/attribute_name) of this entry |
| Prototype | Object Entry::getEntryAttrib(String sAttribPath) |

### getEntryAttribModificationTime

| | |
|---|---|
| Description | Returns the time when the attribute sAttribPath (spec_name/attribute_name) of this entry was last modified |
| Prototype | Date Entry::getEntryAttribModificationTime(String sAttribPath) |

### getEntryAttribModifier

| | |
|---|---|
| Description | Returns the user who last modified the attribute sAttribPath (spec_name/attribute_name) of this entry |
| Prototype | String Entry::getEntryAttribModifier(String sAttribPath) |

### getEntryAttribs

| | |
|---|---|
| Description | Returns a HashMap mapping the paths (spec_name/attribute_name) of attributes to their respective values |
| Prototype | HashMap Entry::getEntryAttribs() |

### getEntryAttribsList

| | |
|---|---|
| Description | Returns an array of String containing the paths (spec_name/attribute_name) of all the attributes of this entry |
| Prototype | String[] Entry::getEntryAttribsList() |

### getEntryContainer

| | |
|---|---|
| Description | Gets the holding container for this Entry. Could be a catalog or category tree. Use isEntryAnItem to determine which one. |
| Prototype | Object Entry::getEntryContainer() |

### getEntryId

| | |
|---|---|
| Description | Returns this entry's ID |
| Prototype | Integer Entry::getEntryId() |

### getEntryRelatedItemInfo

| | |
|---|---|
| Description | Returns an array of length 2 containing: [0]=Related Item's Catalog's Name, [1]=Related Item's Primary Key, for the related item represented by the given internal unique item id, at the browsing version of the catalog of the given entry |
| Prototype | String[] Entry::getEntryRelatedItemInfo(int iItemId) |

### getEntrySaveResult

| | |
|---|---|
| Description | Returns the result of the last save called on this entry. Returns one of the following strings {ADDED,DELETED,MODIFIED,UNKNOWN} |
| Prototype | String Entry::getEntrySaveResult() |

### getEntrySetForPrimaryKeys

| | |
|---|---|
| Description | Returns an EntrySet of the entries in this container for the given primary keys - set bOptimize to true if you don't plan on changing the entries, the entry set is then optimized but this items don't keep track of changed attributes |
| Prototype | EntrySet Container::getEntrySetForPrimaryKeys(Array |

| | |
|---|---|
| | pkeys, Boolean bOptimize) |

### getEntrySetSize

| | |
|---|---|
| Description | Returns the number of entries in an entry set |
| Prototype | Integer EntrySet::getEntrySetSize() |

### getEntryStatus

| | |
|---|---|
| Description | Returns the status of the entry |
| Prototype | String Entry::getEntryStatus() |

### getFlatEntryNodes

| | |
|---|---|
| Description | Returns an array of flat entryNodes of this entry |
| Prototype | EntryNode[] Entry::getFlatEntryNodes() |

### getFlatPrimaryEntryNodes

| | |
|---|---|
| Description | Returns an array of flat primary entryNodes of this entry |
| Prototype | EntryNode[] Entry::getFlatPrimaryEntryNodes() |

### getFlatSecondaryEntryNodes

| | |
|---|---|
| Description | Returns an array of flat secondary entryNodes of this entry |
| Prototype | EntryNode[] Entry::getFlatSecondaryEntryNodes() |

### getItemsInheritingDataForPath

| | |
|---|---|
| Description | Returns a array of pairs consisting of the Catalog Name and Primary Key of Items that may be inheriting data from the given Entry. Array has -- catalogName, PrimaryKey! |
| Prototype | Object[][] Entry::getItemsInheritingDataForPath(String sAttribPath) |

### getOriginalEntry

| | |
|---|---|
| Description | Returns the original picture of the entry as stored in the database. If the entry is new or deleted, this operation returns null. |
| Prototype | Entry Entry::getOriginalEntry() |

### getRootEntryNode

| | |
|---|---|
| Description | Return the root entry node for this entry |
| Prototype | EntryNode Entry::getRootEntryNode() |

### getSourceEntrySetForRelatedEntries

| | |
|---|---|
| Description | Returns EntrySet with all entries that have an attribute related to this entry filtering by container if filterContainer is provided. |
| Prototype | EntrySet Entry::getSourceEntrySetForRelatedEntries(Container filterContainer) |

### isEntryAnItem

| | |
|---|---|
| Description | Returns TRUE if this entry is an Item, FALSE if it is a Category. |
| Prototype | Boolean Entry::isEntryAnItem() |

### isEntryCheckedOut

| | |
|---|---|
| Description | Returns true if the entry is checked out into a collaboration area otherwise it returns false. |
| Prototype | Boolean Entry::isEntryCheckedOut() |

### populateAllNonPersisted

| | |
|---|---|
| Description | Execute non-persisted script for all entrynodes in the entry. Return true if the script was completed succesfully, false otherwise. |
| Prototype | Boolean Entry::populateAllNonPersisted() |

### setEntryAttrib

| | |
|---|---|
| Description | Sets the attribute sAttribPath (spec_name/attribute_name) of this entry to sValue. Perform optional checks before update if bDoChecks is true. |
| Prototype | void Entry::setEntryAttrib(String sAttribPath, Object sValue, [Boolean bDoChecks] |

### setEntryRelationshipAttrib

| | |
|---|---|
| Description | Sets the attribute sAttribPath (spec_name/attribute_path) of type RELATIONSHIP of this entry to the related item |

| | |
|---|---|
| | represented by the given catalog and primary key |
| Prototype | void Entry::setEntryRelationshipAttrib(String sAttribPath, Catalog relatedItemCtg, String sRelatedItemPrimaryKey) |

### setEntryRelationshipAttribUsingItem

| | |
|---|---|
| Description | Sets the attribute sAttribPath (spec_name/attribute_path) of type RELATIONSHIP of this entry to the related item given |
| Prototype | void Entry::setEntryRelationshipAttribUsingItem(String sAttribPath, Item relatedItem) |

### setEntryStatus

| | |
|---|---|
| Description | Sets the status of the entry |
| Prototype | void Entry::setEntryStatus(String status) |

## EntryNode

### deleteEntryNode

| | |
|---|---|
| Description | Remove this entry node from the item. Please note that at least one occurrence of a node needs to be in the entry picture in order to display correctly all the nodes of the chosen attribute collections. The entry picture is the entry node tree image in memory. When you delete the last occurrence of a node, you don't delete the node per se, but rather you're setting its value to null. |
| Prototype | void EntryNode::deleteEntryNode() |

### getEntry

| | |
|---|---|
| Description | Returns the Entry behind the EntryNode. |
| Prototype | Entry EntryNode::getEntry() |

### getEntryNode

| | |
|---|---|
| Description | Return the first entry node matching the path sPath |

| | |
|---|---|
| Prototype | EntryNode EntryNode::getEntryNode(String sPath) |

### getEntryNodeChildren

| | |
|---|---|
| Description | Return the children of this EntryNode |
| Prototype | EntryNode[] EntryNode::getEntryNodeChildren() |

### getEntryNodeExactPath

| | |
|---|---|
| Description | Returns the exact path of this entry node - the following is always true: rootNode.getEntryNode(entryNode.getEntryNodeExactPath()) == entryNode |
| Prototype | String EntryNode::getEntryNodeExactPath() |

### getEntryNodeInheritedValue

| | |
|---|---|
| Description | Return the value of this EntryNode |
| Prototype | Object EntryNode::getEntryNodeInheritedValue() |

### getEntryNodeInheritedValueSourceEntryUniqueID

| | |
|---|---|
| Description | Returns a system wide unique ID for the entry where this value is inherited from. Returns null otherwise. |
| Prototype | String EntryNode::getEntryNodeInheritedValueSourceEntryUniqueID() |

### getEntryNodeInheritedDataContainerName

| | |
|---|---|
| Description | Returns the name of the container of the inherited data. Returns null if there is no inherited data. |
| Prototype | String EntryNode::getEntryNodeInheritedDataContainerName() |

### getEntryNodePath

| | |
|---|---|
| Description | Returns the Spec Node path of this entry node, NOT the relative path of this attr. |
| Prototype | String EntryNode::getEntryNodePath() |

### getEntryNodes

| | |
|---|---|
| Description | Return the entry nodes matching the path sPath |
| Prototype | EntryNode[] EntryNode::getEntryNodes(String sPath) |

### getEntryNodeValue

| | |
|---|---|
| Description | Return the value of this EntryNode |
| Prototype | Object EntryNode::getEntryNodeValue() |

### getNodeFromEntryNode

| | |
|---|---|
| Description | Returns the Node object for this entry node. |
| Prototype | Node Entrynode::getNodeFromEntryNode() |

### getNodePath

| | |
|---|---|
| Description | Returns the path of this node. |
| Prototype | String Node::getNodePath() |

### hasInheritedValue

| | |
|---|---|
| Description | Returns TRUE if this EntryNode has an inherited value (non-null value in category or catalog item associated via an inheritance rule). Returns FALSE if there is no inherited data. |
| Prototype | Boolean EntryNode::hasInheritedValue() |

### hasNonInheritedValue

| | |
|---|---|
| Description | Returns TRUE if there is a non-inherited value. The presence or absence of inherited values makes no difference |
| Prototype | Boolean EntryNode::hasNonInheritedValue() |

### isEntryNodeInheritedDataFromItem

| | |
|---|---|
| Description | Returns TRUE if the inherited data is from an Item. Returns false if there is no inherited data, or data is from a Category. |
| Prototype | Boolean Entrynode::isEntryNodeInheritedDataFromItem( ) |

### populateNonPersistedForEntryNode

| | |
|---|---|
| Description | Execute non-persisted script for this entrynode. Return true if the script was completed successfully, false otherwise. |
| Prototype | Boolean EntryNode::populateNonPersistedForEntryNode () |

### setEntryNode

| | |
|---|---|
| Description | Return the entryNode with path sPath relative to EntryNode, building any node needed along the path, or null if the path is invalid |
| Prototype | EntryNode EntryNode::setEntryNode(String sPath) |

### setEntryNodeRelationshipValue

| | |
|---|---|
| Description | Set the value of this EntryNode of type RELATIONSHIP to the related item represented by the given catalog and primary key |
| Prototype | void EntryNode::setEntryNodeRelationshipValue(Catalog relatedItemCtg, String sRelatedItemPrimaryKey) |

### setEntryNodeRelationshipValueUsingItem

| | |
|---|---|
| Description | Set the value of this EntryNode of type RELATIONSHIP to the related item given |
| Prototype | void EntryNode::setEntryNodeRelationshipValueUsingItem(Item relatedItem) |

### setEntryNodeValue

| | |
|---|---|
| Description | Set the value of this EntryNode and return it |
| Prototype | Object EntryNode::setEntryNodeValue(Object value) |

## Entry Set

### forEachEntrySetElement

| | |
|---|---|
| Description | Executes the statements for each (oEntry) in the entrySet |
| Prototype | forEachEntrySetElement(EntrySet entrySet, Object oEntry) { statements } |

## UserDefinedLog

### dumpUserDefinedLog

| | |
|---|---|
| Description | Dump all log entries from the user defined log to the Writer provided in no specific order. out - this is the output writer you want to dump the UDL to delim - the delimiter used for the current UDL entries outputType - one of COPY_UDE_OUTPUT, CSV_OUTPUT, XML_OUTPUT COPY_UDE_OUTPUT: dump each UDL entry exactly how it is currently stored CSV_OUTPUT: dump each UDL entry as comma seperated values XML_OUTPUT: dump each UDL entry within XML tags; docTag and hmNodeTags must also be specified docTag - this will comprise the XML tag surrounding the UDL dump hmNodeTags - this is the array of labels for each subtag to surround each delimited value |
| Prototype | void UserDefinedLog::dumpUserDefinedLog(Writer out, String delim, String outputType, String docTag, HashMap hmNodeTags) |

### insertUserDefinedLog

| | |
|---|---|
| Description | Persist the new user defined log object to the database. |
| Prototype | void UserDefinedLog::insertUserDefinedLog() |

### newUserDefinedLog

| | |
|---|---|
| Description | Returns a new user defined log object for this container with the given name and description. an exception ifWill throw a log with the same name already exists for the container. |
| Prototype | UserDefinedLog Container::newUserDefinedLog(String name, |

| | |
|---|---|
| | String description, Boolean isRunningLog |

### saveUserDefinedLog

| | |
|---|---|
| Description | Update the persisted user defined log object in the database. |
| Prototype | void UserDefinedLog::saveUserDefinedLog() |

### startBatchProcessingForUserDefinedLog

| | |
|---|---|
| Description | Setup batch processing for the given User Defined Log. This operation is to be used mainly during import/mass update jobs. |
| Prototype | void UserDefinedLog::startBatchProcessingForUserDefinedLog() |

### stopBatchProcessingForUserDefinedLog

| | |
|---|---|
| Description | Stop batch processing for the given User Defined Log. This operation is to be used mainly during import/mass update jobs. |
| Prototype | void UserDefinedLog::stopBatchProcessingForUserDefinedLog() |

### userDefinedLogAddEntry

| | |
|---|---|
| Description | Add an entry to the user defined log. If a message is specified, set that for the UserDefinedLogEntry |
| Prototype | void UserDefinedLog::userDefinedLogAddEntry(Entry entry, [String log_message]) |

### userDefinedLogDelete

| | |
|---|---|
| Description | Remove the user defined log object from the database. This action will also drop all entries to the log. |
| Prototype | void UserDefinedLog::userDefinedLogDelete() |

### userDefinedLogDeleteEntriesFor

| | |
|---|---|
| Description | Delete all log entries for an entry from the user-defined log. |
| Prototype | void UserDefinedLog::userDefinedLogDeleteEntriesF |

or(IEntry entry)

### userDefinedLogDeleteEntry

| | |
|---|---|
| Description | Delete a particular entry from the user defined log. |
| Prototype | void UserDefinedLog::userDefinedLogDeleteEntry(UserDefinedLogEntry entry) |

### userDefinedLogGetContainer

| | |
|---|---|
| Description | Get the container that is logged by the user defined log. |
| Prototype | Container UserDefinedLog::userDefinedLogGetContainer() |

### userDefinedLogGetDescription

| | |
|---|---|
| Description | Get the description of the user defined log. |
| Prototype | String UserDefinedLog::userDefinedLogGetDescription() |

### userDefinedLogIsRunningLog

| | |
|---|---|
| Description | Returns whether this user defined log is a running-log. |
| Prototype | Boolean UserDefinedLog::userDefinedLogIsRunningLog() |

### userDefinedLogSetName

| | |
|---|---|
| Description | Set the name of the user defined log. NOTE: You need to call insertUserDefinedLog/saveUserDefinedLog to persist this change. |
| Prototype | void UserDefinedLog::userDefinedLogSetName(String name) |

## UserDefinedLogEntry

### newUserDefinedLogEntry

| | |
|---|---|
| Description | Returns a new user defined log entry object with for the specified entry which is either an item or category (with date/timestamp and log |
| Prototype | newUserDefinedLogEntry(Date date, Container container, Entry entry, String log) |

### newUserDefinedLogEntry

| | |
|---|---|
| Description | Returns a new user defined log entry object with for the specified entry which is either an item or category (with date/timestamp and log |
| Prototype | newUserDefinedLogEntry(Date date, Container container, Entry entry, String log) |

### userDefinedLogEntryGetTarget

| | |
|---|---|
| Description | Get the entry object of the user defined log entry. |
| Prototype | Entry UserDefinedLogEntry::userDefinedLogEntryGetTarget() |

### userDefinedLogEntrySetDate

| | |
|---|---|
| Description | Set the date of the user defined log entry. |
| Prototype | void UserDefinedLogEntry::userDefinedLogEntrySetDate(Date date) |

### userDefinedLogEntrySetValue

| | |
|---|---|
| Description | Set the log of the user defined log entry. |
| Prototype | void UserDefinedLogEntry::userDefinedLogEntrySetValue(String log_message) |

# InputOutput

## Feed

### createDataSource

| | |
|---|---|
| Description | Creates a Data Source of the type ("PULL_FTP", "PULL_FTP", "PUSH_WWW", "DOC_STORE") |

| | |
|---|---|
| | with given name. Will not create if a source with same name already exists. extraAttribs can be used to set other attributes of the datasource like "SERVER_ADDRESS", "SERVER_PORT", "USERNAME", "PASSWORD", "FILENAME", "DIRECTORY", "DOC_STORE_PATH" |
| Prototype | Creates a Data Source of the type (ONLY PUSH_WWW supported) with given name. Will not create if a source with same name already exists |

### createExport

| | |
|---|---|
| Description | Creates the Export with given params. An optional parameter "charsetName", which may be set in the "optionalArgs" parameter, describes the file encoding of the export. Otherwise, the Cp1252 is chosen as the default file encoding. Returns Done if successful, Error if not. Here is a complete list of the optional arguments which may be set in the "optionalArgs" parameter: String approverUserName, String charsetName, String distributionName, String selectionName, String synType, String diffType,String sParamsDocPath |
| Prototype | String createExport(String marketSpecName, String catalogName, String specMapName, String exportScriptName, String syndicationName, [HashMap optionalArgs]) |

### createImport

| | |
|---|---|
| Description | Creates the Feed with given params. An optional argument sCharsetName, which may be defined in the optionalArgs HashMap, describes the file encoding of the feed. Otherwise, Cp1252 is chosen as the default file encoding. Also, optional parameters to describe if the current container is a collaboration area, and the step path of the workflow step in to which the feed is to be done, could be specified. Returns Done if successful, Error if not. The complete list of optional arguments, which may be set in the optionalArgs parameter, is as follows: String sCharsetName, Boolean bIsCollaborationArea, String sWflStepPath, String sParamsDocPath, String sImportSemantic, and String |

| | |
|---|---|
| | sApproverUserName. |
| Prototype | String createImport(String sImportName, String sImportType, String sSourceName, String sFileSpecName, String sCatalogName, String sSpecMapName, String sCategoryTreeName, String sScriptName, String sACGName, [HashMap optionalArgs]) |

### getExportItemsCount

| | |
|---|---|
| Description | Returns the number of items being syndicated |
| Prototype | `Integer getExportItemsCount()`<br>Note: This operation replaced getSyndicationItemsCount (deprecated in v3.3.1) |

### getExportItemSets

| | |
|---|---|
| Description | Returns an array of ItemSets being syndicated |
| Prototype | `ItemSet[] getExportItemSets()` |

### getFtp

| | |
|---|---|
| Description | Use to get a file via FTP. The seventh parameter set where WPC will store the retrieved file. The eighth and the ninth paramters together are optional. The eigth parameter gets the FTP Operation Status and the ninth paramter ensures that the FTP operations are logged. Returns the result as true/false if the eighth and the ninth are not specified otherwise a HashMap is returned. If a true/false is returned, it indicates if the ftp was a success/failure. If the size of the retrieved file is not the same as the size of the remote file the status is set to false. If a HashMap is returned, the first paramater is the true/false which indicates success/failure, the second paramater is the message string of the FTP Operation Status and the third parameter is the FTP Operation error code |
| Prototype | HashMap/Boolean getFtp(String hostname, String port, String userid, String password, String path, String filename, String sDocStorePath, Boolean deleteRemoteFile, [Boolean detailedTransferStatus, Boolean loggingEnabled]) |

### startAggregationByName

| | |
|---|---|
| Description | Run the feed called sName on the file sDocPath |
| Prototype | void startAggregationByName(String sName, String sDocPath) |

## Export environment

### new$EnvObjectList

| | |
|---|---|
| Description | Returns a container for the WPC objects to be exported. This class is used to add and retrieve the objects to be exported |
| Prototype | new EnvObjectList() |

### setTypeToExport

Description

Sets the object type to be exported. List of acceptable values for sObjectType are:

- ACG
- ATTRIBUTE_COLS
- CATALOG
- CATALOG_CONTENT
- CATALOG_VIEW
- COLLABORATION_AREA
- COLLABORATION_AREA_CONTENT
- COMPANY_ATTRIBUTES
- CONTAINER_ACCESSPRV
- DATASOURCE
- DESTINATION_SPEC
- DISTRIBUTION
- DOC_STORE
- EXPORTS
- FEEDS
- FILE_SPEC
- HIERARCHY
- HIERARCHY_CONTENT
- HIERARCHY_MAPS
- HIERARCHY_VIEW
- INHERITANCE_RULES
- ITEM_CATEGORY_MAPS
- JOBS
- LOOKUP_TABLE
- LOOKUP_TABLE_CONTENT

- LOOKUP_TABLE_SPEC
- MAPS
- MY_SETTINGS
- PRIMARY_SPEC
- REPORTS
- ROLES
- SCRIPT_INPUT_SPEC
- SECONDARY_SPEC
- SPEC
- SUB_SPEC
- USERS
- WORKFLOW

Prototype

void EnvObjectList::setTypeToExport(String sObjectType)

### addObjectByNameToExport

Description

Sets the entity to be exported by specifying the entity name as an argument. sObjectType is optional. In case of Catalog and Hierarchy Content export, this operation is used to specify the attribute collection associated with the object. In case of DocStore partial export, this operation is used to specify the DocStore path. List of acceptable values for sObjectType are:

- ACG
- ATTRIBUTE_COLS
- CATALOG
- CATALOG_VIEW
- COLLABORATION_AREA
- COLLABORATION_AREA_CONTENT
- COMPANY_ATTRIBUTES
- CONTAINER_ACCESSPRV
- DATASOURCE
- DESTINATION_SPEC
- DISTRIBUTION
- DOC_STORE
- EXPORTS
- FEEDS
- FILE_SPEC
- HIERARCHY
- HIERARCHY_VIEW
- INHERITANCE_RULES
- JOBS
- LOOKUP_TABLE

- LOOKUP_TABLE_CONTENT
- LOOKUP_TABLE_SPEC
- MAPS
- MY_SETTINGS
- PRIMARY_SPEC
- REPORTS
- ROLES
- SCRIPT_INPUT_SPEC
- SECONDARY_SPEC
- SPEC
- SUB_SPEC
- USERS
- WORKFLOW

Prototype

void EnvObjectList::addObjectByNameToExport(String sEntityName[, String sObjectType])

### addAllObjectsToExport

Description

Notifies that all the entities of specific object type be exported. sObjectType is optional. List of acceptable values for sObjectType are:

- ACG
- ATTRIBUTE_COLS
- CATALOG
- CATALOG_CONTENT
- CATALOG_VIEW
- COLLABORATION_AREA
- COLLABORATION_AREA_CONTENT
- COMPANY_ATTRIBUTES
- CONTAINER_ACCESSPRV
- DATASOURCE
- DESTINATION_SPEC
- DISTRIBUTION
- DOC_STORE
- EXPORTS
- FEEDS
- FILE_SPEC
- HIERARCHY
- HIERARCHY_CONTENT
- HIERARCHY_MAPS
- HIERARCHY_VIEW
- INHERITANCE_RULES
- ITEM_CATEGORY_MAPS
- JOBS
- LOOKUP_TABLE
- LOOKUP_TABLE_CONTENT

- LOOKUP_TABLE_SPEC
- MAPS
- MY_SETTINGS
- PRIMARY_SPEC
- REPORTS
- ROLES
- SCRIPT_INPUT_SPEC
- SECONDARY_SPEC
- SPEC
- SUB_SPEC
- USERS
- WORKFLOW

| | |
|---|---|
| Prototype | void EnvObjectList::addAllObjectsToExport([String sObjectType]) |

### setCatalogByNameToExport

| | |
|---|---|
| Description | Sets the Catalog whose contents are to be exported |
| Prototype | void EnvObjectList::setCatalogByNameToExport(String sCatalog) |

### setItemCategoryMapToExport

| | |
|---|---|
| Description | Sets the Catalog and Hierarchy whose Item-Category mappings need to be exported |
| Prototype | void EnvObjectList::setItemCategoryMapToExport(String sCatalog, String sHierarchy) |

### setHierarchyMapToExport

| | |
|---|---|
| Description | Sets the source and destination Hierarchies whose mappings need to be exported |
| Prototype | void EnvObjectList::setHierarchyMapToExport(String sourceHierarchy, String destHierarchy) |

### getCatalogNameToExport

| | |
|---|---|
| Description | Returns the last value set with setCatalogByNameToExport |
| Prototype | String EnvObjectList::getCatalogNameToExport() |

### getHierarchyNameToExport

| | |
|---|---|
| Description | Returns the last value set with setHierarchyByNameToExport |
| Prototype | String EnvObjectList::getHierarchyNameToExport() |

### getTypeToExport

| | |
|---|---|
| Description | Returns the last object type set with setTypeToExport |
| Prototype | String EnvObjectList::getTypeToExport() |

### getTypesToExport

| | |
|---|---|
| Description | Returns all the object types, set with setTypeToExport, for exporting |
| Prototype | String[] EnvObjectList::getTypesToExport() |

### exportEnv

| | |
|---|---|
| Description | Exports the WPC objects specified in envObjList at the specified DocStore path. sDocFilePath is the filepath of the zip file that will be exported into the document store - returns the log as a string. |
| Prototype | String exportEnv(EnvObjectList envObjList, String sDocFilePath) |

### importEnv

| | |
|---|---|
| Description | Imports the content of the archive in the docstore at sDocFilePath into this company. - returns the log as a string. |
| Prototype | String importEnv(String sDocFilePath) |

## Messaging

### General

### getDescription

| | |
|---|---|
| Description | Returns the description of result |
| Prototype | int UtResult::getDescription() |

### getMessage

| | |
|---|---|
| Description | Returns the message of result |
| Prototype | int UtMessage::getMessage() |

### getMsgQueueName

| | |
|---|---|
| Description | Returns the name of this message queue. |
| Prototype | String MsgQueue::getMsgQueueName() |

### getStatus

| | |
|---|---|
| Description | Returns the status of result |
| Prototype | Integer UtResult::getStatus() |

### getWebServiceByName

| | |
|---|---|
| Description | Returns the web service with the given name. If there is no such web service, returns null. |
| Prototype | String MsgQueue::getMsgQueueName() |

### new$UCCnetTransporter

| | |
|---|---|
| Description | Builds a new UCC-Net Transporter object; The properties file is defined by the system. |
| Prototype | new UCCnetTransporter() |

### receiveResponse

| | |
|---|---|
| Description | Returns the result of sending the message, the value of reply gets set |
| Prototype | UtResult UCCnetTransporter::receiveResponse(String id, UtMessage reply) |

### setMessage

| | |
|---|---|
| Description | Sets the message |
| Prototype | UtMessage::setMessage(String message) |

### submitRequest

| | |
|---|---|
| Description | Returns the result of sending the message |
| Prototype | UtResult UCCnetTransporter::submitRequest(String id, String docPath) |

## Sample Script for Invoking SOAP Server

```
paramNames =
[];paramNames.add("sUser");paramNames.add("sCmpCode");paramNames.add("sScript");
paramValues =
[];paramValues.add("Admin");paramValues.add("WPC");paramValues.add("list=getCatalogName
sList();
out.writeln(\"list: \" + list); i=5; out.writeln(\"res: \" + i*2);");
res = invokeSoapServer("http://foxy:9100/soap/servlet/rpcrouter", "urn:Script",
"executeInlineScript", paramNames, paramValues);
out.println(res);
```

# UCCnet Operations

### getDescription

| | |
|---|---|
| Description | Returns the description of result |
| Prototype | int UtResult::getDescription() |

### getMessage

| | |
|---|---|
| Description | Returns the message of result |
| Prototype | int UtMessage::getMessage() |

### getStatus

| | |
|---|---|
| Description | Returns the status of result |
| Prototype | Integer UtResult::getStatus() |

### new UCCnetTransporter

| | |
|---|---|
| Description | Builds a new UCC-Net Transporter object; The properties file is defined by the system. |
| Prototype | new UCCnetTransporter() |

### receiveResponse

| | |
|---|---|
| Description | Returns the result of sending the message, the value of reply gets set |
| Prototype | UtResult UCCnetTransporter::receiveResponse(String id, UtMessage reply) |

### setMessage

| | |
|---|---|
| Description | Sets the message |
| Prototype | UtMessage::setMessage(String message) |

## MQ

### mqDisconnect

| | |
|---|---|
| Description | Disconnects from the given queue manager. |
| Prototype | void MQQueueManager::mqDisconnect() |

### mqGetMessageDiagnostics

| | |
|---|---|
| Description | Returns a string containing diagnostic information about the given message. |
| Prototype | String mqGetMessageDiagnostics(MQMessage message) |

### mqGetMessageId

| | |
|---|---|
| Description | Returns the ID of the given message as a String containing a hexadecimal number. |
| Prototype | String MQMessage::mqGetMessageId() |

### mqGetQueueMgr

| | |
|---|---|
| Description | Creates and returns a new MQ queue manager with the given properties. |
| Prototype | MQQueueManager mqGetQueueMgr(String hostname, String port, String channel, String queueMgrName) |

### mqGetReceivedMsg

| | |
|---|---|
| Description | Receives a message from queueName or picks the default inbound queue if queueName not specified. Returns the message, as a MQMessage, or null. |
| Prototype | MQMessage MQQueueManager::mqGetReceivedMsg(String queueName, String queueOpenOptions, String messageGetOptions) |

### mqGetReceivedMsgByMessageID

| | |
|---|---|
| Description | Finds the message in the given queue with given message ID. The ID is passed in a a String containing a hexadecimal number. Returns null if there is no such message in the given queue. |
| Prototype | MQMessage MQQueueManager::mqGetReceivedMsgByMessageID(String queueName, String messageId, String passedInQueueOpenOptions, String passedInMessageGetOptions) |

### mqGetResponseToMsg

| | |
|---|---|
| Description | Gets the response to the given message from the given queue. |
| Prototype | MQMessage MQQueueManager::mqGetResponseToMsg(MQMessage outgoingMessage, String queueOptions, String messageOptions) |

### mqGetTextFromMsg

| | |
|---|---|
| Description | Returns a string containing the entire content of a MQMessage, including headers. |
| Prototype | String mqGetTextFromMsg(MQMessage mqMessage) |

### mqGetXMLMessageContent

| | |
|---|---|
| Description | Discards any garbage at the beginning of the input string to get a XML document. |
| Prototype | String mqGetXMLMessageContent(String orgXmlMsg) |

### mqSendReply

| | |
|---|---|
| Description | Sends a reply to the given message, without indicating success or failure. |
| Prototype | MQMessage MQQueueManager::mqSendReply(MQMessage receivedMsg, String msgText, String passedInQueueOpenOptions, String passedInMessagePutOptions) |

### mqSendReplyWithStatus

| | |
|---|---|
| Description | Sends a reply to the given message, setting the feedback field to indicate the given status. Status must be one of the following (in upper or lower case): SUCCESS, FAIL, VALCHANGE, VALDUPES, MULTIPLE_HITS, FAIL_RETRIEVE_BY_CONTENT, BO_DOES_NOT_EXIST, UNABLE_TO_LOGIN, APP_RESPONSE_TIMEOUT, NONE. |
| Prototype | MQMessage MQQueueManager::mqSendReplyWithStatus(MQMessage receivedMsg, String msgText, String status, String passedInQueueOpenOptions, String passedInMessagePutOptions) |

### mqSendTextMsg

| | |
|---|---|
| Description | Sends a message provided in the String msgText over queueName. Returns the MQMessage |
| Prototype | MQMessage MQQueueManager::mqSendTextMsg(String msgText, String queueName, String queueOpenOptions, String messagePutOptions) |

### mqSendTextMsgWithReply

| | |
|---|---|
| Description | Sends a message provided in the String msgText over queueName. The reply queue is specified. Returns the MQMessage object. |
| Prototype | MQQueueManager::mqSendTextMsgWithReply( String msgText, String queueName, String replyQueueName, String queueOpenOptions, String messagePutOptions) |

## JMS Operations

### jmsDisconnect

| | |
|---|---|
| Description | Disconnects from the given queue manager. |
| Prototype | void QueueSession::jmsDisconnect(QueueConnection qcon) |

### jmsCreateTextMsg

| | |
|---|---|
| Description | Creates a new JMS TextMessage using QueueSession information with the text provided. |
| Prototype | JMSMessage QueueSession::jmsCreateTextMsg(String msgText) |

### jmsGetContext

| | |
|---|---|
| Description | Creates a JMS context. |
| Prototype | Context jmsGetContext(String url, String jndiFactory) |

### jmsGetConnectionFactory

| | |
|---|---|
| Description | Creates and returns a jms connection factory with the specified context. |
| Prototype | QueueConnectionFactory Context::jmsGetConnectionFactory(String jmsFactory) |

### jmsGetMQConnectionFactory

| | |
|---|---|
| Description | Creates and returns a JMS connection factory for communicating with MQ queues. Note that you do not need a Context to get an MQ connection factory whereas you need a Context for connecting to other JMS queues. |
| Prototype | QueueConnectionFactory jmsGetMQConnectionFactory(String mqQueueManager, String mqHostname, String mqChannel, Integer mqPort) |

### jmsGetQueueByName

| | |
|---|---|
| Description | Returns a javax.jms.Queue object from the given JNDI Name and Context. |
| Prototype | javax.jms.Queue jmsGetQueueByName(Context ctx, String name) |

### jmsGetQueueConnection

| | |
|---|---|
| Description | Returns a JMS queue connection from the given connection factory. |

| | |
|---|---|
| Prototype | QueueConnection QueueConnectionFactory::jmsGetQueueConnection() |

### jmsGetQueueSession

| | |
|---|---|
| Description | Returns a JMS queue connection from the given connection factory |
| Prototype | QueueSession QueueConnection::jmsGetQueueSession() |

### jmsDisconnect

| | |
|---|---|
| Description | Disconnects from the given queue manager |
| Prototype | void QueueSession::jmsDisconnect(QueueConnection qcon) |

### jmsCreateTextMsg

| | |
|---|---|
| Description | Creates a new JMS TextMessage using QueueSession information with the text provided. |
| Prototype | Message QueueSession::jmsCreateTextMsg(String msgText) |

### jmsSendMsg

| | |
|---|---|
| Description | Sends a message MSG over queue with name queueName and returns MSG or null. If a MESSAGETOREPLYTO is provided, the reply to queue and message id are read from it. PROPERTIES is a map from string keys to string values. There are three special keys "TRIGO_REPLY_TO_QUEUE", "TRIGO_COPY_CORRELATION_ID_BYTES", and "TRIGO_COPY_CORRELATION_ID". If TRIGO_REPLY_TO_QUEUE is provided, then it overrides the QUEUENAME or replyto queue in MESSAGETOREPLYTO provided. replyto queue in MESSAGETOREPLYTO overrides QUEUENAME. "TRIGO_COPY_CORRELATION_ID" and "TRIGO_COPY_CORRELATION_ID_BYTES" copy over correlation id from MESSAGETOREPLYTO to MSG. Both can be |

| | |
|---|---|
| | provided. Their values need to be boolean (as opposed to strings - as described above) |
| Prototype | Message QueueSession::jmsSendMsg(Message msg, String queueName[, HashMap properties, Message messageToReplyTo]) |

### jmsSendMsgToQueue

| | |
|---|---|
| Description | Sends message MSG and returns MSG or null. The message is sent to the queue specified by OUTBOUNDQUEUE, unless OUTBOUNDQUEUE is null. If OUTBOUNDQUEUE is null, MSG is sent to the reply-to queue of MESSAGETOREPLYTO, if MESSAGETOREPLYTO is provided. If OUTBOUNDQUEUE is null and MESSAGETOREPLYTO is not provided, throws an AustinException. If MESSAGETOREPLYTO is provided, the message id is read from it. PROPERTIES is a map from string keys to string values. There is one special (non-JMS) key: TRIGO_INCOMING_REPLY_QUEUE. TRIGO_INCOMING_REPLY_QUEUE indicates a javax.jms.Queue object to which an external application should send replies to this message. |
| Prototype | JMSMessage QueueSession::jmsSendMsgToQueue(JMSMessage msg, javax.jms.Queue outboundQueue [, HashMap properties, JMSMessage messageToReplyTo,]) |

### jmsReceiveMsg

| | |
|---|---|
| Description | Receives next available message from queue QUEUENAME and times out after TIMEOUT milliseconds |
| Prototype | Message QueueSession::jmsReceiveMsg(String queueName, Integer timeout[, String messageSelector, Message messageToReceiveReplyFor]) |

### jmsReceiveMsgFromQueue

| | |
|---|---|
| Description | Receives a JMS Message. Times out after TIMEOUT milliseconds. If INBOUNDQUEUE is not null, looks on that queue. If INBOUNDQUEUE is null, and |

| | |
|---|---|
| | MESSAGETORECEIVEREPLYFOR is not null, looks on the queue defined in the Reply-To field of MESSAGETORECEIVEREPLYFOR. If INBOUNDQUEUE is null and MESSAGETORECEIVEREPLYFOR is null, throws an AustinException. We now know which queue will be used. If MESSAGESELECTOR and MESSAGETORECEIVEREPLYFOR are both null, selects the first message from that queue. Otherwise selects the first message from the queue (if any) fulfilling all of the conditions defined by MESSAGESELECTOR and MESSAGETORECEIVEREPLYFOR. If MESSAGETORECEIVEREPLYFOR is not null, rejects any message not having a correlation ID equal to MESSAGETORECEIVEREPLYFOR's message ID. If MESSAGESELECTOR is not null, rejects any message not fulfilling the condition defined in messageSelector. If no appropriate message is found, returns null. |
| Prototype | JMSMessage QueueSession::jmsReceiveMsgFromQueue(javax.jms.Queue queue, Integer timeout[, String messageSelector, JMSMessage messageToReceiveReplyFor]) |

### jmsGetTextFromMsg

| | |
|---|---|
| Description | Returns a string containing the entire content of a JMS message, including headers. |
| Prototype | String JMSMessage::jmsGetTextFromMsg() |

### jmsGetMessageID

| | |
|---|---|
| Description | Returns a string containing the JMS message id. |
| Prototype | String JMSMessage::jmsGetMessageID() |

### jmsGetMessageCorrelationID

| | |
|---|---|
| Description | Returns a string containing Correlation Id for the JMS message. |
| Prototype | String JMSMessage::jmsGetMessageCorrelationID() |

### jmsGetMessageProperties

| | |
|---|---|
| Description | Returns a hashmap from string property names to string values for those priorities. |
| Prototype | HashMap JMSMessage::jmsGetMessageProperties() |

### jmsSetMessageText

| | |
|---|---|
| Description | Sets the provided text for the JMS TextMessage. Only JMS TextMessage type is supported |
| Prototype | void Message::setJMSMessageText(String msgText) |

## Page Layout

### getPageLayoutByName

| | |
|---|---|
| Description | Returns the page layout object with the corresponding name |
| Prototype | IPageLayout getPageLayoutByName(String sPageLayoutName) |

### new$PageLayout

| | |
|---|---|
| Description | Returns a new page layout with the given name |
| Prototype | new PageLayout(String sPageLayoutName) |

### savePageLayout

| | |
|---|---|
| Description | Saves the current page layout |
| Prototype | void IPageLayout::savePageLayout() |

## Reader

### forEachLine

| | |
|---|---|
| Description | Executes the statements for each line read from in |
| Prototype | forEachLine(BufferedReader in, String line) { statements } |

### getCurrentLine

| | |
|---|---|
| Description | Returns the current line |

| | |
|---|---|
| Prototype | String getCurrentLine() |

### getFullHTTPResponse

| | |
|---|---|
| Description | Returns a HashMap (with RESPONSE_READER and RESPONSE_HEADER_FIELDS) for the response for posting hmParameters or a doc of sContentType against the server at url, Use hmRequestProperties to send specific header information. An optional parameter bGetReader could be used to specify if the function needs to also return the response reader (default is true). An optional parameter bPostUserInfo could be used to specify if the function would need to post the invoking user information (default is false). The response is optionally stored into a document at sDocStorePath in the docstore. |
| Prototype | HashMap getFullHTTPResponse(String url, HashMap hmRequestProperites, HashMap hmParameters, String sRequestMethod , [ [String sEncoding], [Doc doc, String sContentType], [boolean bGetResponseReader, boolean bPostUserInfo], [String sDocStorePath] ]) |

### getHTTPResponse

| | |
|---|---|
| Description | Returns a reader for the response for posting hmParameters against the server at url, Use hmRequestProperties to send specific header information |
| Prototype | BufferedReader getHTTPResponse(String url, HashMap hmRequestProperites, HashMap hmParameters, String sRequestMethod [,String sEncoding]) |

### new$CSVParser

| | |
|---|---|
| Description | Returns a comma separated parser given the buffered reader |
| Prototype | new CSVParser(BufferedReader reader) |

### new$DelimParser

| | |
|---|---|
| Description | Returns a delimiter parser which parses based on the given delimiter |
| Prototype | new DelimParser(BufferedReader reader, String delimiter) |

| | |
|---|---|
| | delimiter) |

### new FixedWidthParser

| | |
|---|---|
| Description | Returns a fixed width parser given the buffered reader input. fieldPos are optional parameters which indicate the positions of the fields. |
| Prototype | FixedWidthParser newFixedWidthParser(BufferedReader input, [Integer fieldPos1, Integer fieldPos2, ..., Integer fieldPosN] |

### new Reader

| | |
|---|---|
| Description | Returns the buffered reader for the document specified by the path. You may optionally specify a charset that differs from the one stored with the document in the doc store. |
| Prototype | new Reader(String documentPath [, String charsetName]) |

### newCSVParser

| | |
|---|---|
| Description | Returns a Comma Separated Parser given the buffered reader input |
| Prototype | CSVParser newCSVParser(BufferedReader input) |

### newDelimParser

| | |
|---|---|
| Description | Returns a parser which parse based on the delimiter provided |
| Prototype | DelimiterParser newDelimParser(BufferedReader input, String delim) |

### newFixedWidthParser

| | |
|---|---|
| Description | Returns a fixed width parser given the buffered reader input |
| Prototype | FixedWidthParser newFixedWidthParser(BufferedReader input) |

### nextLine

| | |
|---|---|
| Description | Returns the next line from the reader |
| Prototype | String nextLine (BufferedReader in) |

# TarArchive

### addCtgFile

| | |
|---|---|
| Description | Use to add a supplier ctg file (including images) to a tar archive |
| Prototype | Boolean TarArchive::addCtgFile(String sFileName [, Boolean bUpperCaseName]) |

### closeTarArchive

| | |
|---|---|
| Description | Use to close a tar archive and upload to the docstore for future distributions. By default, the archive is deleted after the distribution, unless 'deleteAfterDistribution' is false. |
| Prototype | void TarArchive::closeTarArchive([Boolean deleteAfterDistribution]) |

### new$TarArchive

| | |
|---|---|
| Description | Returns a new tar archive with the given file name |
| Prototype | new TarArchive(String sFileName) |

# Writer

### close

| | |
|---|---|
| Description | Close this writer |
| Prototype | void Writer::close() |

### createOtherOut

| | |
|---|---|
| Description | Returns a new scriptfile output with the given name and an optional charset value. |
| Prototype | Writer createOtherOut(String name, [String charset] |

### print

| | |
|---|---|
| Description | Writes o as a string and appends a new line to it into this writer |
| Prototype | void Writer::print(Object o) |

### println

| | |
|---|---|
| Description | Writes o as a string and appends a new line to it into this writer |
| Prototype | void Writer::println(Object o) |

### printXML

| | |
|---|---|
| Description | Writes an XML tag with the text value sValue, the tag name sTagName and the attributes sAttributes |
| Prototype | void Writer::printXML(String sTagName, String sValue, String sAttributes) |

### save

| | |
|---|---|
| Description | Creates an Doc object with the content in the Writer and saves it in the specified documentPath |
| Prototype | Doc Writer::save(String documentPath) |

### setOutputAttribute

| | |
|---|---|
| Description | Set an attribute of this writer - which becomes an attribute of the document this writer is flushed into, if any |
| Prototype | void Writer::setOutputAttribute(String sAttributeName, String sAttribueValue) |

### setOutputName

| | |
|---|---|
| Description | Set the name of this writer - which becomes the name of the document this writer is flushed into, if any |
| Prototype | void Writer::setOutputName(String sName) |

### write

| | |
|---|---|
| Description | Writes o as a string into this writer |
| Prototype | void Writer::write(Object o) |

### writeBinaryFile

| | |
|---|---|
| Description | Pipes the dostore file represented sOrigFilePath into a new Doc of name sDestFileName in the directory of the current transaction instance |

| | |
|---|---|
| Prototype | void writeBinaryFile(String sDestFileName, String sOrigFilePath) |

### writeDoc

| | |
|---|---|
| Description | Appends doc as a string into this writer |
| Prototype | void Writer::writeDoc(IDoc doc) |

### writeFile

| | |
|---|---|
| Description | Pipes the dostore file represented sFilePath into this writer |
| Prototype | void Writer::writeFile(String sFilePath) |

### writeFileUsingOut

| | |
|---|---|
| Description | Pipes w into this writer |
| Prototype | void Writer::writeFileUsingOut(Writer w) |

### writeln

| | |
|---|---|
| Description | Writes o as a string and appends a new line to it into this writer |
| Prototype | void Writer::writeln(Object o) |

## XMLNode

### forEachXMLNode

| | |
|---|---|
| Description | Executes the statements for each XML node having the relative path xPath - paths in the block are relative to xPath. If the node variable is passed in as an argument, it is populated with the XMLNode that is being operated on in each iteration of forEachXMLNode |
| Prototype | forEachXMLNode(String xPath [, XMLNode node]) { statements } |

### getXMLNodeName

| | |
|---|---|
| Description | Returns the name of the current XMLNode. |
| Prototype | String XMLNode::getXMLNodeName() |

### getXMLNodePath

| | |
|---|---|
| Description | Returns the path of the current XMLNode. This path is not an XPath - it is the concatenation of all the names of the parent XMLNode's path, /, and the name of this XMLNode |
| Prototype | String XMLNode::getXMLNodePath() |

### getXMLNodeValue

| | |
|---|---|
| Description | Returns the value of the current XMLNode. |
| Prototype | String XMLNode::getXMLNodeValue(Boolean bRequired) |

### parseXMLNode

| | |
|---|---|
| Description | Returns the value given by the sXMLSubPath XPath in the current XML document |
| Prototype | String parseXMLNode (String sXMLSubPath) |

# Security

## Company

### getCompanyCode

| | |
|---|---|
| Description | Returns the company code of this company. |
| Prototype | String getCompanyCode() |

### getCompanyName

| | |
|---|---|
| Description | Returns the name of this company. |
| Prototype | String getCompanyName() |

## Role

### createRole

| | |
|---|---|
| Description | Creates a role object with the specified role name and an optional role descriptio. |
| Prototype | Role createRole(String sRoleName, [String sRoleDesc]) |

### getAccessControlGroupPrivsForRole

| | |
|---|---|
| Description | The return parameter is an array of privileges (which are defined in the format: Catalog__list, Selection__list, SelectionMembers__view_items etc.). |
| Prototype | String[] Role::getAccessControlGroupPrivsForRole(String acgName) |

### getAccessControlGroupsForRole

| | |
|---|---|
| Description | Gets the access control groups for the given role |
| Prototype | String[] Role::getAccessControlGroupsForRole() |

### getLocalesForRole

| | |
|---|---|
| Description | Gets the locales that this role has access to for all containers |
| Prototype | String Role::getLocalesForRole() |

### getRoleByName

| | |
|---|---|
| Description | Returns a role object for the specified role |
| Prototype | Role getRoleByName(String sRoleName) |

### getRoleDescription

| | |
|---|---|
| Description | Return the description of the role |
| Prototype | String Role::getRoleDescription() |

### getRoleName

| | |
|---|---|
| Description | Return the name of the role |
| Prototype | String Role::getRoleName() |

### getRoles

| | |
|---|---|
| Description | Returns all roles for the current company |
| Prototype | Role[] getRoles() |

### getRolesForCompany

| | |
|---|---|
| Description | Returns all roles of the given company |
| Prototype | Role[] getRolesForCompany(String sCmpCode) |

### getUsersFromRole

| | |
|---|---|
| Description | Returns all users within the Role |
| Prototype | User[] Role::getUsersFromRole() |

### setAccessControlGroupForRole

| | |
|---|---|
| Description | Sets an access control group with the given set of privileges for the role. The parameter privs is an array of privileges (which are picked from the strings in the format: Catalog__list, Selection__list, SelectionMembers__view_items etc.). Please note the the page privileges like PAGE_OBJ_CTG_CONSOLE__view, PAGE_OBJ_CAT_CREATE__view are stored only in the Default ACG. |
| Prototype | Boolean Role::setAccessControlGroupForRole(String acgName, String[] privs) |

### setAllAccessControlGroupForRole

| | |
|---|---|
| Description | Sets access control group acgName with all privileges except for the ones in privExclusions. |
| Prototype | void Role::setAllAccessControlGroupForRole(String acgName, [String[] privExclusions]) |

### setLocalesForRole

| | |
|---|---|
| Description | Sets the locales that this role has access to for all containers |
| Prototype | void Role::setLocalesForRole(String localesCSVString) |

## User

### cloneUser

| | |
|---|---|
| Description | Clones an existing user info into a new user. Password field is required. The optional roles and organization fields, when specified, override the roles and/or organization of the existing user. |
| Prototype | User ::cloneUser(String original_username, String username, String firstname, String |

| | |
|---|---|
| | String username, String firstname, String lastname, String email, Boolean enabled, String password, Category organization, [HashMap roles]) |

### getCurrentUserName

| | |
|---|---|
| Description | Returns the name of the current user |
| Prototype | String getCurrentUserName() |

### getUserAddress

| | |
|---|---|
| Description | Return the User's Address |
| Prototype | String User::getUserAddress() |

### getUserByUsername

| | |
|---|---|
| Description | Returns the User object for the given User Name |
| Prototype | User getUserByUsername(String sUserName, String sCmpCode) |

### getUserCompanyCode

| | |
|---|---|
| Description | Return the User's Company Code |
| Prototype | String User::getUserCompanyCode() |

### getUserCompanyName

| | |
|---|---|
| Description | Return the User's Company Name |
| Prototype | String User::getUserCompanyName() |

### getUserEmail

| | |
|---|---|
| Description | Return the User's Email Address |
| Prototype | String User::getUserEmail() |

### getUserFax

| | |
|---|---|
| Description | Return the User's Fax Number |
| Prototype | String User::getUserFax() |

### getUserFirstName

| | |
|---|---|
| Description | Return the User's First Name |
| Prototype | String User::getUserFirstName() |

### getUserLastName

| | |
|---|---|
| Description | Return the User's Last Name |
| Prototype | String User::getUserLastName() |

### getUserLocale

| | |
|---|---|
| Description | Returns the locale that is selected by the user for browsing content |
| Prototype | Locale getUserLocaleForContent() |

### getUserName

| | |
|---|---|
| Description | Return the User Name |
| Prototype | String User::getUserName() |

### getUserOrganizations

| | |
|---|---|
| Description | Return the User's Organizations |
| Prototype | Category[] User::getUserOrganizations() |

### getUserPhone

| | |
|---|---|
| Description | Return the User's Phone Number |
| Prototype | String User::getUserPhone() |

### getUserRoles

| | |
|---|---|
| Description | Return the User's Roles |
| Prototype | String[] User::getUserRoles() |

### getUserTitle

| | |
|---|---|
| Description | Return the User's Title |
| Prototype | String User::getUserTitle() |

### saveUser

| | |
|---|---|
| Description | Save the User's Profile. Returns null if the save was successful, otherwise returns an array of ValidationError's. |
| Prototype | ValidationError[] User::saveUser() |

### setUserAddress

| | |
|---|---|
| Description | Set the User's Address |

| | |
|---|---|
| Prototype | void User::setUserAddress(String str) |

### setUserEmail

| | |
|---|---|
| Description | Set the User's Email Address |
| Prototype | void User::setUserEmail(String str) |

### setUserFax

| | |
|---|---|
| Description | Set the User's Fax Number |
| Prototype | void User::setUserFax(String str) |

### setUserFirstName

| | |
|---|---|
| Description | Set the User's First Name |
| Prototype | void User::setUserFirstName(String str) |

### setUserLastName

| | |
|---|---|
| Description | Set the User's Last Name |
| Prototype | void User::setUserLastName(String str) |

### setUserPhone

| | |
|---|---|
| Description | Set the User's Phone Number |
| Prototype | void User::setUserPhone(String str) |

### setUserRoles

| | |
|---|---|
| Description | Sets the roles for an user |
| Prototype | Boolean User::setUserRoles(Role[] roles) |

### setUserTitle

| | |
|---|---|
| Description | Set the User's Title |
| Prototype | void User::setUserTitle(String str) |

### validateUser

| | |
|---|---|
| Description | Validates user based on Username, User Password, and User Company Code |
| Prototype | boolean validateUser(String sUserName, String sPassword, String sCmpCode) |

## AccessPrivilege

### createAccessControlGroup

| | |
|---|---|
| Description | Creates an access control group object with the specified ACG name and an optional ACG description |
| Prototype | ACG createAccessControlGroup(String sACGName, [String sACGDesc]) |

### getAccessControlGroupName

| | |
|---|---|
| Description | Return the name of the access control group |
| Prototype | String ACG::getAccessControlGroupName() |

### getAccessControlGroupByName

| | |
|---|---|
| Description | Returns a access control group object for the specified acg name |
| Prototype | ACG getAccessControlGroupByName(String sACGName) |

### getCtgAccessPrvByRole

| | |
|---|---|
| Description | Returns the catalog access privilege for the catalog and role. Returns catalog access privilege with full access if none was found. |
| Prototype | CtgAccessPrv Container::getCtgAccessPrvByRole(String sRoleName) |

### getCtgAccessPrvPermission

| | |
|---|---|
| Description | Returns the permission [E-editable\|V-viewable] for the node specified by the path in the current catalog access prv. |
| Prototype | String CtgAccessPrv::getCtgAccessPrvPermission(String attributeCollectionName) |

### new CtgAccessPrv

| | |
|---|---|
| Description | Builds a new catalog access privilege object |
| Prototype | new CtgAccessPrv(Container container, String roleName) |

### saveCtgAccessPrv

| | |
|---|---|
| Description | Saves the current catalog access prv to the database |
| Prototype | Boolean CtgAccessPrv::saveCtgAccessPrv() |

### setAccessPrv

| | |
|---|---|
| Description | Returns an access privilege object with the new permissions set for the attrGroupName. Permission is [V\|E\|null], and in case the permission is NULL the path is removed from the access Privilege. Returns TRUE if successful, FALSE if not |
| Prototype | Boolean CtgAccessPrv::setAccessPrv(String attrGroupName, String permission) |

### setCtgAccessPrv

| | |
|---|---|
| Description | Returns a catalog access privilege object with the permissions set according to the attribute collections. Permissions are [V\|E] |
| Prototype | CtgAccessPrv CtgAccessPrv::setCtgAccessPrv(String[] attrGroups, String[] permissions) |

### Sample Script for Creating Access Privilege

```
// sample 1: create catalog access prv
ctg = getCtgByName("Ctg1");
newCtgView = new CtgAccessPrv(ctg, "All Roles");

mypath = [];
mypath.add("Ctg1/Key");
mypath.add("Ctg1/Group/EAN/en_MY");
mypath.add("Ctg1/Group/EAN/zh_MY");
mypath.add("Ctg1/Group/EAN/ms_MY");
myper = [];
myper.add("E");
myper.add("V");
myper.add("E");
myper.add("V");

newCtgView = newCtgView.setCtgAccessPrv(mypath, myper);
newCtgView.saveCtgAccessPrv();
```

(to modify existing cap, just fetch a cap and set a new set of path/permissions. Setting empty path/permissions will delete the catalog access privilege)

## Spec

# Inheritance

### addAttributeGroup

| | |
|---|---|
| Description | Add the Attribute Groups to this inheritnace rule. |
| Prototype | void InheritanceRule::addAttributeGroup(String attributeGroupName) |

### deleteInheritanceRule

| | |
|---|---|
| Description | Delete the inheritance rule. |
| Prototype | void InheritanceRule::deleteInheritanceRule() |

### getInheritanceRuleByName

| | |
|---|---|
| Description | Returns the inheritance rule for the attribute. |
| Prototype | InheritanceRule getInheritanceRuleByName(String sRuleName) |

### getInheritanceTargets

| | |
|---|---|
| Description | Gets the inheritance target list. Targets are defined by an array of name and type. For example [my catalog name, CATALOG] |
| Prototype | String[][] InheritanceRule::getInheritanceTargets() |

### getMappedAttributeGroups

| | |
|---|---|
| Description | Returns an array of Strings representing the names of Attribute Collections mapped to this inheritance rule. |
| Prototype | String[] InheritanceRule::getMappedAttributeGroups() |

### new$InheritanceRule

| | |
|---|---|
| Description | Builds a new InheritanceRule object for the specified catalog and attribute |
| Prototype | new InheritanceRule(Container container, String ruleName) |

### reflattenAllInheritanceRules

| | |
|---|---|
| Description | Reflatten all the inheritance rules. WARNING operation might take time |

| | |
|---|---|
| Prototype | void reflattenAllInheritanceRules() |

### removeAttributeGroup

| | |
|---|---|
| Description | Removes the Attribute Groups from this inheritnace rule. Returns true of attribute group is removed, false if not. |
| Prototype | Boolean InheritanceRule::removeAttributeGroup(String attributeGroupName) |

### saveRule

| | |
|---|---|
| Description | Saves the inheritance rule (adding it if it is new). The rule must have at least one attribute colletions associated with it. It the rule is a catalog rule then it must have at least one target; if it's a hierarchy rule, then the rule shoudn't have any targets. |
| Prototype | void InheritanceRule::saveRule() |

### setInheritanceTargets

| | |
|---|---|
| Description | Sets the inheritance target list to the new list of containers. Container is defined by an array of name and type. For example ["my catalog name", "CATALOG" |
| Prototype | void InheritanceRule::setInheritanceTargets(String[][] containers) |

## IMutable Spec

### exportXML

| | |
|---|---|
| Description | Exports a WPC Spec to a String representing a XML file. |
| Prototype | String IMutableSpec::exportXML() |

### exportXSD

| | |
|---|---|
| Description | Exports a WPC Spec to a String representing the contents of XML Schema Definition. |
| Prototype | String IMutableSpec::exportXSD() |

### importXML

| | |
|---|---|
| Description | Imports a XML file to a WPC Spec. |
| Prototype | IMutableSpec importXML(String filename) |

### importXSD

| | |
|---|---|
| Description | Imports a XML Schema Definition file (.xsd) to a WPC Spec, using the given parameters. |
| Prototype | IMutableSpec importXSD(String filename,<br>String specName,<br>String specType,<br>String primaryKeyPath,<br>String maxAncestors,<br>String topLevelNamespace,<br>String topLevelName,<br>String archivedFilename) |

## Locale

### addToCompanyLocales

| | |
|---|---|
| Description | Adds the given locales to the list of locales that are defined for the company. |
| Prototype | void addToCompanyLocales(Locale []companyLocales) |

### getLocaleCode

| | |
|---|---|
| Description | Returns the 5 letter code (2 letter country code + underscore + 2 letter language code) for the given locale. |
| Prototype | String Locale::getLocaleCode() |

### getLocaleDisplayName

| | |
|---|---|
| Description | Returns a description of the locale suitable for display. |
| Prototype | String Locale::getLocaleDisplayName() |

### getLocalizedSpecNames

| | |
|---|---|
| Description | Returns all the specs that are localized. |
| Prototype | Spec[] getLocalizedSpecNames() |

### getLocales

| | |
|---|---|
| Description | return locales assiciated with the spec |
| Prototype | Object Spec:getLocales() |

### getNodeByPath

| | |
|---|---|
| Description | Returns the node object for path in this spec. |
| Prototype | Node Spec::getNodeByPath(String path) |

### getNodeLocale

| | |
|---|---|
| Description | Returns the locale object for this node if it is a locale specific node. |
| Prototype | Boolean Node::getNodeLocale() |

### new$Locale

| | |
|---|---|
| Description | Returns a locale with the country and language (two letter codes) combination specified and null if it is not supported |
| Prototype | new$Locale(String country_code, String language_code) |

### removeFromCompanyLocales

| | |
|---|---|
| Description | Removed the given locales from the list of locales that are defined for the company. |
| Prototype | void removeFromCompanyLocales(Locale []companyLocales) |

### replaceCompanyLocales

| | |
|---|---|
| Description | Sets the given locales for the company. Removes any existing locales. |
| Prototype | void replaceCompanyLocales(Locale []companyLocales) |

## SpecNode

### buildSpecNode

| | |
|---|---|
| Description | Returns a new node object of a spec with the given path and node order. Please make sure to use a spec that has been obtained using the new Spec() or buildSpec operation |
| Prototype | Node buildSpecNode(Spec spec, String path, Integer order) |

### buildSpecNodeName

| | |
|---|---|
| Description | Returns the parsed name that was passed in so that it can be used as a spec node name (spec node name only accept letters and characters, others are converted to an underscore _) |
| Prototype | String buildSpecNodeName(String name) |

### getNodeAttributeValue

| | |
|---|---|
| Description | Returns the value of this node's attribute, i.e. MAXLENGTH, MAX_OCCURRENCE, MIN_OCCURRENCE, HELP_URL, TYPE, etc. |
| Prototype | String Node::getNodeAttributeValue(String attributeName) |

### getNodeAttributeValues

| | |
|---|---|
| Description | Returns the values of this node's attribute, i.e. STRING_ENUMERATION. |
| Prototype | HashMap Node::getNodeAttributeValues(String attributeName) |

### getNodeChildren

| | |
|---|---|
| Description | Returns the children for the node. |
| Prototype | INode[] Node::getNodeChildren() |

### getNodeLookupTableName

| | |
|---|---|
| Description | Returns the name of the Lookup Table associated with this node, if one exists. |
| Prototype | String Node::getNodeLookupTableName() |

### getNodeName

| | |
|---|---|
| Description | Returns the name of this node. |

| | |
|---|---|
| Prototype | String Node::getNodeName() |

### getNodePath

| | |
|---|---|
| Description | Returns the path of this node. |
| Prototype | String Node::getNodePath() |

### getNodeSpec

| | |
|---|---|
| Description | Returns the spec object for this node. |
| Prototype | Spec Node::getNodeSpec() |

### getSpecNodes

| | |
|---|---|
| Description | Returns map of node paths to node objects for this spec. |
| Prototype | HashMap Spec::getSpecNodes() |

### isNodeEditable

| | |
|---|---|
| Description | Returns true if the node is editable, false otherwise |
| Prototype | Boolean Node::isNodeEditable() |

### isNodeGrouping

| | |
|---|---|
| Description | Returns true if the node is a grouping node, false otherwise |
| Prototype | Boolean Node::isNodeGrouping() |

### isNodeNonPersisted

| | |
|---|---|
| Description | Returns true if the node is a non-persisted node, false otherwise |
| Prototype | Boolean Node::isNodeNonPersisted() |

### isNodeSpecRoot

| | |
|---|---|
| Description | Returns true if the node is a spec root node, false otherwise |
| Prototype | Boolean Node::isNodeSpecRoot() |

## Sequence

### getSequenceByName

| | |
|---|---|
| Description | Gets the sequence object with the corresponding name where name is defined by the name of the catalog/category tree + "_" + "CTG" / "CATTREE" + "_" + the path of the node the sequence is defined for. |
| Prototype | Sequence getSequenceByName(String name) |

### getSequenceCurrentValue

| | |
|---|---|
| Description | Returns the current value of this sequence |
| Prototype | String Sequence::getSequenceCurrentValue() |

### getSequenceNextValue

| | |
|---|---|
| Description | Returns the next value of this sequence |
| Prototype | String Sequence::getSequenceNextValue() |

## Spec

### addToSpecLocales

| | |
|---|---|
| Description | Adds the given locales to the list of locales that are defined for the spec. |
| Prototype | void Spec::addToSpecLocales(Locale []newLocales) |

### addSubNode

| | |
|---|---|
| Description | Adds a SubNode from a SubSpec. |
| Prototype | Boolean Spec::addSubNode(Node node) |

### addSubSpec

| | |
|---|---|
| Description | Adds an entire SubSpec using a SubSpec. |
| Prototype | Boolean Spec::addSubSpec(Spec subSpec) |

### buildSpec

| | |
|---|---|
| Description | Returns a spec object given the name and the type of the spec |
| Prototype | Spec buildSpec(String specName, String specType) |

### buildTestSpec

| | |
|---|---|
| Description | Returns a new spec object with the specified name, type and number of fields in the spec |
| Prototype | Spec buildTestSpec(String name, String type, Integer fields) |

### deleteSpec

| | |
|---|---|
| Description | Delete this spec |
| Prototype | void Spec::deleteSpec() |

### getSpecAttribNames

| | |
|---|---|
| Description | returns the names of each attribute(node) specified in the spec |
| Prototype | String[] Spec::getSpecAttribNames() |

### getSpecAttribPaths

| | |
|---|---|
| Description | returns the names of each attribute(node) specified in the spec |
| Prototype | String[] Spec::getSpecAttribPaths() |

### getSpecByName

| | |
|---|---|
| Description | Returns the spec object with the corresponding name |
| Prototype | Spec getSpecByName(String name) |

### getSpecName

| | |
|---|---|
| Description | Returns the name of this spec |
| Prototype | String Spec::getSpecName() |

### getSpecMultiOccurAttributePaths

| | |
|---|---|
| Description | Returns the multi occurrence attribute paths for this spec |
| Prototype | HashMap Spec::getSpecMultiOccurAttributePaths() |

### getSpecPrimaryKeyAttributePath

| | |
|---|---|
| Description | Returns the primary key attribute path for this spec, null if it doesn't apply |

| | |
|---|---|
| Prototype | String Spec::getSpecPrimaryKeyAttributePath() |

### getSpecSequenceAttributePaths

| | |
|---|---|
| Description | Returns the sequence attribute paths for this spec. |
| Prototype | HashMap Spec::getSpecSequenceAttributePaths() |

### getSpecType

| | |
|---|---|
| Description | Returns the type of this spec |
| Prototype | String Spec::getSpecType() |

### getSpecUniqueAttributePaths

| | |
|---|---|
| Description | Returns the unique attribute paths for this spec. |
| Prototype | HashMap Spec::getSpecUniqueAttributePaths() |

### isLocalized

| | |
|---|---|
| Description | Returns a boolean if a spec is localized |
| Prototype | Boolean Spec::isLocalized() |

### new$Spec

| | |
|---|---|
| Description | Returns a new spec object with the given name and type |
| Prototype | new Spec(String specName, String specType) |

### new$SpecNode

| | |
|---|---|
| Description | Returns a new node created in the spec according to the path and order |
| Prototype | new SpecNode(Spec spec, String path, Integer order) |

### removeFromSpecLocales

| | |
|---|---|
| Description | Removes the given locales from the list of locales that are defined for the spec. |
| Prototype | void Spec::removeFromSpecLocales(Locale []newLocales) |

### replaceSpecLocales

| | |
|---|---|
| Description | Sets the given locales for the spec. Removes any existing locales. |

| | |
|---|---|
| Prototype | void Spec::replaceSpecLocales(Locale []newLocales) |

### saveSpec

| | |
|---|---|
| Description | Save this spec to the database |
| Prototype | void Spec::saveSpec() |

### saveSpecMap

| | |
|---|---|
| Description | Save this spec map to the database |
| Prototype | void Spec::saveSpecMap() |

### setAttribute

| | |
|---|---|
| Description | Set an attribute of a node or a spec. Please consult the documentation for allowable values of sAttributeName. Common values are MAX_OCCURRENCE, MIN_OCCURRENCE, TYPE, DEFAULT_VALUE. If the optional third parameter "dontReplace" is supplied, and is true, or we are dealing with a node rather than a spec, sValue is added to any existing values for this attribute rather than replacing them. |
| Prototype | void Node::setAttribute(String sAttributeName, String sValue), void Spec::setAttribute(String sAttributeName, String sValue) |

### setLocalized

| | |
|---|---|
| Description | Sets the localized property of a spec |
| Prototype | void Spec::setLocalized(Boolean localized) |

### setNodeEditable

| | |
|---|---|
| Description | Sets the node to be editable or non-editable |
| Prototype | void Node::setNodeEditable(Boolean) |

## SpecMap

### buildTestSpecMap

| | |
|---|---|
| Description | Returns a new spec map on the specified map type between the source and the destination specs - first delete existing map if there is one |

| | |
|---|---|
| Prototype | SpecMap buildTestSpecMap(String mapName, String mapType, Spec source, Spec destination) |

### getDefaultSpecMapName

| | |
|---|---|
| Description | See getSpecMapByName. Returns the name of the spec map being used for an aggregation/syndication. |
| Prototype | (deprecated) String getDefaultSpecMapName() |

### getSpecMapByName

| | |
|---|---|
| Description | Returns the specmap object with the corresponding name |
| Prototype | SpecMap getSpecMapByName([String name]) |

### getSpecMapDstObject

| | |
|---|---|
| Description | Returns the destination object of this spec map |
| Prototype | Object SpecMap::getSpecMapDstObject() |

### getSpecMapSrcObject

| | |
|---|---|
| Description | Returns the source object of this specmap |
| Prototype | Object SpecMap::getSpecMapSrcObject() |

### map

| | |
|---|---|
| Description | Add a mapping from sSrcPath to sDstPath to this spec map |
| Prototype | void SpecMap::map(String sSrcPath, String sDstPath) |

### new$SpecMap

| | |
|---|---|
| Description | Creates a new spec map of the given type between the source and destination objects |
| Prototype | new SpecMap(String mapType, Object source, Object destination) |

## SystemAdmin

# Logger

### debug

| | |
|---|---|
| Description | Write s to this logger |
| Prototype | String dumpContext([Logger l]) |

### dumpContext

| | |
|---|---|
| Description | Return the script context in a string (and dumps it to the logger l if specified) |
| Prototype | String dumpContext([Logger l]) ]] |

### dumpSystemLog

| | |
|---|---|
| Description | Return the last nLines of the system log sName |
| Prototype | String dumpSystemLog(String sName, int nbLines) |

## PerformanceTest

### beginPerf

| | |
|---|---|
| Description | Starts timing current block for perf. logging |
| Prototype | beginPerf(String name) |

### endPerf

| | |
|---|---|
| Description | Ends timing current block for perf. logging |
| Prototype | endPerf(String name) |

## SystemDB

### new$SystemDB

| | |
|---|---|
| Description | Returns an object that represents the status of the current database |
| Prototype | new SystemDB() |

### reportAllTableIndexes

| | |
|---|---|
| Description | Reports all the tables and their indexes |
| Prototype | String SystemDB::reportAllTableIndexes() |

### reportExtraIndexes

| | |
|---|---|
| Description | Reports the list of indexts that are extra in the current database that sould not be there |
| Prototype | String SystemDB::reportExtraIndexes() |

### reportIndexStatistics

| | |
|---|---|
| Description | Reports all the indexes and their current statistics and whether or not they need to be rebuilt |
| Prototype | String SystemDB::reportIndexStatistics() |

### reportMissingIndexes

| | |
|---|---|
| Description | Reports the list of indexts that are missing in the current database that sould be there |
| Prototype | String SystemDB::reportMissingIndexes() |

## Web Services

### createWebService

| | |
|---|---|
| Description | Creates a new web service with the given parameters. To save and deploy the service x(if DEPLOYED is true), call save(). NAME is the name of the service. DESC is the description of the service. WSDLDOCPATH is the doc path at which the WSDL is stored. PROTOCOL is the protocol. Currently SOAP_HTTP is the only supported protocol. IMPLSCRIPTPATH is the doc path of the service implementation script. It is the callers responsibility to ensure that WSDLDOCPATH and IMPLSCRIPTPATH do not cause the documents for any other web service to be overwritten. STOREINCOMING determines whether incoming requests are stored. STOREOUTGOING determines whether outgoing request are stored. DEPLOYED determines whether the service will be deployed. If a web service with the name of NAME already exists, throws an AustinException. |
| Prototype | WebService createWebService(String name, String desc, String wsdlDocPath, String protocol, String implScriptPath, Boolean storeIncoming, Boolean storeOutgoing, Boolean deployed) |

### deleteWebService

| | |
|---|---|
| Description | Deletes the web service in the DB and undeploys it. |
| Prototype | `void WebService::deleteWebService()` |

### getName implementation

| | |
|---|---|
| Description | Returns the name of this web service |
| Prototype | String WebService::getName() |

### getDesc

| | |
|---|---|
| Description | Returns the description of this web service |
| Prototype | String WebService::getDesc() |

### getUrl

| | |
|---|---|
| Description | Returns the URL for this web service |
| Prototype | String WebService::getUrl() |

### getWsdlUrl

| | |
|---|---|
| Description | Returns the WSDL URL for this web service |
| Prototype | String WebService::getWsdlUrl() |

### getWsdlDocPath

| | |
|---|---|
| Description | Returns the docstore path where the WSDL for this web service is stored. |
| Prototype | String WebService::getWsdlDocPath() |

### getProtocol

| | |
|---|---|
| Description | Returns the protocol for this web service. |
| Prototype | String WebService::getProtocol() |

### getImplScriptPath

| | |
|---|---|
| Description | Returns the docstore path where the implementation script for this web service is stored. |
| Prototype | String WebService::getImplScriptPath() |

### getStoreIncoming

| | |
|---|---|
| Description | Returns whether incoming messages for this web service are stored. |
| Prototype | Boolean WebService::getStoreIncoming() |

### getStoreOutgoing

| | |
|---|---|
| Description | Returns whether incoming messages for this web service are stored. |
| Prototype | Boolean WebService::getStoreOutgoing() |

### isDeployed

| | |
|---|---|
| Description | Returns whether this web service is deployed. |
| Prototype | Boolean WebService::isDeployed() |

### setName

| | |
|---|---|
| Description | Sets the name of the given WebService. |
| Prototype | `void WebService::setName(String name)` |

### setDesc

| | |
|---|---|
| Description | Sets the description of the given WebService. |
| Prototype | `void WebService::setDesc(String desc)` |

### setStoreIncoming

| | |
|---|---|
| Description | Sets the storeIncoming of the given WebService. |
| Prototype | `void WebService::setStoreIncoming(Boolean storeIncoming)` |

### setWsdlDocPath

| | |
|---|---|
| Description | Sets the docstore path of the WSDL document. The caller must ensure that this does not overwrite the WSDL for any other service. |
| Prototype | `void WebService::setWsdlDocPath(String wsdlDocPath)` |

### setImplScriptPath

| | |
|---|---|
| Description | Sets the docstore path of the implementation script for this webservice. The caller must ensure that this does not overwrite the implementation script for any other service. |
| Prototype | `void WebService::setImplScriptPath(String implScriptPath)` |

### setProtocol

| | |
|---|---|
| Description | Sets the protocol of the given WebService. |
| Prototype | `void WebService::setProtocol(String protocol)` |

### setStoreOutgoing

| | |
|---|---|
| Description | Sets whether this WebService should store outgoing messages. |
| Prototype | `void WebService::setStoreOutgoing(Boolean storeOutgoing)` |

### setDeployed

| | |
|---|---|
| Description | Sets whether this WebService is deployed. The setting will take effect upon saving. |
| Prototype | `void WebService::setDeployed(Boolean deployed)` |

### saveWebService

| | |
|---|---|
| Description | Saves the web service in the DB. If deployment settings have changed, they take effect upon saving. |
| Prototype | `void WebService::saveWebService()` |

## WorkEntryList

# WorkEntry

### getEntryFromWorkEntry

| | |
|---|---|
| Description | Get the Entry held by this WorkEntry |
| Prototype | Entry WorkEntry::getEntryFromWorkEntry() |

### getWorkEntryState

| | |
|---|---|
| Description | Get the current state of this WorkEntry |
| Prototype | String WorkEntry::getWorkEntryState() |

### isWorkEntryMarked

| | |
|---|---|
| Description | Is the current WorkEntry marked |
| Prototype | Boolean WorkEntry::isWorkEntryMarked() |

### isWorkEntryMarkedNew

| | |
|---|---|
| Description | Is the current WorkEntry marked new |
| Prototype | Boolean WorkEntry::isWorkEntryMarkedNew() |

### markWorkEntryDirty

| | |
|---|---|
| Description | Mark this WorkEntry as being dirty |
| Prototype | void WorkEntry::markWorkEntryDirty() |

### new WorkEntry

| | |
|---|---|
| Description | Creates a workentry for a given entry |
| Prototype | new WorkEntry(Entry entry, [Boolean markAsNew] |

### setWorkEntryMarked

| | |
|---|---|
| Description | Marks/unmarks this WorkEntry |
| Prototype | void WorkEntry::setWorkEntryMarked(Boolean mark) |

## WorkEntryList

### addWorkEntry

| | |
|---|---|
| Description | Insert a WorkEntry into the WorkEntryList at the specified index |
| Prototype | void WorkEntryList::addWorkEntry(int index, WorkEntry workEntry) |

### getIndexesOfEntriesHavingState

| | |
|---|---|
| Description | Get the current indexes of the worklist entries having a particular state |
| Prototype | Map WorkEntryList::getIndexesOfEntriesHavingState( String state) |

### getMarkedEntries

| | |
|---|---|
| Description | Return an entry set containing the marked entries in this work entry list - with indexes between start and end - |
| Prototype | EntrySet WorkEntryList::getMarkedEntries([start, |

| | |
|---|---|
| | end]) |

### getWorkEntryAt

| | |
|---|---|
| Description | Get the WorkEntry for the specified index in the WorkEntryList |
| Prototype | WorkEntry WorkEntryList::getWorkEntryAt(int i) |

### getWorkEntryListSize

| | |
|---|---|
| Description | Gets the size of this work entry list |
| Prototype | Integer WorkEntryList::getWorkEntryListSize() |

### new$WorkEntryList

| | |
|---|---|
| Description | Create a new work entry list from a catalog or a selection |
| Prototype | new WorkEntryList(ctgOrSelection, [sortingNodeId], [sortingOrder]) |

### removeWorkEntry

| | |
|---|---|
| Description | Removes the WorkEntry at the specified index from the WorkEntryList |
| Prototype | void WorkEntryList::removeWorkEntry(int index) |

### saveMarkedEntries

| | |
|---|---|
| Description | Save the set of marked entries for this work entry list - with indexes between start and end - for entries in the step specified by path in the collaboration area colArea with given comment. |
| Prototype | WorkEntryList::saveMarkedEntries(workList, [start, end, [colArea, path, comment] |

### syncWorkEntryAt

| | |
|---|---|
| Description | Sync the work entry at the specified index with it's database picture |
| Prototype | void WorkEntryList::syncWorkEntryAt(int i) |

## Workflow

# ColAreaEntryHistory

The following is a list of Collaboration Area Entry History/Reporting Operations.

### getColAreaEntryHistory

| | |
|---|---|
| Description | Return the entire history of the entry in the given collaboration area. |
| Prototype | ColAreaEntryHistory[] getColAreaEntryHistory(String colAreaName, String wflName, String primaryKey) |

### getColAreaHistoryByTimePeriod

| | |
|---|---|
| Description | Return the entire history given collaboration area for the given time period. |
| Prototype | ColAreaEntryHistory[] getColAreaHistoryByTimePeriod(String colAreaName, Date beginDate, Date endDate) |

### getColAreaHistoryDate

| | |
|---|---|
| Description | Returns the date for the given collaboration area history event. |
| Prototype | Date ColAreaEntryHistory::getColAreaHistoryDate() |

### getColAreaHistoryEntryKey

| | |
|---|---|
| Description | Returns the entry key for the given collaboration area history event. |
| Prototype | String ColAreaEntryHistory::getColAreaHistoryEntryKey() |

### getColAreaHistoryEventAttribute

| | |
|---|---|
| Description | Returns the attribute value for the given collaboration area history event type attribute name. attrName could be one of the following: COMMENT, EXIT_VALUE, ENTRY_DIFFERENCES |
| Prototype | String ColAreaEntryHistory::getColAreaHistoryEventAttribute(String attrName) |

### getColAreaHistoryEventType

| | |
|---|---|
| Description | Returns the event type for the given collaboration area history event. Event types could be one of the following: CHECKOUT, CHECKIN, ENTERSTEP, LEAVESTEP, SAVEENTRY, DROP, TIMEOUT. |
| Prototype | String ColAreaEntryHistory::getColAreaHistoryEventType() |

### getColAreaHistoryStepPath

| | |
|---|---|
| Description | Returns the step path for the given collaboration area history event. |
| Prototype | String ColAreaEntryHistory::getColAreaHistoryStepPath() |

### getColAreaHistoryUser

| | |
|---|---|
| Description | Returns the username for the given collaboration area history event. |
| Prototype | String ColAreaEntryHistory::getColAreaHistoryUser() |

### getColAreaStepHistory

| | |
|---|---|
| Description | Return the entire history of the step in the given collaboration area. |
| Prototype | ColAreaEntryHistory[] getColAreaStepHistory(String colAreaName, String wflName, String stepPath) |

## CollaborationArea

### addEntryIntoColArea

| | |
|---|---|
| Description | Posts a message to add tthe entry in the given stepPath of the collaboration area. Returns a boolean depending on whether the entry was successfully added or not. You cannot assume that the entry is in the collaboration area when this method returns. |
| Prototype | boolean CollaborationArea::addEntryIntoColArea(Entry |

| | |
|---|---|
| | CollaborationArea::addEntryIntoColArea(Entry entry, String stepPath ) |

### checkOutEntries

| | |
|---|---|
| Description | Checks-out the entries in the entrySet into the collaboration area. If stepPath is not specified the entries will be checked-out into the Initial step. The event id is returned. If waitForStatus is false, always return null. If waitForStatus is true, then this operation returns when the separate workflow engine has processed the event. Default is false. Returns a HashMap of entry primary key to the status of the checkout. This method blocks until. Checkout status could be one of the following: CHECKOUT_SUCCESSFUL and ATTRIBUTE_LOCKED. If any attribute is locked in some other collaboration area, then the status of ATTRIBUTE_LOCKED is returned for that entry primary key. |
| Prototype | HashMap CollaborationArea::checkOutEntries(EntrySet entrySet, [String stepPath], [boolean waitForStatus) |

### dropEntries

| | |
|---|---|
| Description | Posts an event to drops the entries in the entrySet from the collaboration area and to unlock the attributes which were locked in the source catalog for the entry. You cannot assume that this operation has completed when this method returns. |
| Prototype | void CollaborationArea::dropEntries(EntrySet entrySet) |

### getColAreaName

| | |
|---|---|
| Description | Returns the name of the collaboration area. |
| Prototype | String CollaborationArea::getColAreaName() |

### getColAreaNames

| | |
|---|---|
| Description | Returns all of the Collaboration Area Names for the current Company |
| Prototype | String[] getColAreaNames() |

### getColAreaContainer

| | |
|---|---|
| Description | Returns the collaboration area as a container. |
| Prototype | Container CollaborationArea::getColAreaContainer() |

### getColAreaSrcContainer

| | |
|---|---|
| Description | Returns the source container which this collaboration area is tied to. |
| Prototype | Container CollaborationArea::getColAreaSrcContainer() |

### getColAreaWorkflow

| | |
|---|---|
| Description | Returns the workflow that this collaboration area is tied to. |
| Prototype | Workflow CollaborationArea::getColAreaWorkflow() |

### getCountOfEntriesInColArea

| | |
|---|---|
| Description | Returns the entries currently in ALL the steps of the collaboration area. |
| Prototype | int CollaborationArea::getCountofEntriesInColArea( ) |

### getCountOfEntriesInColAreaStep

| | |
|---|---|
| Description | Returns the entries currently in the given stepPath of the collaboration area. |
| Prototype | int CollaborationArea::getCountofEntriesInColAreaStep(String stepPath) |

### getEntries

| | |
|---|---|
| Description | Returns the entry set for the entries currently in the collaboration area. |
| Prototype | EntrySet CollaborationArea::getEntries() |

### getEntriesInStep

| | |
|---|---|
| Description | Returns the entry set for the entries currently in the step of the collaboration area. The format of the stepPath is Stepname |

| | |
|---|---|
| Prototype | EntrySet CollaborationArea::getEntriesInStep(String stepPath) |

### getReservedEntriesInStep

| | |
|---|---|
| Description | Returns the entry set for the reserved entries currently in the step of the collaboration area. The format of the stepPath is Stepname |
| Prototype | EntrySet CollaborationArea::getReservedEntriesInStep(String stepPath) |

### getUsernameForReservedEntryInStep

| | |
|---|---|
| Description | Returns the username of the user who locked the entry in a wfl step for a given collaboration area, otherwise it returns null. |
| Prototype | String CollaborationArea::getUsernameForReservedEntryInStep(Entry entry, String stepPath) |

### isEntryReservedInStep

| | |
|---|---|
| Description | Returns true if the entry is locked in a wfl step for a given collaboration area, otherwise it returns false. |
| Prototype | Boolean CollaborationArea::isEntryReservedInStep(IEntry entry, String stepPath) |

### lockColArea

| | |
|---|---|
| Description | Locks the Collaboration Area so that no more entries can be checked out into it. Returns true or false depending on whether the lock was successfully applied or not. |
| Prototype | Boolean CollaborationArea::lockColArea() |

### moveEntriesToColArea

| | |
|---|---|
| Description | Moves the entrySet of entries in the collaboration area to another collaboration area. For now, use only within the IN() script of a workflow step. destColAreaName specifies the name of the destination collaboration area, into whose Initial step the entries will be checked into. Returns a |

| | |
|---|---|
| | boolean depending on whether the entrySet was successfully moved or not. |
| Prototype | boolean CollaborationArea::moveEntriesToColArea(EntrySet entrySet, String destColAreaName) |

### moveEntriesToNextStep

| | |
|---|---|
| Description | Moves the entries in the entrySet from the step to the next step depending on the exitValue. You cannot assume that this operation has completed when this method returns. |
| Prototype | void CollaborationArea::moveEntriesToNextStep(EntrySet entrySet, String stepPath, String exitValue) |

### new$CollaborationArea

| | |
|---|---|
| Description | Create a new collaboration area with the given name, wfl and srcContainer |
| Prototype | new CollaborationArea(String colAreaName, Workflow wfl, Container srcContainer) |

### releaseEntryInStep

| | |
|---|---|
| Description | Returns true if the entry was unlocked in a wfl step for a given collaboration area, otherwise it returns false. Operation runs synchronously. |
| Prototype | Boolean CollaborationArea::releaseEntryInStep(Entry entry, String stepPath) |

### reserveEntryInStep

| | |
|---|---|
| Description | Returns true if the entry was reserved in a wfl step for a given collaboration area, otherwise it returns false. Operation runs synchronously. |
| Prototype | Boolean CollaborationArea::reserveEntryInStepForUser(IEntry entry, String stepPath, [String username]) |

### saveColArea

| | |
|---|---|
| Description | Saves the collaboration area. |
| Prototype | void CollaborationArea::saveColArea() |

### setColAreaAdminRoles

| | |
|---|---|
| Description | Sets the admin roles for the collaboration area. |
| Prototype | void CollaborationArea::setColAreaAdminRoles(String[] roles) |

### setColAreaAdminRoles

| | |
|---|---|
| Description | Sets the admin roles for the collaboration area. |
| Prototype | void CollaborationArea::setColAreaAdminRoles(String[] roles) |

### setColAreaAdminUsers

| | |
|---|---|
| Description | Sets the admin users for the collaboration area. |
| Prototype | void CollaborationArea::setColAreaAdminUsers(String[] users) |

### setStepEntryTimeout

| | |
|---|---|
| Description | Expects the entry to actually be in the given collaboration area's specified stepPath.<br>Provided the entry is found to actually be in the step, its timeout is set to be the moment in time specified by the date argument.<br>If any of the assumptions are not met (collaboration area has no such stepPath, entry are not in that stepPath, etc.), the operation simply does nothing, i.e. no Exception thrown.<br>The operation doesn't modify the collaboration area's underlying workflow at all. It should be thought of as operating on an entry in a collaboration area, that is expected to be in a particular stepPath at the moment in time when the op is executed. |
| Prototype | void CollaborationArea::setStepEntryTimeout(Entry entry, String stepPath, Date date) |

### unlockColArea

| | |
|---|---|
| Description | Unlocks the Collaboration Area so that entries can be checked out into it again. Returns true or |

| | |
|---|---|
| | false depending on whether the unlock was successful or not. |
| Prototype | Boolean CollaborationArea::unlockColArea() |

# Workflow

### createNestedWflStep

| | |
|---|---|
| Description | Adds a nested workflow step to the workflow. Returns the WorkflowStep object. |
| Prototype | WorkflowStep Workflow::createNestedWflStep (Workflow nestedWfl) |

### createWflStep

| | |
|---|---|
| Description | Adds a new step to the workflow if the step with the given name does not exists. StepType can be one of the following: AND_APPROVAL, OR_APPROVAL, MODIFY, DISPATCH, MERGE, GENERAL, AUTOMATED, PARTIAL_UNDO, CONDENSER. Returns the WorkflowStep object. |
| Prototype | WorkflowStep Workflow::createNestedWflStep (Workflow nestedWfl) |

### deleteWfl

| | |
|---|---|
| Description | Delete a workflow. It throws an exception if the workflow can not be deleted (if used by any collaboration area) |
| Prototype | void Workflow::deleteWfl() |

### getAllWflNames

| | |
|---|---|
| Description | Returns a list of all workflow names. |
| Prototype | String[] getAllWflNames() |

### getWflAccessControlGroup

| | |
|---|---|
| Description | Returns access control group name of the workflow. |
| Prototype | String Workflow::getWflAccessControlGroup() |

### getWflByName

| | |
|---|---|
| Description | Returns the workflow if found otherwise null. |
| Prototype | Workflow getWflByName(String wflName) |

### getWflFailureStep

| | |
|---|---|
| Description | Returns the failure step of the workflow. |
| Prototype | WorkflowStep Workflow::getWflFailureStep() |

### getWflInitialStep

| | |
|---|---|
| Description | Returns the initial step of the workflow. |
| Prototype | WorkflowStep Workflow::getWflInitialStep() |

### getWflName

| | |
|---|---|
| Description | Returns the workflow name. |
| Prototype | String Workflow::getWflName() |

### getWflStepByName

| | |
|---|---|
| Description | Returns the step of the workflow otherwise null. |
| Prototype | WorkflowStep Workflow::getWflStepByName(String stepName) |

### getWflStepPaths

| | |
|---|---|
| Description | Returns the paths for all the steps of the workflow. |
| Prototype | String[] Workflow::getWflStepPaths() |

### getWflSteps

| | |
|---|---|
| Description | Returns the list of all the steps in the workflow. |
| Prototype | WorkflowStep[] Workflow::getWflSteps() |

### getWflSuccessStep

| | |
|---|---|
| Description | Returns the success step of the workflow. |
| Prototype | WorkflowStep Workflow::getWflSuccessStep() |

### new$Workflow

| | |
|---|---|
| Description | Create a new workflow of the given container type and with the given name. Container type can be one of the following: CATALOG, |

| | |
|---|---|
| | CATEGORY_TREE |
| Prototype | new Workflow(String wflName, String containerType) |

### saveWfl

| | |
|---|---|
| Description | Saves the workflow. Returns true or false depending on whether the workflow was successfully saved or not. |
| Prototype | Boolean Workflow::saveWfl() |

### setWflAccessControlGroup

| | |
|---|---|
| Description | Sets access control group name of the workflow. |
| Prototype | void Workflow::setWflAccessControlGroup(String acg) |

### setWflDesc

| | |
|---|---|
| Description | Sets the workflow description |
| Prototype | Workflow::setWflDesc(String wflDesc) |

### setWflName

| | |
|---|---|
| Description | Sets the workflow name |
| Prototype | Workflow::setWflName(String wflName) |

## Workflow Step

### getNextWflStepsForExitValue

| | |
|---|---|
| Description | Returns the names of the next steps for a particular exitValue of a WorkflowStep. |
| Prototype | String[] WorkflowStep::getNextWflStepsForExitValue(String exitValue) |

### getWflStepAddEntries

| | |
|---|---|
| Description | Returns value of 'allow import into step' flag. |
| Prototype | Boolean WorkflowStep::getWflStepAddEntries() |

### getWflStepCategorizeEntries

| | |
|---|---|
| Description | Returns value of 'allow recategorization' flag. |
| Prototype | Boolean WorkflowStep::getWflStepCategorizeEntries() |

### getWflStepDefaultScriptPath

| | |
|---|---|
| Description | Gets the default path of the workflow script for the step: scripts/workflow/<workflow name>/<step name>. |
| Prototype | String WorkflowStep::getWflStepDefaultScriptPath() |

### getWflStepEntryNotification

| | |
|---|---|
| Description | Gets the notification emails that will get sent when the item gets into the step. |
| Prototype | String WorkflowStep::getWflStepEntryNotification() |

### getWflStepExitValues

| | |
|---|---|
| Description | Retrieve the exit values of the WorkflowStep. |
| Prototype | String[] WorkflowStep::getWflStepExitValues() |

### getWflStepName

| | |
|---|---|
| Description | Returns the workflow step name. |
| Prototype | String WorkflowStep::getWflStepName() |

### getWflStepPerformerRoles

| | |
|---|---|
| Description | Returns the list of user roles for the workflow step. |
| Prototype | String[] WorkflowStep::getWflStepPerformerRoles() |

### getWflStepPerformerUsers

| | |
|---|---|
| Description | Returns the list of user names for the workflow step. |
| Prototype | String[] WorkflowStep::getWflStepPerformerUsers() |

### getWflStepScriptPath

| | |
|---|---|
| Description | Gets the path of the workflow script for the step. If no script is defined, returns null. |
| Prototype | String WorkflowStep::getWflStepScriptPath() |

### getWflStepTimeoutDuration

| | |
|---|---|
| Description | Gets the timeout duration for the workflow step. Returns a string in milliseconds. |
| Prototype | String WorkflowStep::getWflStepTimeoutDuration() |

### getWflStepTimeoutNotification

| | |
|---|---|
| Description | Gets the notification emails, which will get sent when the step times out. |
| Prototype | String WorkflowStep::getWflStepTimeoutNotification() |

### getWflStepType

| | |
|---|---|
| Description | Returns the workflow step type. |
| Prototype | String WorkflowStep::getWflStepType() |

### getValidationErrorEntryNode

| | |
|---|---|
| Description | Return the EntryNode associated with this ValidationError. |
| Prototype | EntryNode ValidationError::getValidationErrorEntryNode() |

### getValidationErrorMsg

| | |
|---|---|
| Description | Return the error message associated with this ValidationError |
| Prototype | String ValidationError::getValidationErrorMsg() |

### getWflStepReserveToEdit

| | |
|---|---|
| Description | Returns the reserve for edit flag for a workflow step. |
| Prototype | Boolean WorkflowStep::getWflStepReserveToEdit() |

### mapWflStepExitValueToNextStep

| | |
|---|---|
| Description | Maps the exit value of the WorkflowStep to the nextStep. The nextStep can either be the stepName or one WorkflowStep or an array of StepNames or an array of WorkflowSteps. |
| Prototype | void WorkflowStep::mapWflStepExitValueToNextStep(String exitValue, String \| WorkflowStep \| WorkflowStep[] nextStep) |

### setWflStepReserveToEdit

| | |
|---|---|
| Description | Sets the reserve for edit flag for a workflow step. |
| Prototype | void WorkflowStep::setWflStepReserveToEdit(Boolean flag) |

### setWflStepAddEntries

| | |
|---|---|
| Description | Sets value of 'allow import into step' flag. |
| Prototype | void WorkflowStep::setWflStepAddEntries(Boolean flag) |

### setWflStepAttributeGroups

| | |
|---|---|
| Description | Sets the attrinute groups for the workflow step. |
| Prototype | void WorkflowStep::setWflStepAttributeGroups(String[]/AttrGroup[] attrGroups) |

### setWflStepCategorizeEntries

| | |
|---|---|
| Description | Sets value of 'allow recategorization' flag. |
| Prototype | void WorkflowStep::setWflStepCategorizeEntries(Boolean flag) |

### setWflStepDesc

| | |
|---|---|
| Description | Sets the desc for the workflow step. |
| Prototype | void WorkflowStep::setWflStepDesc(String stepDesc) |

### setWflStepEntryNotification

| | |
|---|---|
| Description | Sets up the notification emails which will get sent when the item gets into the step. Email addresses must be seperated by semi-colons. |
| Prototype | void WorkflowStep::setWflStepEntryNotification(String emailAdresses) |

### setWflStepExitValues

| | |
|---|---|
| Description | Sets the exit values for the workflow step. |
| Prototype | void WorkflowStep::setWflStepExitValues(String[] exitValues) |

### setWflStepPerformerRoles

| | |
|---|---|
| Description | Sets the user roles for the workflow step. |
| Prototype | void WorkflowStep::setWflStepPerformerRoles(String[] roles) |

### setWflStepPerformerUsers

| | |
|---|---|
| Description | Sets the users for the workflow step. |
| Prototype | void WorkflowStep::setWflStepPerformerUsers(String[] users) |

### setWflStepScriptPath

| | |
|---|---|
| Description | Sets up the workflow script path for this step. If no argument is passed, the default location is used (script/<workflow name>/<step name>). Note that this operation does not check that the script is already loaded (it allows you to load the script later if needed). |
| Prototype | void WorkflowStep::setWflStepScriptPath([String scriptPath]) |

### setWflStepTimeoutDate

| | |
|---|---|
| Description | Sets up the timeout date for the workflow step. |
| Prototype | void WorkflowStep::setWflStepTimeoutDate(Date |

| | |
|---|---|
| | date) |

### setWflStepTimeoutDuration

| | |
|---|---|
| Description | Sets up the timeout duration for the workflow step. The duration must be in seconds. |
| Prototype | void WorkflowStep::setWflStepTimeoutDuration(int seconds) |

### setWflStepTimeoutNotification

| | |
|---|---|
| Description | Sets up the notification emails which will get sent when the step times out. Email addresses must be seperated by semi-colons. |
| Prototype | void WorkflowStep::setWflStepTimeoutNotification(String emailAdresses) |

## Widget

### addListener

| | |
|---|---|
| Description | Hook the onchange function handlerFunctionName to changes of widgetObserved - return false iff the operation fails |
| Prototype | boolean addListener(Widget widgetObserved, String handlerFunctionName) |

### addListenerForProperty

| | |
|---|---|
| Description | Hook the onchange function handlerFunctionName to changes of widgetObserved's sProperty - return false iff the operation fails |
| Prototype | boolean addListenerForProperty(Widget widgetObserved, String sProperty, String handlerFunctionName) |

### buildWidget

| | |
|---|---|
| Description | Creates a widget of type sType and name sName |
| Prototype | Widget buildWidget(String sType, String sName) |

### getWidget

| | |
|---|---|
| Description | Returns the relative widget sRelativePath |
| Prototype | Widget Widget::getWidget(String sRelativePath) |

### getWidgetProperty

| | |
|---|---|
| Description | Return the property sPropertyName of this widget |
| Prototype | Object Widget::getWidgetProperty(String sPropertyName) |

### initWidgetWithArgs

| | |
|---|---|
| Description | Call initWidgetWithArgs on the widget |
| Prototype | void Widget::initWidgetWithArgs(alArgs) |

### invalidate

| | |
|---|---|
| Description | Invalidate the widget - so that it gets re-rendered |
| Prototype | void Widget::invalidate() |

### isFullScreen

| | |
|---|---|
| Description | Return true the left navigation bar is hidden |
| Prototype | Boolean isFullScreen() |

### renderHorizontalBars

| | |
|---|---|
| Description | Return an HTML table to display horizontal bars - anHeights[i] should have the length of the i-th bar and asLabels[i] the tooltip for the i-th bar |
| Prototype | String renderHorizontalBars(Integer barWidth, Integer barHeight, Integer[] anLengths, String[] asLabels) |

### renderVerticalBars

| | |
|---|---|
| Description | Return an HTML table to display vertical bars - anHeights[i] should have the length of the i-th column and asLabels[i] the tooltip for the i-th column |
| Prototype | String renderVerticalBars(Integer barWidth, Integer barHeight, Integer[] anLengths, String[] asLabels) |

### renderWidget

| | |
|---|---|
| Description | Render the widgetw |
| Prototype | Widget::renderWidget(Writer out) |

### setWidgetProperty

| | |
|---|---|
| Description | Set the property sPropertyName of this widget to the value oValue |
| Prototype | void Widget::setWidgetProperty(String sPropertyName, Object oValue) |

# Updates to script operations

This document provides a list of script operations that have been modified in WebSphere Product Center 5.1 from version 5.0.

**Note**: The script operations listed in this document are for reference only and are subject to change. Refer to the Script Sandbox in the WebSphere Product Center product for more accurate information.

Each modified script operation listed in the following sections is preceded with one of the acronyms listed in the table below, which represent the type of change that was made.

| | |
|---|---|
| N | New |
| D | Deprecated |
| PU | Prototype updated |
| DU | Description updated |

### Catalog

N setOrdered

- ***Prototype***: Boolean Catalog::setOrdered(Boolean bOrder)
- ***Description***: Alters the catalog to allow ordering or not

### Category

N reorderEntry

- ***Prototype:*** Integer Category::reorderEntry(Catalog ctg, Entry child, Integer position, Boolean bInsertBefore)
- ***Description***: Allows users to adjust the ordering of a child Entry within a parent category in catalog ctg. Entry child is moved before (bInsertBefore=true) or after (bInsertBefore=false) the position (zero is the first element) specified. Returns the ordered entry id (if it works) or null (if it fails). This method should not be used in conjunction with a transaction. The Boolean flag is optional and if not specified defaults to true.

N getEntryPosition

- ***Prototype***: Integer Category::getEntryPosition(Catalog ctg, Entry child)
- ***Description***: Allows users to get the position of a child Entry within a parent category. This will only work in an ordered catalog. Returns the position (if it works) or null (if it

fails).

### Container

DU setContainerProperties(HashMap properties)

- ***Description:*** The properties specified in the PROPERTIES hashmap are set for the container in question. The hashmap keys can be one of "SCRIPT_NAME" "PRE_SCRIPT_NAME" "POST_SAVE_SCRIPT_NAME" "ENTRY_BUILD_SCRIPT" "DISPLAY_ATTRIBUTE" "USER_DEFINED_CORE_ATTRIBUTE_GROUP" "SCRIPT_RESTRICT_LOCALES". The values are required to be string names for scripts, Node object for "DISPLAY_ATTRIBUTE", an AttrGroup object for "USER_DEFINED_CORE_ATTRIBUTE_GROUP" and "true" or "false" for "SCRIPT_RESTRICT_LOCALES". If "SCRIPT_RESTRICT_LOCALES" is set to "false" (case insensitive) then script operations on entries in this container will not take account of the locale restrictions defined in User Settings

### String

N getSystemMessageById

- ***Prototype***: String getSystemMessageById(int id, [Locale loc])
- ***Description***: Given message id (and locale), returns description of the message.

N getSystemMessageByName

- ***Prototype***: String getSystemMessageByName(String msg_name, [Locale loc])
- ***Description***: Given message name (and locale), returns description of the message.

N getCustomMessage

- ***Prototype***: String getCustomMessage(String id, [Locale loc])
- ***Description***: Given message id (and locale), returns description of the message.

N bidiTransform

- ***Prototype***: public String bidiTransform(String srcStr, String direction, String typeOfText, String orientation, String swap, String numShapes, String textShapes)
- ***Description*:** If direction is "IMPORT", using the BiDi attributes specified in the parameters to create a BiDiText and then tranform it to BiDiText with default attributes.\
  If direction is "EXPORT", create a BiDiText using default attribute then tranform it to BiDiText with attributes specified in the parameters.\
  typeOfText can be : "IMPLICIT", "VISUAL". \
  orientation can be : "LTR", "RTL", "CONTEXTUAL_LTR", "CONTEXTUAL_RTL".\
  swap can be : "YES", "NO".\
  numShapes can be : "NOMINAL", "NATIONAL", "CONTEXTUAL", "ANY".\
  textShapes can be : "NOMINAL", "SHAPED", "INITIAL", "MIDDLE", "FINAL", "ISOLATED".\
  default value is: typeOfText:"IMPLICIT" orientation:"LTR" swap:"YES" numShapes:NOMINAL textShapes:NOMINAL\

### Currency

N getCurrencySymbolByCode

- ***Prototype***: String getCurrencySymbolByCode(String code)
- ***Description*** This operation return currency symbol from currency code,such as input "USD",currency symbol return will be "$".

N getCurrencyDescByCode

- ***Prototype***: String getCurrencyDescByCode(String code)
- ***Description:*** This operation return currency description from currency code

N getCompanyCurrencies

- ***Prototype***: String[] getCompanyCurrencies()
- ***Description:*** This operation return currencies code selected in company attribute.

N setCompanyCurrencies

- ***Prototype***: void setCompanyCurrencies(String listOfCodes[])
- ***Description:*** This operation set the list of codes to the company datebase.

N getAllCurrencies

- ***Prototype***: String[] getAllCurrencies()
- ***Description:*** This operation return all currency codes that WPC supports.

### Date

N getDateTimeInUserTimeZone

- ***Prototype***: Date getDateTimeInUserTimeZone()
- ***Description:*** Returns the number of seconds since January 1, 1970, 00:00:00 GMT represented by this Date object

N getDateInputFormat

- ***Prototype***: String getDateInputFormat()
- ***Description:*** Returns the date input format set in my setting

N getDateOutputFormat

- ***Prototype***: String getDateOutputFormat()
- ***Description:*** Returns the date output format set in my setting

N setDateInputFormat

- ***Prototype***: void setDateInputFormat(String format)

- ***Description:*** Set the Date input format

N setDateOutputFormat

- ***Prototype***: void setDateOutputFormat(String format)
- ***Description:*** Set the Date output format

### Number

N formatNumberByPrecision

- ***Prototype***: String formatNumberByPrecision(Double number,Integer precision)
- ***Description:*** This operation returns a string format along with defined precision

N formatNumberByLocPrecision

- ***Prototype***: String formatNumberByLocPrecision(Double number, Locale loc, Integer precision)
- ***Description:*** This operation returns a string format along with defined precision and locale

N parseDouble

- ***Prototype***: Double parseDouble(String str, Locale loc)
- ***Description:*** Pass string to double value based on locale

N parseNumber

- ***Prototype***: String parseNumber(String str, String numberFormat, Locale locale)
- ***Description:*** Use to parse a String to Number by numberFormat and locale. If locale is null, it will use the locale of user setting .If numberFormat is null, it will use the default format of the locale. The numberFormat string is a pattern whose format is identical to the number format used by Java

N formatNumber

- ***Prototype***: String Number::formatNumber(String numberFormat, Locale loc)
- ***Description:*** Use to format an Number to a human readable format according to the locale specified in the parameter.If locale is null,it will use the locale of user setting. If numberFormat is null, it will use the default format of the locale.

### Timezone

N getUserTimeZoneDesc

- ***Prototype***: String getUserTimeZoneDesc()
- ***Description:*** Get the user setting time zone's description in native language.

N getUserTimeZoneOffset

- ***Prototype***: Number getUserTimeZoneOffset()

 

- ***Description:*** Get user setting time zone's offset from GMT in minutes.

N setUserTimeZone

- ***Prototype***: void setUserTimeZone(int offset)
- ***Description:*** Change user setting's time zone with the offset value in minutes.

N getTimeZoneDesc

- ***Prototype***: String getTimeZoneDesc(int offsetInMinutes,Locale locale)
- ***Description:*** Get the time zone's description with the offset value in minutes.

N parseTimeZoneToDBValue

- ***Prototype***: String parseTimeZoneToDBValue(String srcStr)
- ***Description:*** Parse the string to time zone then return the db value.

N getTimeZoneOffsetFromDBValue

- ***Prototype***: Number getTimeZoneOffsetFromDBValue(String dbValue)
- ***Description:*** Get time zone from the db value and return the offset from GMT in minutes.

# Script Types

## Report Scripts

- Report scripts are used to create custom reports
- When creating a report in WPC, a script is required to define the report output
- The report script is used to define how the information is ordered and formatted

## Validation Rule Script Expressions

Definition:

- A validation rule, like a value rule, is used within a specification
- A validation rule is used to validate an attribute based value on a business rule
- A validation rule must return a value of true or false

Other details:

- Implicit object: item object (item)
- Implicit variable: to set the attribute to the resulting value, use "res = "
- The value for "res" needs to be either TRUE or FALSE

## Distribution Scripts

Definition:

- Distributions scripts are used to create a custom distribution that is not addressed by the built-in WPC distributions
- Examples: Ariba Catalog Upload, FTP, HTTP POST, email

## Import and Export Scripts

Definition:

- Import and Export scripts are used to import data into and export data out of WPC.

Other details:

- Use the operation "logWarning()" to report on lines that have something that you want to flag but still want to load in the catalog

- Use the operation "logError()" to report on lines that have something that you want to flag and when you want to prevent the line to be loaded in the catalog
- To make use of the "Item.saveCtgItem()" operation to save an item in a catalog other than the one specified in the import feed itself, you will need to put the following line at the top of the import script:
  `setScriptContextValue("$action", null);`

## Entry Macro Scripts

Definition:

- The Entry Macro script allows a user to execute a script within the data entry screens
- For example, a script could be written to replace all strings with a given value

Other details:

- Implicit object: item object (item)
- In the context of a macro script, the script runs for each item selected in the data entry screen
- Macro scripts distinguish themselves from Mass Update scripts in that they do not save the items automatically - it is recommended that they do not
- Changes are visible on the screen and users can chose to either save or discard them

## Catalog Difference Export Scripts

Definition:

- The Catalog Differences Export script allows one to perform a comparison of two catalog versions
- For each item, the status between the two versions can be accessed
- Four possible types of status: Modified, Added, Deleted, Unmodified

Other details:

- Implicit objects: file object (out), catalog object
- While the catalog object is implicit, there is not actually any implicit variable name through which to reference the catalog. To obtain a variable representing the catalog, simply execute a "getCtgByName()" passing in no arguments.

- In the context of an export, the operation "forEachCtgItem()" doesn't require the catalog name, it will by default get the Catalog Name selected when creating the syndication
- The difference status can be: M (modified), A (added), D (deleted), U (unchanged)
- The operation "getCtgItemAtOldVersion()" allows you to get the item object in the old version; you can then use it to compare it with its more recent version

# Catalog Import Scripts

Definition:

- Catalog Import scripts are used during aggregation and can be used to perform advanced operations on incoming data before it is imported into a catalog
- For every import feed, WPC generates a simple script by default, based on the file spec to catalog (or catalog spec) mapping
- Instead of using the default generated script, users can create and use new scripts as needed to perform advanced operations

Other details:

- Implicit objects: file object (in), catalog object
- While the catalog object is implicit, there is not actually any implicit variable name through which to reference the catalog. To obtain a variable representing the catalog, simply execute a "getCtgByName()" passing in no arguments.
- The operation "new CtgItem()" in the context of an item feed or an item to category map feed doesn't require the catalog name; it will by default get the Catalog selected when creating the feed
- To make use of the "Item.saveCtgItem()" operation to save an item in a catalog other than the one specified in the import feed itself, you will need to put the following line at the top of the import script: setScriptContextValue("$action", null);
- Catalog import scripts also need to feed a "dummy" catalog if you wish to add or delete entry nodes. In other words, to add and/or delete entry nodes from Catalog A, the import feed itself must be mapped / attached to some other unused Catalog B.

# Mapping Script Expressions

Definition:

- Mappings are used during imports and exports to describe, respectively, how incoming data maps to the catalog and how data in the catalog maps to the output file
- Mapping expressions can be associated to individual attribute mappings to create complex mappings involving more than one field

Other details:

- Implicit object: item object (item)

## Entry Preview Scripts

Definition:

- The entry preview script allows a user to create a sample view of a current item set, which can be executed from the data entry screens
- For example, a script could be written to view how an item would display using an XML format

Other details:

- The operation "forEachCtgItem()" in the context of a preview script will run on each item selected in the data entry screen

## Catalog Scripts

Definition:

- A Catalog script is a sequence of operations that a user specifies to be run at the time of item creation and edit
- This function provides another layer of functionality over the attribute level operations available via catalog specs

Other details:

- Implicit object: item object (item)
- Catalog scripts can be run before or after any other operation (value rules, validation rules) on the item
- The order sequence might be important if there are any dependencies
- The order in which the catalog script will run depends on whether it is selected as a "pre" or "post" script
- Catalog scripts are typically used instead of value rules when more then one field is modified by a rule

## String Enumeration Rule Script Expressions

Definition:

- A String Enumeration Rule, like a value rule or a validation rule, is used within a specification
- A String Enumeration Rule can only be used with attributes of type "string enumeration"
- This rule is used to create a list of values available for that attribute

Other details:

- Implicit object: item object (item)
- Implicit variable: to set the attribute to the resulting value, use "res = "
- The value provided as a result must be an array

## Trigger Scripts

Definition:

- Trigger scripts are created to avoid the need to populate the same script operations in multiple places
- Trigger scripts are stored in the Document Store and can be called from another script function
- Used to externally trigger events in WPC (e.g. aggregations, syndications, etc.)

Other details:

- To run a trigger script from a browser type the corresponding URL; the URL consists of the WPC application URL, with the company code and the name of the script; for example: http://www.WPC.com/utils/invoker.jsp?company_code=<enterYouCompanyCodeHere>&script=<enterTriggerScriptNameHere>

## Mass Update Scripts

Definition:

- The mass update script allows for greater control over multiple updates for a group of items
- Mass Update Scripts run on a scheduler and can report on warnings and errors

Other details:

- Implicit object: item object (item)
- The mass update script will run for each item in the selection
- A "saveCtgItem()" is performed for each item modified

## Catalog Export Scripts

Definition:

- Catalog Export scripts are used to perform advanced, on-the-fly operations on data contained in the catalog before it is actually exported to an output file
- Modifications made to the content through the scripting engine at the time of syndication are not applied to the catalog, but rather simply applied to the output file as a one-time content modification
- All syndications require the use of a script
- Contrary to aggregation, selecting a script during syndication cannot be skipped
- However, for each new destination spec you create, three default generated scripts will be available to chose from: CSV, tab-delimited, and fixed-width

Other details:

- Implicit objects: file object (out), catalog object
- While the catalog object is implicit, there is not actually any implicit variable name through which to reference the catalog. To obtain a variable representing the catalog, simply execute a "getCtgByName()" passing in no arguments.
- In the context of an export, the operation "forEachCtgItem()" doesn't require the catalog name, it will by default get the Catalog Name selected when creating the syndication

## Value Rule Script Expressions

Definition:

- A value rule is created as a parameter of an attribute in a specification
- A value rule calculates the value of the attribute it to which it is attached
- When an item is create or saved, the value rule is computed

Other details:

- Implicit object: item object (item)
- Implicit variable: to set the attribute to the resulting value, use "res = "

- The result you provide must be in the format of the attribute (Example: if the attribute is of type "date" then the result must be in the date format)
- If you do not provide a value in your rule, the attribute value will be set to null

## Lookup Table Import Scripts

Definition:

- Lookup table scripts are very similar to aggregation scripts; they are used to populate the contents of a lookup table instead of a catalog
- Navigation to access lookup table import scripts is the same as catalog import scripts

## Category Tree Import Scripts

Definition:

- Category Tree Import scripts are used during category tree aggregation
- Although a user can create a category tree manually, the Category Tree Import script allows you to build a full category tree in WPC out of an incoming flat file

Other details:

- Implicit objects: file object (in), catalog object
- While the catalog object is implicit, there is not actually any implicit variable name through which to reference the catalog. To obtain a variable representing the catalog, simply execute a "getCtgByName()" passing in no arguments.
- An implicit "saveCategoryTree()" at the end of the aggregation saves all the new categories or modified categories
- The operation "getCategoryTreeByName()" in the context of a category tree feed doesn't require the tree name; it will by default get the Category Tree selected when creating the feed
- Make sure to use a path delimiter that you are not likely to find in your category names (for example, "/" might not be a good choice if you are likely to have categories containing this symbol; instead, select something less likely to be part of the category name (e.g. "////"))

## Predefined Scripts

### Search and Replace Macro Including Attribute Drop-down

# Box

Use the following steps to setup and use a search and replace macro for a catalog that includes a drop-down menu.

1. Navigate to the “Scripts Console” and select “Entry Macro Script”.

2. Click on the “NEW” button.

3. Select the catalog you would like the macro to appear in.

4. For “Select input parameters spec” click on “NEW”.

5. Enter spec name, e.g.: “S & R input spec”.

6. Click on the “+” to add a node to the spec. Call it “Attribute Path”. Create the node by clicking on the “+” to the right of the text box.

7. Set “Type” to “String enumeration” and then select “String enumeration rule” from the pulldown under “Type” and click on the “+” to the right of that pulldown to create the rule.

8. Click on the “trash can” next to “String enumeration”.

9. Next to “String enumeration rule”, click on “CLICK HERE” to enter the “String Enumeration Rule Editor”.

10. Copy the following script into the editor, replacing the text “Enter_Desired_Catalog_Spec_Name_Here” with the name of the catalog spec you require:

```
my_spec = getSpecByName("Enter_Desired_Catalog_Spec_Name_Here");
res = my_spec.getSpecAttribPaths();
```

11. Click on "Save" and then "Close".

12. Click on the “+” next to “S & R input spec” to add another node and name it “Search Pattern”. Click on the “+” next to the text box to create the new node and modify “Maximum Length” to the value “50”.

13. Click on the “+” next to “S & R input spec” to add another node and name it “Replace String”. Click on the “+” next to the text box to create the new node and modify “Maximum Length” to the value “50”.

14. Click on “SAVE” and then “<<|” to return to the previous screen.

15. Select the catalog you would like the macro to appear in. (as in step 3)

16. For “Select input parameters spec”, select the spec you just created: “S & R input spec”.

17. For “type” select “Regular”.

18. For “Entry Macro Script”, enter the name of the macro script, e.g.: “S & R macro”.

19. In the “Catalog Script Editor”, paste the following script:

```
attribPath = inputs["Attribute Path"];
attribValue = item.getCtgItemAttrib(attribPath);
myRe = new RE(inputs["Search Pattern"]);
newAttribValue = myRe.substitute(attribValue, inputs["Replace
String"]);
item.setCtgItemAttrib(attribPath, newAttribValue);
```

20. Click on “save” and then “<<|” to return to the Scripts Console.

**Usage**
From the Multiple Edit Data Entry screen, check the items you want to run the search and replace against, select “S & R macro” from the “MACRO” pull-down and click on “MACRO”. The macro will now work like the “Sample Replace String Macro Script” except you can now select the desired field via pull-down.

## Post-Save Audit Log Script

Here is the post-save audit log script that can be directly leveraged to support the Attribute Change Audit Logs for Catalogs:

```
oItem = item.getOriginalItem();
changedAttributes = item.getChangedAttributes(oItem);
catalogName = item.getCatalog().getCtgName();
userName = getCurrentUserName();

logName = "Item Audit Trail";
containerType = "CATALOG";
containerObject = item.getCatalog();
entryType = "ITEM";
entryObject = item;

userDefinedLog = getUserDefinedLogByName(logName, containerType,
containerObject);

changed = false;

logMessage = "<table width='100%' cellpadding=2 cellspacing=0 border=0
style='border-style:solid; border-width:1; border-color:#646464'><tr
bgcolor='#d3d3d3'><td bgcolor='#d8d8d8' width=120 rowspan='" +
changedAttributes.size() + "'> " + userName + "</td><td rowspan='" +
changedAttributes.size() + "' width=1 bgcolor='#646464'
style='padding:0'><img src='/locales/en_US/images/newlook/spacer.gif'
width=1 border=0></td>";
if(oItem != null)
{
    for(i=0; i<changedAttributes.size(); i++)
    {
```

 

```
        changed = true;
        string = catalogName + ":" + changedAttributes[i];

        oldValue = oItem.getCtgItemAttrib(string);
        newValue = item.getCtgItemAttrib(string);

        if(i!=0)
        {
            logMessage = logMessage + "<tr bgcolor='" + ((((i/2.0)-
(i/2))==0)? "#d0d0d0":"#dfdfdf") + "'>";
        }

        logMessage = logMessage + "<td>   " + changedAttributes[i] + "
was changed from [<b>"  + oldValue + "</b>] to [<b>" + newValue +
"</b>]<td></tr>";
    }
}
else
{
    if(item.getCtgItemId() == -1)
    {
        logMessage = logMessage + "<td>Item deleted [<b>" +
item.getCtgItemPrimaryKey() + "</b>]</td></tr>";
    }
    else
    {
        logMessage = logMessage + "<td>Item added [<b>" +
item.getCtgItemPrimaryKey() + "</b>]</td></tr>";
    }
    changed = true;
}

logMessage = logMessage + "</table>";
if(changed == true)
{
    udlLogMessage = new UserDefinedLogEntry(today(), containerType,
containerObject, entryType, entryObject, logMessage);
    userDefinedLog.insertEntryToLog(udlLogMessage);
}
```

# Notices

IBM may not offer the products, services, or features discussed in this document in all countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Burlingame Laboratory
Director IBM Burlingame Laboratory
577 Airport Blvd., Suite 800
Burlingame, CA 94010
U.S.A

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not necessarily tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information may contain examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples may include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

---

# Programming interface information

Programming interface information, if provided, is intended to help you create application software using this program.

General-use programming interfaces allow you to write application software that obtain the services of this program's tools.

However, this information may also contain diagnosis, modification, and tuning information. Diagnosis, modification and tuning information is provided to help you debug your application software.

Warning: Do not use this diagnosis, modification, and tuning information as a programming interface because it is subject to change.

---

## Trademarks and service marks

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States or other countries, or both:

IBM
the IBM logo
AIX
CrossWorlds
DB2
DB2 Universal Database
Domino
Lotus
Lotus Notes
MQIntegrator
MQSeries
Tivoli
WebSphere

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

MMX, Pentium, and ProShare are trademarks or registered trademarks of Intel Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product or service names may be trademarks or service marks of others.

This product includes software (Axis, Jakarta Commons Collection, Jakarta Commons DBCP, Jakarta Commons Pool, Jakarta Commons Regular Expression, Log4J, Regexp, Xalan, Xerces, XML4J) developed by the Apache Software Foundation (http://www.apache.org/).

CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or

bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

a. You must give any other recipients of the Work or Derivative Works a copy of this License; and

b. You must cause any modified files to carry prominent notices stating that You changed the files; and

c. You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those

notices that do not pertain to any part of the Derivative Works; and

d. If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

CUP Parser Generator Copyright Notice, License, and Disclaimer
Copyright 1996-1999 by Scott Hudson, Frank Flannery, C. Scott Ananian

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both the copyright notice and this permission notice and warranty disclaimer appear in supporting documentation, and that the names of the authors or their employers not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

The authors and their employers disclaim all warranties with regard to this software, including all implied warranties of merchantability and fitness. In no event shall the authors or their employers be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use or performance of this software.

JLEX COPYRIGHT NOTICE, LICENSE AND DISCLAIMER.

Copyright 1996-2003 by Elliot Joel Berk and C. Scott Ananian

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both the copyright notice and this permission notice and warranty disclaimer appear in supporting documentation, and that the name of the authors or their employers not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

The authors and their employers disclaim all warranties with regard to this software, including all implied warranties of merchantability and fitness. In no event shall the authors or their employers be liable for any special, indirect or consequential damages or any damages

whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use or performance of this software.

Java is a trademark of Sun Microsystems, Inc. References to the Java programming language in relation to JLex are not meant to imply that Sun endorses this product.