SAS® Life Science Analytics Framework: SAS® Macro API 2.6 Reference Guide

SAS® Documentation
April 26, 2022

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2022. *SAS® Life Science Analytics Framework: SAS® Macro API 2.6 Reference Guide*. Cary, NC: SAS Institute Inc.

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

April 2022

2.6-P1:lsafmapiref

Contents

PART 1

Overview

2

1

Common Macro Types, Values, and Parameters

Format Macro Values

For a macro with a parameter that specifies a URL, such as a macro that sets properties, you cannot embed an ampersand (&) in the URL. If you embed an ampersand, SAS attempts to resolve the subsequent text as a macro variable.

Although SAS is case-insensitive, the parameter values passed to SAS Life Science Analytics Framework might be case-sensitive.

To ensure consistent results, it is recommended that parameters of type String be specified using one of the string functions, such as %str() or %nrbquote(). Using double quotation marks results in a SAS syntax error.

Format Date Values

Dates that are specified as input, either as a macro parameter or in an input data set, must be in one of the following formats. Using other formats results in an execution error.

SAS DATETIME20.
An offset of zero hours from GMT is used. For example: `02NOV2013:09:33:22`

SAS E8601DZ25.
Use this format to specify an offset from GMT. For example:
`2013-11-02T09:33:22-04:00`

Dates that are returned as character variables in output data sets use the *EEE MMM dd HH:mm:ss zzz yyyy* Java format. When character variables are translated to numeric variables within the same data set, the SAS DATETIME. format is used.

Time lapse values, such as those that are returned by the %LSAF_GETJOBMANIFESTMETRICS macro, are returned in the *HH:mm:ss:SSS*. format.

Boolean Macros

Boolean macros are used to set a macro variable value based on a true/false inquiry against SAS Life Science Analytics Framework. If the call returns true, the applicable macro variable is set to `1`, otherwise the variable is set to `0`.

Boolean macros fail only under two conditions:

- The user calling the macro is not logged in. (This condition is applicable only when running from PC SAS.)
- A required parameter is missing.

In the following example, the values for LSAF_PATH, LSAF_MEMBER, and LSAF_GROUP_CONTEXT do not exist in the system. The call runs successfully and sets macro variable _lsafIsMember_ to `0`.

```
%lsaf_ismember(lsaf_path=/contextDoesNotExist, lsaf_member=mygroup,
   lsaf_type=GROUP,lsaf_group_context=/groupContextDoesNotExist);
NOTE: SAS Life Science Analytics Framework Macro: *
The GROUP /groupContextDoesNotExist mygroup is not a member of /contextDoesNotExist
NOTE: SAS Life Science Analytics Framework Macro: *  lsaf_ismember
NOTE: SAS Life Science Analytics Framework Macro: *  _lsafMsg_= Success.
NOTE: SAS Life Science Analytics Framework Macro: *  _lsafRC_= 0
NOTE: SAS Life Science Analytics Framework Macro: *  option lsaf_includeImplicit = 0
NOTE: SAS Life Science Analytics Framework Macro: *  _lsafIsMember_= 0
```

Value Macros

Value macros are used to set a macro variable to a non-Boolean value based on a value fetched from SAS Life Science Analytics Framework. In the event of a failure, the macro variable is set to blank.

In the following example, the call runs successfully and the _lsafType_ macro variable is set to **businessunit**:

```
%lsaf_gettype(lsaf_path=/SASMACROTESTS2);

NOTE: SAS Life Science Analytics Framework Macro: * The item type was returned for path:
   /SASMACROTESTS2
NOTE: SAS Life Science Analytics Framework Macro: *  lsaf_gettype
NOTE: SAS Life Science Analytics Framework Macro: * _lsafMsg_=Success.
NOTE: SAS Life Science Analytics Framework Macro: * _lsafRC_=0
NOTE: SAS Life Science Analytics Framework Macro: * _lsafType_=businessunit
```

Data Set Macros

Data set macros are used to return a SAS data set that contains a list of data retrieved from SAS Life Science Analytics Framework.

The parameter that specifies the name of the output data set is optional. If not provided, the macro sets the default value as WORK.*macro-name-with-no-underbar*. If specified, the value must be in the form <*libref*.>*dataset*.

In the event of a failure or if no records are found for the call, a data set with no observations is created. Any data set existing with the same <*libref*.>*dataset* name is overwritten with each call of the macro. If the specified data set name is invalid, the macro fails, _LSAFRC_ is not reset, no call to the system is made, and no data set is created.

Common Parameters

The following parameters are some that are commonly used for referencing, creating, and updating items in SAS Life Science Analytics Framework:

LSAF_PATH=*path*
: Specifies the path to an item in either the workspace or the repository. Paths are case-sensitive and must begin with /.

LSAF_VERSION=*version*
Indicates a specific version of a repository item. Version numbers are represented in the format *x.n*, where the decimal is required, for example, `1.2`, `2.0`. For macros that reference a versioned repository file, such as lsaf_getproperties, the parameter is optional. If the parameter is omitted, the latest version of the file is used. If the parameter is supplied and the specified version does not exist or the repository item is not versioned, the macro fails.

For macros that can also reference items in the workspace, such as lsaf_createweeklyschedule, the version number is ignored when the source location is `WORKSPACE`.

LSAF_OVERWRITE=`0` (Default) | `1`
Indicates whether an existing unversioned repository or existing workspace file is overwritten by a new file with a same path. When a versioned repository file exists, this argument is ignored and the parameters for version type and custom version, or, if not specified, their defaults are applied to the new version of the file.

Note: This description also applies to the LSAF_REPORTOVERWRITE argument.

LSAF_ENABLEVERSIONING=`0` (Default) | `1`
Indicates whether to enable versioning for a new file in the repository. If you are writing to the workspace or if the repository file already exists, this argument is ignored. This argument does not enable or disable versioning of an existing repository file.

Note: This description also applies to the LSAF_REPORTENABLEVERSIONING and LSAF_ENABLEVERSIONINGFORNEWFILES arguments.

LSAF_VERSIONTYPE=`MAJOR` | `MINOR` | `CUSTOM`
Specifies the version type to apply to new files or a new version of an existing versioned repository file. If the parameter is not specified, the defaults are applied. When you are writing to the workspace, or if the existing repository file is not versioned, this parameter is ignored.

Note: This description also applies to the LSAF_REPORTVERSIONTYPE, LSAF_VERSIONTYPEFORNEWFILES, and LSAF_VERSIONTYPEFOREXISTINGFILES arguments.

LSAF_CUSTOMVERSION=*version*
Specifies the version number to use to create a customized version of a new or existing versioned file. If the corresponding parameter for version type is CUSTOM, this argument is required, otherwise it is ignored. The parameter is ignored when you are writing to the workspace. See “LSAF_VERSION=version” for the expected format.

Note: This description also applies to the LSAF_REPORTCUSTOMVERSION, LSAF_CUSTOMVERSFORNEWFILES, and LSAF_CUSTOMVERSFOREXISTINGFILES arguments.

Job Parameter Override Values

The default values of job parameters can be overridden at the time of execution. Parameter override values are provided to applicable macros as an input data set, specified as *libref.>dataset*. If the data set is not provided, default job parameters are applied when the job is executed. The data set must contain the following character variables. Additional variables are ignored.

Table 1.1 *Parameter Override Variables*

Column Name	Description
name	Specifies the parameter name.
type	Specifies the parameter type. For a list of valid types, call %LSAF_GETSYSTEMCONSTANTS where `name=ParameterType`.
value	Specifies the parameter value to use when the job is executed. The value must match the parameter type. Date values must use the allowed formats. See “Format Date Values” for more information about date formats.
fileVersion	If the parameter type is FILE, this parameter specifies the file version to use. For versioned files, a blank value means that the latest file version is used. If the file is unversioned, is a workspace file, or the parameter type is not FILE, this value is ignored.
includeSubFolders	If the parameter type is FOLDER, this parameter indicates whether the subfolders of the container are included during job execution. If the parameter type is not FOLDER, this value is ignored.

The output data sets from calling either %LSAF_GETJOBPARAMETERS or %LSAF_GETWORKSPACEJOBPARAMETERS contains the data for all parameters that are defined for a specified job. That data set can be used as a source for the input data set for parameter overrides. To modify the data set for calling this macro:

- Rename the defaultValue variable to “value" or add a variable with the name "value" to indicate the values to use at the time of execution.
- Specify an empty string for override parameter values where a blank value is to be used.
- Delete records for parameters where the default value is to be used.

When the job is run, the values of the override parameters that match the names and types of parameters for the job are used. Override parameters that do not have a match on the job are ignored.

Standard Macro Variables

The following macro variables are set each time that a macro is run:

lsafMsg
: Contains the return message. If the macro ran without issue, the value is SUCCESS. Otherwise, the value is a detailed message that indicates the reason for the failure.

lsafRC
: Contains the return code. Valid values are based on the type of action that is performed by the macro and are listed by those actions in the following tables.

Note: Other macro variables might also be set and are noted along with the syntax for each macro.

Table 1.2 *Standard Return Codes*

Code	Description
0	The macro executed without error.
-1	The macro executed with an error. See _lsafMsg_ for the error details.
-100	There is no SAS Life Science Analytics Framework session. This is applicable only when calling a macro from PC SAS.
-500	An unexpected error occurred. This might result in a Java stack trace being printed in the SAS log file.
-999	No return code was set.

Table 1.3 *Query Service Return Codes*

Code	Description
0	The query ran successfully. The CSV file with the query results was created. The LSAF_VALIDATEDDATASET data set was created with no issues reported.
-200	Invalid records were found in the input data set for a query macro. The errors are reported in the LSAF_VALIDATEDDATASET data

Code	Description
	set. The query was not executed. Therefore, no CSV file was created.

Table 1.4 Clinical Data Import Return Codes

Code	Description
0	The import was successful. If specified, import details are reported in the sas_result_dsname data set and in the validation report.
-300	The import was successful, but issues were noted. If specified, the notes are included in the sas_result_dsname data set and the validation report. Each note is also printed to the SAS log file and is designated as a NOTE.
-301	The import was successful, but warnings were reported. If specified, warnings and notes are included in the sas_result_dsname data set and the validation report. Each warning or note is also printed to the SAS log file and is designated as a WARNING or NOTE, respectively.
-302	The import failed due to input data errors. If specified, errors, warnings, and notes are included in the sas_result_dsname data set and the validation report. Each error, warning, or note is also printed to the SAS log file and is designated as an ERROR, WARNING, or NOTE, respectively.

Note: In the event of an import failure, the sas_result_dsname data set is empty. No validation report is created.

2

Alphabetical Module List

Modules

Module	Description
AclService	Macros to manage owner and access permission information of a repository item.
AuditService	Macros to access the audit history metadata.
AuthenticationService	Macros to authenticate user access to SAS Life Science Analytics Framework.
AutoflowService	Macros to create autoflow process flows.
BatchRepositoryService	Macros to perform batch operations in the repository.
BatchWorkspaceService	Macros to perform batch operations in the workspace.
DictionaryService	Macros to get and export information that pertains to global dictionaries (controlled terminology).
ExternalDictionaryService	Macros to get and export dictionary information in SAS Life Science Analytics Framework.
GlobalPrivilegeService	Macros to manage the privileges that are available to user accounts at the global level.

Module	Description
GroupService	Macros to manage and create user groups, their availability in contexts, and their membership lists.
JobManifestService	Macros to retrieve the information from a job manifest file.
JobService	Macros to create, update, and retrieve information for jobs.
JobSubmissionService	Macros to execute jobs and to report the status of a submitted job.
MembershipService	Macros to manage and retrieve information about the members of a context.
ProcessDefinitionMappingService	Macros to manipulate process flow definition autoflow mapping.
ProcessDefinitionService	Macros to manage process flow definitions.
ProcessFlowService	Macros to manage process flows.
ProcessFlowSetupService	Macros to manage process flow setup elements.
ProcessFlowTaskService	Macros to manage process flow active tasks and elements.
QueryService	Macros to help construct and query data within SAS Life Science Analytics Framework. Query macros extract data based on a query built from an input data set. The extracted data is stored in a CSV file in the workspace or repository.
RecycleBinService	Macros to retrieve and permanently delete items in the recycle bin.
RepositoryService	Macros to manage content in the repository.
ResourceUtilizationService	Macros to get information about resource utilization.
RoleService	Macros to manage and create roles, their availability in contexts, and their membership lists.
ScheduleService	Macros to manage and create schedules.
ScopedPrivilegeService	Macros to manage the privileges that are available at the context membership level.

Module	Description
SessionService	Macros to manage and create interactive sessions.
SignatureService	Macros to retrieve electronic signature information for files that are in the repository.
StandardsService	Macros to get and export information that pertains to global standards.
StudyService	Macros to get and update study information.
StudyTlfService	Macros to manipulate the Tables, Figures, and Listings (TLFs) of a study.
SubscriptionService	Macros to manage subscriptions.
SystemService	Macros to report the configuration of the system.
TLFTemplateService	Macros to manage the global Titles, Listings, and Figures (TLF) templates.
TypeService	Macros to describe the item type definitions.
UserService	Macros to manage and create user accounts.
UtilityService	Macros to derive relative and absolute paths.
WorkspaceService	Macros to manage content in the workspace.

3

Alphabetical Macro List

A - C

%LSAF_ACTIVATEPROCESSDEFINITION

%LSAF_ACTIVATEPROCESSFLOW

%LSAF_ADDGROUPMEMBER

%LSAF_ADDINHERITEDROLE

%LSAF_ADDMEMBER

%LSAF_ADDROLEMEMBER

%LSAF_ADDROLEPRIVILEGE

%LSAF_ADDSTUDYEXTDICTIONARY

%LSAF_ADDSTUDYSTANDARD

D - F

G

H - L

M - R

S - T

U - Z

%LSAF_USEREXISTS

PART 2

Clinical Macros

4

DictionaryService Module

Overview

Macros to get and export information that pertains to global dictionaries (controlled terminology).

Macro Summary Table

Category	Language Elements	Description
DictionaryService Module	%LSAF_EXPORTDICTIONARY Macro (p. 34)	Exports the metadata for a controlled terminology dictionary. The metadata is stored in a SAS data set based on the specified location (WORKSPACE or REPOSITORY).
	%LSAF_GETDICTIONARIESBYSTATUS Macro (p. 35)	Gets the metadata for all of the dictionaries by the specified status. The metadata is stored in a SAS data set.
	%LSAF_GETDICTIONARYID Macro (p. 37)	Gets the dictionary identifier for a controlled terminology dictionary.

Dictionary

%LSAF_EXPORTDICTIONARY Macro

Exports the metadata for a controlled terminology dictionary. The metadata is stored in a SAS data set based on the specified location (WORKSPACE or REPOSITORY).

Category: DictionaryService Module

Note: This macro sets the standard macro variables and _lsafDictionaryPath_, which is the full path of the dictionary data set.

Syntax

%LSAF_EXPORTDICTIONARY(LSAF_DICTIONARYID=*id*, LSAF_STATUS=*entity-status*, LSAF_LOCATION=WORKSPACE | REPOSITORY, LSAF_PATH=*path* <, LSAF_DICTIONARYVERSION=*version*><, LSAF_OVERWRITE=0 | 1><, LSAF_ENABLEVERSIONING=0 | 1><, LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM><, LSAF_CUSTOMVERSION=*version*><, LSAF_COMMENT=*comment*>);

Required Arguments

LSAF_DICTIONARYID=*id*
The identifier of the dictionary. To get the identifier, call %LSAF_GETDICTIONARYID.

LSAF_STATUS=*entity-status*
The case-insensitive status of the dictionary. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=EntityStatus`.

LSAF_LOCATION=WORKSPACE | REPOSITORY
The case-insensitive output location for the dictionary file.

LSAF_PATH=*path*
The output path for the dictionary file. If the file extension is not specified, it is added during processing.

Optional Arguments

LSAF_DICTIONARYVERSION=*version*
The version of a PUBLISHED dictionary to export. If the status is not PUBLISHED, this argument is ignored.

LSAF_OVERWRITE=1 | 0
Indicates whether an existing unversioned repository or existing workspace exported file is overwritten by the file with a same path. See “LSAF_OVERWRITE=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_ENABLEVERSIONING=0 | 1
Indicates whether a new exported file is versioned. See “LSAF_ENABLEVERSIONING=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM
The case-insensitive type for a versioned file in the repository. See “LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM” on page 6 for more information about this argument.

LSAF_CUSTOMVERSION=*version*
If LSAF_VERSIONTYPE is CUSTOM, the specified version value. See “LSAF_CUSTOMVERSION=version” on page 6 for more information about this argument.

LSAF_COMMENT=*comment*
The check-in comment to associate with the action of adding the data set to the repository.If LSAF_LOCATION is WORKSPACE, this argument is ignored.

Details

Sets the value of the macro variable _lsafDictionaryPath_ to the full path of the dictionary data set. The path includes the file extension. See “Value Macros” on page 5 for information about value macros.

%LSAF_GETDICTIONARIESBYSTATUS Macro

Gets the metadata for all of the dictionaries by the specified status. The metadata is stored in a SAS data set.

Category: DictionaryService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETDICTIONARIESBYSTATUS(LSAF_STATUS=*entity-status*<, SAS_DSNAME=*name*>);

Required Argument

LSAF_STATUS=*entity-status*
The status of the dictionaries to return. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=EntityStatus`.

Optional Argument

SAS_DSNAME=*name*

The name of the output data set. See “Data Set Macros” on page 5 for information about specifying the name of the data set.

The data set contains a row for each dictionary, sorted by the dictionary identifier, and the following columns:

Column Name	Description
id	The unique dictionary identifier.
name	The dictionary name.
standard	The associated data standard.
description	The dictionary description.
isCheckedOut	Indicates whether the dictionary is checked out.
checkedOutBy	If the dictionary is checked out, the identifier of the user who has it checked out.
checkedOutDate	If the dictionary is checked out, the date on which the dictionary was checked out.
checkedOutDateSASFormat	If the dictionary is checked out, the date on which the dictionary was checked out (SAS format).
createdDate	The date on which the dictionary was created.
createdDateSASFormat	The date on which the dictionary was created (in SAS format).
createdBy	The identifier of the user who created the dictionary.
state	The state of the dictionary. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=EntityStatus`.
publishedState	The published state of the dictionary. This value is blank if the dictionary has never been published.

Column Name	Description
currentVersion	The current published version of the dictionary.
associatedStudiesCount	The number of studies that are associated with the dictionary.
lastModifiedDate	The date on which the dictionary was last modified.
lastModifiedDateSASFormat	The date on which the dictionary was last modified (in SAS format).
lastModifiedBy	The identifier of the user of who last modified the dictionary.
releaseDate	The date on which the dictionary was released.
dictionaryStatus	The status of the dictionary. This value is the same as the value that was specified for LSAF_STATUS.

Note: See "Format Date Values" on page 4 for information about date formats.

%LSAF_GETDICTIONARYID Macro

Gets the dictionary identifier for a controlled terminology dictionary.

Category: DictionaryService Module

Note: This macro sets the standard macro variables and _lsafDictionaryId_, which is the dictionary identifier.

Syntax

%LSAF_GETDICTIONARYID(LSAF_NAME=*name*<, LSAF_STATUS=*entity-status*>);

Required Argument

LSAF_NAME=*name*
The case-sensitive name of the dictionary.

Optional Argument

LSAF_STATUS=*entity-status*
The case-insensitive status of the dictionary. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=EntityStatus`.

Details

Sets the value of the macro variable _lsafDictionaryId_ to the dictionary identifier. See “Value Macros” on page 5 for information about value macros.

5

ExternalDictionaryService Module

Overview

Macros to get and export dictionary information in SAS Life Science Analytics Framework.

Macro Summary Table

Category	Language Elements	Description
ExternalDictionaryService Module	%LSAF_EXPORTEXTDICTS Macro (p. 40)	Exports the metadata for external dictionaries. The metadata is stored in a SAS data set in the specified location (WORKSPACE or REPOSITORY).
	%LSAF_GETEXTDICTIONARYID Macro (p. 41)	Gets the dictionary identifier for an external dictionary.

Category	Language Elements	Description
	%LSAF_GETEXTERNALDICTIONARIES Macro (p. 42)	Gets the metadata for all of the external dictionaries. The metadata is stored in a SAS data set.

Dictionary

%LSAF_EXPORTEXTDICTS Macro

Exports the metadata for external dictionaries. The metadata is stored in a SAS data set in the specified location (WORKSPACE or REPOSITORY).

Category: ExternalDictionaryService Module

Note: This macro sets the standard macro variables and _lsafExtDictionaryPath_, which is the full path of the external dictionary data set.

Syntax

%LSAF_EXPORTEXTDICTS(LSAF_LOCATION=WORKSPACE | REPOSITORY, LSAF_PATH=*path*<, LSAF_OVERWRITE=0 | 1><, LSAF_ENABLEVERSIONING=0 | 1><, LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM><, LSAF_CUSTOMVERSION=*version*><, LSAF_COMMENT=*comment*>);

Required Arguments

LSAF_LOCATION=WORKSPACE | REPOSITORY
The case-insensitive output location for the exported external dictionary file.

LSAF_PATH=*path*
The output path for the data set that contains the exported external dictionaries. If the file extension is not specified, it is added during processing.

Optional Arguments

LSAF_OVERWRITE=0 (Default) | 1
Indicates whether to overwrite an existing unversioned data set. See “LSAF_OVERWRITE=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_ENABLEVERSIONING=0 (Default) | 1
Indicates whether a new exported file is versioned. See “LSAF_ENABLEVERSIONING=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM
The version type applied to a versioned file. See "LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM" on page 6 for more information about this argument.

LSAF_CUSTOMVERSION=*version*
The version number to use to create a customized versioned file. See "LSAF_CUSTOMVERSION=version" on page 6 for more information about this argument.

LSAF_COMMENT=*comment*
The check-in comment to associate with the action of adding the data set to the repository. If LSAF_LOCATION is WORKSPACE, this argument is ignored.

Details

Sets the value of the macro variable _lsafExtDictionaryPath_ to the full path of the external dictionary data set, which includes the file extension. See "Value Macros" on page 5 for information about value macros.

%LSAF_GETEXTDICTIONARYID Macro

Gets the dictionary identifier for an external dictionary.

Category: ExternalDictionaryService Module

Note: This macro sets the standard macro variables and _lsafExtDictionaryId_, which is the external dictionary identifier.

Syntax

%LSAF_GETEXTDICTIONARYID(LSAF_DICTIONARY=*dictionary-name*, LSAF_VERSION=*version*, LSAF_DATATYPE=*core-data-type*);

Required Arguments

LSAF_DICTIONARY=*dictionary-name*
The case-sensitive name of the dictionary. This is the value listed in the dictionary column in the data set that is generated by %LSAF_GETEXTERNALDICTIONARIES.

LSAF_VERSION=*version*
The referenced version of the external dictionary. See "LSAF_VERSION=version" on page 6 for more information about this argument.

LSAF_DATATYPE=*core-data-type*
The data type of the external dictionary. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=DataType` and `source=core.DataType`.

Details

Sets the value of the macro variable _lsafExtDictionaryId_ to the external dictionary identifier. See “Value Macros” on page 5 for information about value macros. To get the dictionary name, status, and data type, call %LSAF_GETEXTERNALDICTIONARIES.

%LSAF_GETEXTERNALDICTIONARIES Macro

Gets the metadata for all of the external dictionaries. The metadata is stored in a SAS data set.

Category: ExternalDictionaryService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETEXTERNALDICTIONARIES(<SAS_DSNAME=*name*>);

Optional Argument

SAS_DSNAME=*name*

The name of the output data set that contains the metadata for the external dictionaries. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETEXTERNALDICTIONARIES.

The data set contains a row for each external dictionary, sorted by the external dictionary name, and the following columns:

Column Name	Description
id	The unique external dictionary identifier.
name	The external dictionary name.
description	The external dictionary description.
dataType	The external dictionary data type. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=DataType` and `source=core.DataType`.
shortName	The short name of the external dictionary.
dictionary	The external dictionary.

Column Name	Description
version	The current published version of the external dictionary.
href	The HTTP reference for the external dictionary.
createdDate	The date on which the external dictionary was created.
createdDateSASFormat	The date on which the external dictionary was created (in SAS format).
createdBy	The identifier of the user who created the external dictionary.
associatedStudiesCount	The number of studies that are associated with the external dictionary.
lastModifiedDate	The date on which the external dictionary was last modified.
lastModifiedDateSASFormat	The date on which the external dictionary was last modified (in SAS format).
lastModifiedBy	The identifier of the user of who last modified the external dictionary.

Note: See "Format Date Values" on page 4 for information about date formats.

6

StandardsService Module

Overview

Macros to get and export information that pertains to global standards.

Macro Summary Table

Category	Language Elements	Description
StandardsService Module	%LSAF_EXPORTSTANDARD Macro (p. 46)	Exports the metadata for a global standard. The metadata is stored in a SAS data set in the repository. The paths of the exported files are stored in a SAS data set.
	%LSAF_EXPORTSTANDARDTOWS Macro (p. 48)	Exports to the workspace of the current user the metadata for a global data standard. The paths of the exported files are stored in a SAS data set.
	%LSAF_GETSTANDARDID Macro (p. 49)	Gets the identifier of a global standard.

Category	Language Elements	Description
	%LSAF_GETSTANDARDSBYSTATUS Macro (p. 50)	Gets the metadata for all of the data standards by the specified status. The metadata is stored in a SAS data set.

Dictionary

%LSAF_EXPORTSTANDARD Macro

Exports the metadata for a global standard. The metadata is stored in a SAS data set in the repository. The paths of the exported files are stored in a SAS data set.

Category: StandardsService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_EXPORTSTANDARD(LSAF_STANDARDID=*standard-id*, LSAF_STATUS=*status*, LSAF_PATH=*path*, LSAF_PREFIX=*prefix*<, LSAF_STANDARDVERSION=*version*><, LSAF_OVERWRITE=0 | 1><, LSAF_ENABLEVERSIONING=0 | 1><, LSAF_VERSIONTYPEFORNEWFILES=MINOR | MAJOR | CUSTOM><, LSAF_CUSTOMVERSFORNEWFILES=*version*><, LSAF_VERSIONTYPEFOREXISTINGFILES=MINOR | MAJOR | CUSTOM><, LSAF_CUSTOMVERSFOREXISTINGFILES=*version*><, LSAF_COMMENT=*comment*><, SAS_RESULT_DSNAME=*name*>);

Required Arguments

LSAF_STANDARDID=*standard-id*
The identifier of the data standard. To get the identifier, call %LSAF_GETSTANDARDID.

LSAF_STATUS=*status*
The status of the standard. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=EntityStatus`.

LSAF_PATH=*path*
The output path for the exported files.

LSAF_PREFIX=*prefix*
The prefix of the SAS data sets that are created for the exported data standards. All text is converted to lowercase.

Optional Arguments

LSAF_STANDARDVERSION=*version*
The version of a PUBLISHED standard to export. If status is not PUBLISHED, this argument is ignored.

LSAF_OVERWRITE=0 (Default) | 1
Indicates whether existing unversioned data sets are overwritten. See "LSAF_OVERWRITE=0 (Default) | 1" on page 6 for more information about this argument.

LSAF_ENABLEVERSIONING=0 (Default) | 1
Indicates whether a new exported file is versioned. See "LSAF_ENABLEVERSIONING=0 (Default) | 1" on page 6 for more information about this argument.

LSAF_VERSIONTYPEFORNEWFILES=MAJOR | MINOR | CUSTOM
If enableversioning is True, the version type to use to create a new file. See "LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM" on page 6 for more information about this argument.

LSAF_CUSTOMVERSFORNEWFILES=*version*
The version number to use to create a customized versioned file. See "LSAF_CUSTOMVERSION=version" on page 6 for more information about this argument.

LSAF_VERSIONTYPEFOREXISTINGFILES=MAJOR | MINOR | CUSTOM
The version type for an existing file. See "LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM" on page 6 for more information about this argument.

LSAF_CUSTOMVERSFOREXISTINGFILES=*version*
The version number to use for an existing customized versioned file. See "LSAF_CUSTOMVERSION=version" on page 6 for more information about this argument.

LSAF_COMMENT=*comment*
The check-in comment to associate with adding the data sets to the repository.

SAS_RESULT_DSNAME=*name*
The name of the output data set that contains the path, name, and location of the exported data standards SAS data sets. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFEXPORTSTANDARD.

The data set contains a row for each export data set, sorted by path, and the following columns:

Column Name	Description
location	The location of the exported file (REPOSITORY).
name	The name of the exported file.
path	The path of the exported file.

%LSAF_EXPORTSTANDARDTOWS Macro

Exports to the workspace of the current user the metadata for a global data standard. The paths of the exported files are stored in a SAS data set.

Category: StandardsService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_EXPORTSTANDARDTOWS(LSAF_STANDARDID=*standard-id*, LSAF_STATUS=*status*, LSAF_PATH=*path*, LSAF_PREFIX=*prefix*<, LSAF_STANDARDVERSION=*version*><, LSAF_OVERWRITE=0 | 1><, SAS_RESULT_DSNAME=*name*>);

Required Arguments

LSAF_STANDARDID=*standard-id*
The identifier of the data standard. To get the identifier, call %LSAF_GETSTANDARDID.

LSAF_STATUS=*status*
The status of the data standard to export. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=EntityStatus`.

LSAF_PATH=*path*
The output path for the export files.

LSAF_PREFIX=*prefix*
The prefix for the SAS data sets that are created for the exported data standards. All letters are converted to lowercase.

Optional Arguments

LSAF_STANDARDVERSION=*version*
The version of a PUBLISHED data standard to export. If the status is not PUBLISHED, this argument is ignored.

LSAF_OVERWRITE=0 (Default) | 1
Indicates whether existing data sets are overwritten. See “LSAF_OVERWRITE=0 (Default) | 1” on page 6 for more information about this argument.

SAS_RESULT_DSNAME=*name*
The name of the output data set that contains the path, name, and location of the exported data standards SAS data sets. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFEXPORTSTANDARDTOWS.

The data set contains a row for the export data set, sorted by path, and the following columns:

Column Name	Description
location	The location of the exported file (WORKSPACE).
name	The name of the exported file.
path	The path of the exported file.

%LSAF_GETSTANDARDID Macro

Gets the identifier of a global standard.

Category: StandardsService Module

Note: This macro sets the standard macro variables and _lsafStandardId_, which is the standard identifier.

Syntax

%LSAF_GETSTANDARDID(LSAF_NAME=*name*<, LSAF_STATUS=*status*>);

Required Argument

LSAF_NAME=*name*
The case-sensitive name of the standard.

Optional Argument

LSAF_STATUS=*status*
The status of the standard. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=EntityStatus`.

Details

Sets the value of the macro variable _lsafStandardId_ to the standard identifier. See “Value Macros” on page 5 for information about value macros.

%LSAF_GETSTANDARDSBYSTATUS Macro

Gets the metadata for all of the data standards by the specified status. The metadata is stored in a SAS data set.

Category: StandardsService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETSTANDARDSBYSTATUS(LSAF_STATUS=*status*<, SAS_DSNAME=*name*>);

Required Argument

LSAF_STATUS=*status*
The status of the data standards to return. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=EntityStatus`.

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the data standards. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETSTANDARDSBYSTATUS.

The data set contains a row for each data standard, sorted by the standard identifier, and the following columns:

Column Name	Description
id	The unique standard identifier.
name	The data standard name.
baseStandardType	The base data standard type.
baseStandardName	The base data standard name.
baseStandardVersion	The base data standard version.
modelId	The model identifier.
description	The data standard description.

Column Name	Description
isCheckedOut	Indicates whether the data standard is checked out.
checkedOutBy	If the data standard is checked out, the identifier of the user who has it checked out.
checkedOutDate	If the data standard is checked out, the date on which it was checked out.
checkedOutDateSASFormat	If the data standard is checked out, the date on which it was checked out (in SAS format).
createdDate	The date on which the data standard was created.
createdDateSASFormat	The date on which the data standard was created (in SAS format).
createdBy	The identifier of the user who created the data standard.
state	The state of the data standard. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=EntityState`.
publishedState	The published state of the data standard. This value is blank if the data standard has never been published.
currentVersion	The current published version of the data standard.
associatedStudiesCount	The number of studies that are associated with the data standard.
lastModifiedDate	The date on which the data standard was last modified.
lastModifiedDateSASFormat	The date on which the data standard was last modified (in SAS format).
lastModifiedBy	The identifier of the user who last modified the data standard.
standardStatus	The status of the data standard. This value is the same as the value that was specified for LSAF_STATUS.

Note: See “Format Date Values” on page 4 for information about date formats.

7

StudyService Module

Overview

Macros to get and update study information.

Macro Summary Table

Category	Language Elements	Description
StudyService Module	%LSAF_ADDSTUDYEXTDICTIONARY Macro (p. 57)	Adds an external dictionary to a study.
	%LSAF_ADDSTUDYSTANDARD Macro (p. 57)	Adds a standard to a study.
	%LSAF_CREATEEMPTYSTUDYTABLES Macro (p. 58)	Creates a zero-observation data set in the repository for each table in a study.
	%LSAF_CREATEEMPTYSTUDYTABLESINWS Macro (p. 59)	Creates a zero-observation data set in the workspace for each table in a study.
	%LSAF_EXPORTSTUDYCODELISTS Macro (p. 60)	Exports the metadata for the study code lists. The metadata is stored in a SAS data set in the workspace or repository.
	%LSAF_EXPORTSTUDYDETAILS Macro (p. 61)	Exports the metadata for the study details. The metadata is stored in a SAS data set that is located in the workspace or the repository.
	%LSAF_EXPORTSTUDYDICTS Macro (p. 63)	Exports the metadata for the study controlled terminology dictionaries. The metadata is stored in a SAS data set in the workspace or repository.
	%LSAF_EXPORTSTUDYDOCS Macro (p. 64)	Exports the metadata for the study supporting documents. The metadata is stored in a SAS data set in the workspace or repository.
	%LSAF_EXPORTSTUDYEXTDICTS Macro (p. 65)	Exports the metadata for the study external dictionaries. The metadata is stored in a SAS data set in the workspace or repository.

Category	Language Elements	Description
	%LSAF_EXPORTSTUDYSTANDARDS Macro (p. 66)	Exports the metadata for the study standards. The metadata is stored in a SAS data set in the workspace or repository.
	%LSAF_EXPORTSTUDYTABLES Macro (p. 68)	Exports the metadata for the study tables to the specified area in the repository. The metadata is stored in SAS data sets. The paths of the exported files are stored in a SAS data set.
	%LSAF_EXPORTSTUDYTABLESTOWS Macro (p. 69)	Exports the metadata for the study tables to a location in the workspace. The metadata is stored in SAS data sets. The paths of the exported files are stored in a SAS data set.
	%LSAF_EXPORTSTUDYTLFS Macro (p. 71)	Exports the metadata for the study tables, listings, and figures to the specified area in the repository. The metadata is stored in SAS data sets. The paths of the exported files are stored in a SAS data set.
	%LSAF_EXPORTSTUDYTLFSTOWORKSPACE Macro (p. 72)	Exports the metadata for the study tables, listings, and figures to the specified area in the workspace. The metadata is stored in SAS data sets. The paths of the exported files are stored in a SAS data set.
	%LSAF_EXPORTSTUDYVLM Macro (p. 74)	Exports the metadata for the study value level metadata (VLM). The metadata is stored in a SAS data set in the workspace or repository.
	%LSAF_GETDICTIONARYSTUDIES Macro (p. 75)	Gets the metadata for all of the studies that are associated with a dictionary. The metadata is stored in a SAS data set.
	%LSAF_GETEXTDICTIONARYSTUDIES Macro (p. 76)	Gets the metadata for all of the studies that are associated with an external dictionary. The metadata is stored in a SAS data set.
	%LSAF_GETSTANDARDSTUDIES Macro (p. 77)	Gets the metadata for all of the studies that are associated with a data standard. The metadata is stored in a SAS data set.
	%LSAF_GETSTDYCOMPONENTEDITEDBY Macro (p. 78)	Gets the name of the user who is the current editor for a study component.
	%LSAF_GETSTUDYDICTIONARIES Macro (p. 79)	Gets the metadata for all of the controlled terminology dictionaries that are associated with a study. The metadata is stored in a SAS data set.
	%LSAF_GETSTUDYEXTDICTIONARIES Macro (p. 81)	Gets the metadata for all of the external dictionaries that are associated with a study. The metadata is stored in a SAS data set.
	%LSAF_GETSTUDYID Macro (p. 82)	Gets the identifier for a study. You must have Read access permission to the study.
	%LSAF_GETSTUDYSTANDARDS Macro (p. 83)	Gets the metadata for all of the standards that are associated with a study. The metadata is stored in a SAS data set.

Category	Language Elements	Description
	%LSAF_IMPORTSTUDYCODELISTS Macro (p. 85)	Imports study codelists from the specified data set, and can create a validation report as a CSV file in the specified location.
	%LSAF_IMPORTSTUDYDETAILS Macro (p. 87)	Imports study details from the specified SAS data set.
	%LSAF_IMPORTSTUDYDOCS Macro (p. 88)	Imports study supporting documents from the specified data set and can create a validation report as a CSV file in the specified location.
	%LSAF_IMPORTSTUDYFROMDEFINEXML Macro (p. 90)	Imports metadata from a Define-XML file and updates study components based on the contents of the file. It can create a validation report as a CSV file in the specified location.
	%LSAF_IMPORTSTUDYTABLES Macro (p. 92)	Imports study tables from the specified data sets and can create a validation report as a CSV file in the specified location.
	%LSAF_IMPORTSTUDYTLFS Macro (p. 94)	Imports study tables, figures, and listings from the specified data sets and can create a validation report as a CSV file in the specified location.
	%LSAF_IMPORTSTUDYVLM Macro (p. 97)	Imports study value level metadata from the specified data set and can create a validation report as a CSV file in the specified location.
	%LSAF_ISSTUDYDICTIONARY Macro (p. 99)	Indicates whether a controlled terminology dictionary is associated with a study.
	%LSAF_ISSTUDYEXTDICTIONARY Macro (p. 99)	Indicates whether an external dictionary is associated with a study.
	%LSAF_ISSTUDYSTANDARD Macro (p. 100)	Indicates whether a standard is associated with a study.
	%LSAF_REMOVEALLSTUDYDICTIONARIES Macro (p. 101)	Removes all controlled terminology dictionaries from a study.
	%LSAF_REMOVESTUDYEXTDICTIONARY Macro (p. 101)	Removes an external dictionary from a study.
	%LSAF_REMOVESTUDYSTANDARD Macro (p. 102)	Removes a standard from a study.
	%LSAF_SETSTDYCOMPONENTEDITSTATUS Macro (p. 102)	Sets the component edit status for a study.
	%LSAF_UPDATESTUDYDICTIONARIES Macro (p. 103)	Updates the controlled terminology dictionaries for a study using a SAS data set as input.

Dictionary

%LSAF_ADDSTUDYEXTDICTIONARY Macro

Adds an external dictionary to a study.

Category: StudyService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_ADDSTUDYEXTDICTIONARY(LSAF_STUDYID=*study-id*, LSAF_EXTDICTIONARYID=*external-dictionary-id*);

Required Arguments

LSAF_STUDYID=*study-id*
The identifier of the study. To get the identifier, call %LSAF_GETSTUDYID.

LSAF_EXTDICTIONARYID=*external-dictionary-id*
The identifier of the external dictionary. To get the identifier, call %LSAF_GETEXTDICTIONARYID.

%LSAF_ADDSTUDYSTANDARD Macro

Adds a standard to a study.

Category: StudyService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_ADDSTUDYSTANDARD(LSAF_STUDYID=*study-id*, LSAF_STANDARDID=*standard-id*);

Required Arguments

LSAF_STUDYID=*study-id*
The identifier of the study. To get the identifier, call %LSAF_GETSTUDYID.

LSAF_STANDARDID=*standard-id*
The identifier of the standard. To get the identifier, call %LSAF_GETSTANDARDID.

%LSAF_CREATEEMPTYSTUDYTABLES Macro

Creates a zero-observation data set in the repository for each table in a study.

Category: StudyService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_CREATEEMPTYSTUDYTABLES(LSAF_STUDYID=*study-id*, LSAF_BASESTANDARDNAME=*name*, LSAF_MODELID=*model-id*, LSAF_PATH=*path*<, LSAF_OVERWRITE=0 | 1><, LSAF_ENABLEVERSIONING=0 | 1><, LSAF_VERSIONTYPEFORNEWFILES=MAJOR | MINOR | CUSTOM><, LSAF_CUSTOMVERSFORNEWFILES=*version*><, LSAF_VERSIONTYPEFOREXISTINGFILES=MINOR | MAJOR | CUSTOM><, LSAF_CUSTOMVERSFOREXISTINGFILES=*version*><, LSAF_COMMENT=*comment*>);

Required Arguments

LSAF_STUDYID=*study-id*
The identifier of the study. To get the identifier, call %LSAF_GETSTUDYID.

LSAF_BASESTANDARDNAME=*name*
The case-sensitive name of the base standard, such as SDTM and ADaM.

LSAF_MODELID=*model-id*
The identifier of the model. To get the identifier, call %LSAF_GETSTUDYSTANDARDS.

LSAF_PATH=*path*
The case-sensitive output path for the exported files

Optional Arguments

LSAF_OVERWRITE=0 (Default) | 1
Indicates whether existing unversioned data sets are overwritten. See “LSAF_OVERWRITE=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_ENABLEVERSIONING=0 (Default) | 1
Indicates whether versioning is enabled for new exported files. See “LSAF_ENABLEVERSIONING=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_VERSIONTYPEFORNEWFILES=MAJOR (Default) | MINOR | CUSTOM
If enableversioning is True, the version type to use to create a file. See “LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM” on page 6 for more information about this argument.

LSAF_CUSTOMVERSFORNEWFILES=*version*
The version number to use to create a customized versioned file. See “LSAF_CUSTOMVERSION=version” on page 6 for more information about this argument.

LSAF_VERSIONTYPEFOREXISTINGFILES=MAJOR | MINOR (Default) | CUSTOM
The version type for an existing file. See “LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM” on page 6 for more information about this argument.

LSAF_CUSTOMVERSFOREXISTINGFILES=*version*
The version number to use for an existing customized versioned file. See “LSAF_CUSTOMVERSION=version” on page 6 for more information about this argument.

LSAF_COMMENT=*comment*
The check-in comment to associate with adding the data sets to the repository.

Details

Creates a zero-observation data set in the repository for each table in a study.

To get the base standard name and model identifier, call %LSAF_GETSTUDYSTANDARDS.

%LSAF_CREATEEMPTYSTUDYTABLESINWS Macro

Creates a zero-observation data set in the workspace for each table in a study.

Category: StudyService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_CREATEEMPTYSTUDYTABLESINWS(LSAF_STUDYID=*study-id*, LSAF_BASESTANDARDNAME=*name*, LSAF_MODELID=*model-id*, LSAF_PATH=*lsaf-path*<, LSAF_OVERWRITE=0 (Default) | 1>);

Required Arguments

LSAF_STUDYID=*study-id*
The identifier of the study. To get the identifier, call %LSAF_GETSTUDYID.

LSAF_BASESTANDARDNAME=*name*
The case-sensitive name of the base standard, such as SDTM and ADaM.

LSAF_MODELID=*model-id*
The identifier of the model. To get the identifier, call %LSAF_GETSTUDYSTANDARDS.

LSAF_PATH=*lsaf-path*
The case-sensitive output path for the exported files.

Optional Argument

LSAF_OVERWRITE=0 (Default) | 1
Indicates whether existing data sets are overwritten. See “LSAF_OVERWRITE=0 (Default) | 1” on page 6 for more information about this argument.

Details

Creates a zero-observation data set in the workspace for each table in a study.

To get the base standard name and model identifier, call %LSAF_GETSTUDYSTANDARDS.

%LSAF_EXPORTSTUDYCODELISTS Macro

Exports the metadata for the study code lists. The metadata is stored in a SAS data set in the workspace or repository.

Category: StudyService Module

Note: This macro sets the standard macro variables and _lsafexportstudycodelistspath_, which is the full path of the code lists data set.

Syntax

%LSAF_EXPORTSTUDYCODELISTS(LSAF_STUDYID=*study-id*, LSAF_LOCATION=WORKSPACE | REPOSITORY, LSAF_PATH=*path*<, LSAF_OVERWRITE=0 | 1><, LSAF_ENABLEVERSIONING=0 | 1><, LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM><, LSAF_CUSTOMVERSION=*version*><, LSAF_COMMENT=*comment*>);

Required Arguments

LSAF_STUDYID=*study-id*
The identifier of the study. To get the identifier, call %LSAF_GETSTUDYID.

LSAF_LOCATION=WORKSPACE | REPOSITORY
The case-insensitive output location for the exported file.

LSAF_PATH=*path*
The case-sensitive output path for the exported files.

Optional Arguments

LSAF_OVERWRITE=0 (Default) | 1
Indicates whether existing unversioned data sets are overwritten. See "LSAF_OVERWRITE=0 (Default) | 1" on page 6 for more information about this argument.

LSAF_ENABLEVERSIONING=0 (Default) | 1
Indicates whether versioning is enabled for new exported files. See "LSAF_ENABLEVERSIONING=0 (Default) | 1" on page 6 for more information about this argument.

LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM
The case-insensitive type for a versioned file in the repository. See "LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM" on page 6 for more information about this argument.

LSAF_CUSTOMVERSION=*version*
If LSAF_VERSIONTYPE is CUSTOM, the specified version value. See "LSAF_CUSTOMVERSION=version" on page 6 for more information about this argument.

LSAF_COMMENT=*comment*
The check-in comment to associate with the action of adding the data set to the repository. If LSAF_LOCATION is WORKSPACE, this argument is ignored.

Details

Sets the value of the macro variable _lsafexportstudycodelistspath_ to the full path of the code lists data set, which includes the file extension. See "Value Macros" on page 5 for information about value macros.

%LSAF_EXPORTSTUDYDETAILS Macro

Exports the metadata for the study details. The metadata is stored in a SAS data set that is located in the workspace or the repository.

Category: StudyService Module

Note: This macro sets the standard macro variables and _lsafexportstudydetailspath_, which is the full path of the details data set.

Syntax

%LSAF_EXPORTSTUDYDETAILS(LSAF_STUDYID=*study-id*, LSAF_LOCATION=WORKSPACE | REPOSITORY, LSAF_PATH=*path*<, LSAF_OVERWRITE=0 | 1><, LSAF_ENABLEVERSIONING=0 | 1><, LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM><, LSAF_CUSTOMVERSION=*version*><, LSAF_COMMENT=*comment*>);

Required Arguments

LSAF_STUDYID=*study-id*
The identifier of the study. To get the identifier, call %LSAF_GETSTUDYID.

LSAF_LOCATION=WORKSPACE | REPOSITORY
The case-insensitive output location for the exported file.

LSAF_PATH=*path*
The case-sensitive output path for the exported files.

Optional Arguments

LSAF_OVERWRITE=0 (Default) | 1
Indicates whether existing unversioned data sets are overwritten. See “LSAF_OVERWRITE=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_ENABLEVERSIONING=0 (Default) | 1
Indicates whether versioning is enabled for new exported files. See “LSAF_ENABLEVERSIONING=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM
The case-insensitive type for a versioned file in the repository. See “LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM” on page 6 for more information about this argument.

LSAF_CUSTOMVERSION=*version*
If LSAF_VERSIONTYPE is CUSTOM, the specified version value. See “LSAF_CUSTOMVERSION=version” on page 6 for more information about this argument.

LSAF_COMMENT=*comment*
The check-in comment to associate with the action of adding the data set to the repository. If LSAF_LOCATION is WORKSPACE, this argument is ignored.

Details

Sets the value of the macro variable _lsafexportstudydetailspath_ to the full path of the details data set, which includes the file extension. See “Value Macros” on page 5 for information about value macros.

%LSAF_EXPORTSTUDYDICTS Macro

Exports the metadata for the study controlled terminology dictionaries. The metadata is stored in a SAS data set in the workspace or repository.

Category: StudyService Module

Note: This macro sets the standard macro variables and _lsafexportstudydictspath_, which is the full path of the dictionaries data set.

Syntax

%LSAF_EXPORTSTUDYDICTS(LSAF_STUDYID=*study-id*, LSAF_LOCATION=WORKSPACE | REPOSITORY, LSAF_PATH=*path*<, LSAF_OVERWRITE=0 | 1><, LSAF_ENABLEVERSIONING=0 | 1><, LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM><, LSAF_CUSTOMVERSION=*version*><, LSAF_COMMENT=*comment*>);

Required Arguments

LSAF_STUDYID=*study-id*
The identifier of the study. To get the identifier, call %LSAF_GETSTUDYID.

LSAF_LOCATION=WORKSPACE | REPOSITORY
The case-insensitive output location for the exported file.

LSAF_PATH=*path*
The case-sensitive output path for the exported files.

Optional Arguments

LSAF_OVERWRITE=0 (Default) | 1
Indicates whether existing unversioned data sets are overwritten. See “LSAF_OVERWRITE=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_ENABLEVERSIONING=0 (Default) | 1
Indicates whether versioning is enabled for new exported files. See “LSAF_ENABLEVERSIONING=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM
The case-insensitive type for a versioned file in the repository. See “LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM” on page 6 for more information about this argument.

LSAF_CUSTOMVERSION=*version*
If LSAF_VERSIONTYPE is CUSTOM, the specified version value. See “LSAF_CUSTOMVERSION=version” on page 6 for more information about this argument.

LSAF_COMMENT=*comment*
The check-in comment to associate with the action of adding the data set to the repository. If LSAF_LOCATION is WORKSPACE, this argument is ignored.

Details

Sets the value of the macro variable _lsafexportstudydictspath_ to the full path of the dictionaries data set, which includes the file extension. See “Value Macros” on page 5 for information about value macros.

%LSAF_EXPORTSTUDYDOCS Macro

Exports the metadata for the study supporting documents. The metadata is stored in a SAS data set in the workspace or repository.

Category: StudyService Module

Note: This macro sets the standard macro variables and _lsafexportstudydocspath_, which is the full path of the documents data set.

Syntax

%LSAF_EXPORTSTUDYDOCS(LSAF_STUDYID=*study-id*, LSAF_BASESTANDARDNAME=*name*, LSAF_MODELID=*model-id*, LSAF_LOCATION=WORKSPACE | REPOSITORY, LSAF_PATH=*path*<, LSAF_OVERWRITE=0 | 1><, LSAF_ENABLEVERSIONING=0 | 1><, LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM><, LSAF_CUSTOMVERSION=*version*><, LSAF_COMMENT=*comment*>);

Required Arguments

LSAF_STUDYID=*study-id*
The identifier of the study. To get the identifier, call %LSAF_GETSTUDYID.

LSAF_BASESTANDARDNAME=*name*
The case-sensitive name of the base standard, such as SDTM and ADaM.

LSAF_MODELID=*model-id*
The identifier of the model. To get the identifier, call %LSAF_GETSTUDYSTANDARDS.

LSAF_LOCATION=WORKSPACE | REPOSITORY
The case-insensitive output location for the exported file.

LSAF_PATH=*path*
The case-sensitive output path for the exported files.

Optional Arguments

LSAF_OVERWRITE=0 (Default) | 1
Indicates whether existing unversioned data sets are overwritten. See “LSAF_OVERWRITE=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_ENABLEVERSIONING=0 (Default) | 1
Indicates whether versioning is enabled for new exported files. See “LSAF_ENABLEVERSIONING=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM
The case-insensitive type for a versioned file in the repository. See “LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM” on page 6 for more information about this argument.

LSAF_CUSTOMVERSION=*version*
If LSAF_VERSIONTYPE is CUSTOM, the specified version value. See “LSAF_CUSTOMVERSION=version” on page 6 for more information about this argument.

LSAF_COMMENT=*comment*
The check-in comment to associate with the action of adding the data set to the repository. If LSAF_LOCATION is WORKSPACE, this argument is ignored.

Details

Sets the value of the macro variable _lsafexportstudydocspath_ to the full path of the documents data set, which includes the file extension. See “Value Macros” on page 5 for information about value macros.

%LSAF_EXPORTSTUDYEXTDICTS Macro

Exports the metadata for the study external dictionaries. The metadata is stored in a SAS data set in the workspace or repository.

Category: StudyService Module

Note: This macro sets the standard macro variables and _lsafexportstudyextdictspath_, which is the full path of the external dictionaries data set.

Syntax

%LSAF_EXPORTSTUDYEXTDICTS(LSAF_STUDYID=*study-id*, LSAF_LOCATION=WORKSPACE | REPOSITORY, LSAF_PATH=*path*<, LSAF_OVERWRITE=0 | 1><, LSAF_ENABLEVERSIONING=0 | 1><, LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM><, LSAF_CUSTOMVERSION=*version*><, LSAF_COMMENT=*comment*>);

Required Arguments

LSAF_STUDYID=*study-id*
The identifier of the study. To get the identifier, call %LSAF_GETSTUDYID.

LSAF_LOCATION=WORKSPACE | REPOSITORY
The case-insensitive output location for the exported file.

LSAF_PATH=*path*
The case-sensitive output path for the exported files.

Optional Arguments

LSAF_OVERWRITE=0 (Default) | 1
Indicates whether existing unversioned data sets are overwritten. See “LSAF_OVERWRITE=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_ENABLEVERSIONING=0 (Default) | 1
Indicates whether versioning is enabled for new exported files. See “LSAF_ENABLEVERSIONING=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM
The case-insensitive type for a versioned file in the repository. See “LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM” on page 6 for more information about this argument.

LSAF_CUSTOMVERSION=*version*
If LSAF_VERSIONTYPE is CUSTOM, the specified version value. See “LSAF_CUSTOMVERSION=version” on page 6 for more information about this argument.

LSAF_COMMENT=*comment*
The check-in comment to associate with the action of adding the data set to the repository. If LSAF_LOCATION is WORKSPACE, this argument is ignored.

Details

Sets the value of the macro variable _lsafexportstudyextdictspath_ to the full path of the external dictionaries data set, which includes the file extension. See “Value Macros” on page 5 for information about value macros.

%LSAF_EXPORTSTUDYSTANDARDS Macro

Exports the metadata for the study standards. The metadata is stored in a SAS data set in the workspace or repository.

Category: StudyService Module

Note: This macro sets the standard macro variables and _lsafexportstudystandardspath_, which is the full path of the standards data set.

Syntax

%LSAF_EXPORTSTUDYSTANDARDS(LSAF_STUDYID=*study-id*, LSAF_LOCATION=WORKSPACE | REPOSITORY, LSAF_PATH=*path*<, LSAF_OVERWRITE=0 | 1><, LSAF_ENABLEVERSIONING=0 | 1><, LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM><, LSAF_CUSTOMVERSION=*version*><, LSAF_COMMENT=*comment*>);

Required Arguments

LSAF_STUDYID=*study-id*
The identifier of the study. To get the identifier, call %LSAF_GETSTUDYID.

LSAF_LOCATION=WORKSPACE | REPOSITORY
The case-insensitive output location for the exported file.

LSAF_PATH=*path*
The case-sensitive output path for the exported files.

Optional Arguments

LSAF_OVERWRITE=0 (Default) | 1
Indicates whether existing unversioned data sets are overwritten. See “LSAF_OVERWRITE=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_ENABLEVERSIONING=0 (Default) | 1
Indicates whether versioning is enabled for new exported files. See “LSAF_ENABLEVERSIONING=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM
The case-insensitive type for a versioned file in the repository. See “LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM” on page 6 for more information about this argument.

LSAF_CUSTOMVERSION=*version*
If LSAF_VERSIONTYPE is CUSTOM, the specified version value. See “LSAF_CUSTOMVERSION=version” on page 6 for more information about this argument.

LSAF_COMMENT=*comment*
The check-in comment to associate with the action of adding the data set to the repository. If LSAF_LOCATION is WORKSPACE, this argument is ignored.

Details

Sets the value of the macro variable _lsafexportstudystandardspath_ to the full path of the standards data set, which includes the file extension. See “Value Macros” on page 5 for information about value macros.

%LSAF_EXPORTSTUDYTABLES Macro

Exports the metadata for the study tables to the specified area in the repository. The metadata is stored in SAS data sets. The paths of the exported files are stored in a SAS data set.

Category: StudyService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_EXPORTSTUDYTABLES(LSAF_STUDYID=*study-id*, LSAF_BASESTANDARDNAME=*name*, LSAF_MODELID=*model-id*, LSAF_PATH=*path*, LSAF_PREFIX=*prefix*<, LSAF_OVERWRITE=0 | 1><, LSAF_ENABLEVERSIONING=0 | 1><, LSAF_VERSIONTYPEFORNEWFILES=MAJOR | MINOR | CUSTOM><, LSAF_CUSTOMVERSFORNEWFILES=*version*><, LSAF_VERSIONTYPEFOREXISTINGFILES=MAJOR | MINOR | CUSTOM><, LSAF_CUSTOMVERSFOREXISTINGFILES=*version*><, LSAF_COMMENT=*comment*><, SAS_RESULT_DSNAME=*name*>);

Required Arguments

LSAF_STUDYID=*study-id*
The identifier of the study. To get the identifier, call %LSAF_GETSTUDYID.

LSAF_BASESTANDARDNAME=*name*
The case-sensitive name of the base standard, such as SDTM and ADaM.

LSAF_MODELID=*model-id*
The identifier of the model. This can be retrieved calling %LSAF_GETSTUDYSTANDARDS.

LSAF_PATH=*path*
The case-sensitive output path for the exported files.

LSAF_PREFIX=*prefix*
The lowercase prefix of the SAS data sets that are created for the exported tables.

Optional Arguments

LSAF_OVERWRITE=0 (Default) | 1
Indicates whether existing unversioned data sets are overwritten. See “LSAF_OVERWRITE=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_ENABLEVERSIONING=0 (Default) | 1
Indicates whether versioning is enabled for new exported files. See “LSAF_ENABLEVERSIONING=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_VERSIONTYPEFORNEWFILES=MAJOR | MINOR | CUSTOM
If enableversioning is True, the version type to use to create a file. See "LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM" on page 6 for more information about this argument.

LSAF_CUSTOMVERSFORNEWFILES=*version*
The version number to use to create a customized versioned file. See "LSAF_CUSTOMVERSION=version" on page 6 for more information about this argument.

LSAF_VERSIONTYPEFOREXISTINGFILES=MAJOR | MINOR | CUSTOM
The version type for an existing file. See "LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM" on page 6 for more information about this argument.

LSAF_CUSTOMVERSFOREXISTINGFILES=*version*
The version number to use for an existing customized versioned file. See "LSAF_CUSTOMVERSION=version" on page 6 for more information about this argument.

LSAF_COMMENT=*comment*
The check-in comment to associate with the action of adding the data set to the repository. If LSAF_LOCATION is WORKSPACE, this argument is ignored.

SAS_RESULT_DSNAME=*name*
The name of the output data set that contains the path, name, and location of the exported table SAS data sets. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFEXPORTSTUDYTABLES.

The data set contains a row for each export data set, sorted by path, and the following columns:

Column Name	Description
location	The location of the exported file (REPOSITORY).
name	The name of the exported file.
path	The path of the exported file.

%LSAF_EXPORTSTUDYTABLESTOWS Macro

Exports the metadata for the study tables to a location in the workspace. The metadata is stored in SAS data sets. The paths of the exported files are stored in a SAS data set.

Category: StudyService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_EXPORTSTUDYTABLESTOWS(LSAF_STUDYID=*study-id*, LSAF_BASESTANDARDNAME=*name*, LSAF_MODELID=*model-id*, LSAF_PATH=*path*, LSAF_PREFIX=*prefix*<, LSAF_OVERWRITE=0 | 1><, SAS_RESULT_DSNAME=*name*>);

Required Arguments

LSAF_STUDYID=*study-id*
The identifier of the study. To get the identifier, call %LSAF_GETSTUDYID.

LSAF_BASESTANDARDNAME=*name*
The case-sensitive name of the base standard, such as SDTM and ADaM.

LSAF_MODELID=*model-id*
The identifier of the model. To get the identifier, call %LSAF_GETSTUDYSTANDARDS.

LSAF_PATH=*path*
The case-sensitive output path for the exported files.

LSAF_PREFIX=*prefix*
The lowercase prefix of the SAS data sets that are created for the exported tables.

Optional Arguments

LSAF_OVERWRITE=0 (Default) | 1
Indicates whether existing unversioned data sets are overwritten. See “LSAF_OVERWRITE=0 (Default) | 1” on page 6 for more information about this argument.

SAS_RESULT_DSNAME=*name*
The name of the output data set that contains the path, name, and location of the exported table SAS data sets. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFEXPORTSTUDYTABLESTOWS.

The data set contains a row for each exported data set, sorted by path, and the following columns:

Column Name	Description
location	The location of the exported file (WORKSPACE).
name	The name of the exported file.
path	The path of the exported file.

%LSAF_EXPORTSTUDYTLFS Macro

Exports the metadata for the study tables, listings, and figures to the specified area in the repository. The metadata is stored in SAS data sets. The paths of the exported files are stored in a SAS data set.

Category: StudyService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_EXPORTSTUDYTLFS(LSAF_STUDYID=*study-id*,
LSAF_BASESTANDARDNAME=*name*, LSAF_MODELID=*model-id*,
LSAF_PATH=*path*, LSAF_PREFIX=*prefix*<, LSAF_OVERWRITE=0 | 1><,
LSAF_INCLUDERESULTSANDDATASETS=0 | 1><,
LSAF_ENABLEVERSIONING=0 | 1><,
LSAF_VERSIONTYPEFORNEWFILES=MAJOR | MINOR | CUSTOM><,
LSAF_CUSTOMVERSFORNEWFILES=*version*><,
LSAF_VERSIONTYPEFOREXISTINGFILES=MAJOR | MINOR | CUSTOM><,
LSAF_CUSTOMVERSFOREXISTINGFILES=*version*><,
LSAF_COMMENT=*comment*><, SAS_RESULT_DSNAME=*name*>);

Required Arguments

LSAF_STUDYID=*study-id*
The identifier of the study. To get the identifier, call %LSAF_GETSTUDYID.

LSAF_BASESTANDARDNAME=*name*
The case-sensitive name of the base standard, such as SDTM and ADaM.

LSAF_MODELID=*model-id*
The identifier of the model. To get the identifier, call %LSAF_GETSTUDYSTANDARDS.

LSAF_PATH=*path*
The case-sensitive output path for the exported files.

LSAF_PREFIX=*prefix*
The lowercase prefix of the SAS data sets that are created for the exported tables.

Optional Arguments

LSAF_OVERWRITE=0 (Default) | 1
Indicates whether existing unversioned data sets are overwritten. See “LSAF_OVERWRITE=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_INCLUDERESULTSANDDATASETS=0 | 1 (Default)
Indicates whether to include the results data sets. This argument is relevant only when model=Analysis, otherwise, the flag is ignored.

LSAF_ENABLEVERSIONING=0 (Default) | 1
Indicates whether versioning is enabled for new exported files. See “LSAF_ENABLEVERSIONING=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_VERSIONTYPEFORNEWFILES=MAJOR | MINOR | CUSTOM
If enableversioning is True, the version type to use to create a file. See “LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM” on page 6 for more information about this argument.

LSAF_CUSTOMVERSFORNEWFILES=*version*
The version number to use to create a customized versioned file. See “LSAF_CUSTOMVERSION=version” on page 6 for more information about this argument.

LSAF_VERSIONTYPEFOREXISTINGFILES=MAJOR | MINOR | CUSTOM
The version type for an existing file. See “LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM” on page 6 for more information about this argument.

LSAF_CUSTOMVERSFOREXISTINGFILES=*version*
The version number to use for an existing customized versioned file. See “LSAF_CUSTOMVERSION=version” on page 6 for more information about this argument.

LSAF_COMMENT=*comment*
The check-in comment to associate with the action of adding the data set to the repository. If LSAF_LOCATION is WORKSPACE, this argument is ignored.

SAS_RESULT_DSNAME=*name*
The name of the output data set that contains the path, name, and location of the exported tables, listings, and figures SAS data sets. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFEXPORTSTUDYTLFS.

The data set contains a row for each exported data set, sorted by path, and the following columns:

Column Name	Description
location	The location of the exported file (REPOSITORY).
name	The name of the exported file.
path	The path of the exported file.

%LSAF_EXPORTSTUDYTLFSTOWORKSPACE Macro

Exports the metadata for the study tables, listings, and figures to the specified area in the workspace. The metadata is stored in SAS data sets. The paths of the exported files are stored in a SAS data set.

Category: StudyService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_EXPORTSTUDYTLFSTOWORKSPACE(LSAF_STUDYID=*study-id*, LSAF_BASESTANDARDNAME=*name*, LSAF_MODELID=*model-id*, LSAF_PATH=*path*, LSAF_PREFIX=*prefix*<, LSAF_OVERWRITE=0 | 1><, LSAF_INCLUDERESULTSANDDATASETS=0 | 1><, SAS_RESULT_DSNAME=*name*>);

Required Arguments

LSAF_STUDYID=*study-id*
The identifier of the study. To get the identifier, call %LSAF_GETSTUDYID.

LSAF_BASESTANDARDNAME=*name*
The case-sensitive name of the base standard, such as SDTM and ADaM.

LSAF_MODELID=*model-id*
The identifier of the model. To get the identifier, call %LSAF_GETSTUDYSTANDARDS.

LSAF_PATH=*path*
The case-sensitive output path for the exported files.

LSAF_PREFIX=*prefix*
The lowercase prefix of the SAS data sets that are created for the exported tables.

Optional Arguments

LSAF_OVERWRITE=0 (Default) | 1
Indicates whether existing unversioned data sets are overwritten. See “LSAF_OVERWRITE=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_INCLUDERESULTSANDDATASETS=0 | 1 (Default)
Indicates whether to include the results data sets. This argument is relevant only when model=Analysis, otherwise, the flag is ignored.

SAS_RESULT_DSNAME=*name*
The name of the output data set that contains the path, name, and location of the exported tables, listings, and figures SAS data sets. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFEXPORTSTUDYTLFSTOWORKSPACE.

The data set contains a row for each exported data set, sorted by path, and the following columns:

Column Name	Description
location	The location of the exported file (WORKSPACE).

Column Name	Description
name	The name of the exported file.
path	The path of the exported file.

%LSAF_EXPORTSTUDYVLM Macro

Exports the metadata for the study value level metadata (VLM). The metadata is stored in a SAS data set in the workspace or repository.

Category: StudyService Module

Note: This macro sets the standard macro variables and _lsafexportstudyvlmpath_, which is the full path of the VLM data set.

Syntax

%LSAF_EXPORTSTUDYVLM(LSAF_STUDYID=*study-id*, LSAF_BASESTANDARDNAME=*name*, LSAF_MODELID=*model-id*, LSAF_LOCATION=WORKSPACE | REPOSITORY, LSAF_PATH=*path*<, LSAF_OVERWRITE=0 | 1><, LSAF_ENABLEVERSIONING=0 | 1><, LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM><, LSAF_CUSTOMVERSION=*version*><, LSAF_COMMENT=*comment*>);

Required Arguments

LSAF_STUDYID=*study-id*
The identifier of the study. To get the identifier, call %LSAF_GETSTUDYID.

LSAF_BASESTANDARDNAME=*name*
The case-sensitive name of the base standard, such as SDTM and ADaM.

LSAF_MODELID=*model-id*
The identifier of the model. To get the identifier, call %LSAF_GETSTUDYSTANDARDS.

LSAF_LOCATION=WORKSPACE | REPOSITORY
The case-insensitive output location for the exported file.

LSAF_PATH=*path*
The case-sensitive output path for the exported files.

Optional Arguments

LSAF_OVERWRITE=0 (Default) | 1
Indicates whether existing unversioned data sets are overwritten. See “LSAF_OVERWRITE=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_ENABLEVERSIONING=0 (Default) | 1
Indicates whether versioning is enabled for new exported files. See “LSAF_ENABLEVERSIONING=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM
The case-insensitive type for a versioned file in the repository. See “LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM” on page 6 for more information about this argument.

LSAF_CUSTOMVERSION=*version*
If LSAF_VERSIONTYPE is CUSTOM, the specified version value. See “LSAF_CUSTOMVERSION=version” on page 6 for more information about this argument.

LSAF_COMMENT=*comment*
The check-in comment to associate with the action of adding the data set to the repository. If LSAF_LOCATION is WORKSPACE, this argument is ignored.

Details

Sets the value of the macro variable _lsafexportstudyvlmpath_ to the full path of the VLM data set, which includes the file extension. See “Value Macros” on page 5 for information about value macros.

%LSAF_GETDICTIONARYSTUDIES Macro

Gets the metadata for all of the studies that are associated with a dictionary. The metadata is stored in a SAS data set.

Category: StudyService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETDICTIONARYSTUDIES(LSAF_DICTIONARYID=*dictionary-id*<, SAS_DSNAME=*name*>);

Required Argument

LSAF_DICTIONARYID=*dictionary-id*
The identifier of the dictionary. To get the identifier, call %LSAF_GETDICTIONARYID.

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the studies. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETDICTIONARYSTUDIES.

The data set contains a row for each study, sorted by the study identifier, and the following columns:

Column Name	Description
id	The unique study identifier.
name	The study name.
path	The path of the study.
typeId	The context type identifier.

%LSAF_GETEXTDICTIONARYSTUDIES Macro

Gets the metadata for all of the studies that are associated with an external dictionary. The metadata is stored in a SAS data set.

Category: StudyService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETEXTDICTIONARYSTUDIES(LSAF_EXTDICTIONARYID=*external-dictionary-id*<, SAS_DSNAME=*name*>);

Required Argument

LSAF_EXTDICTIONARYID=*external-dictionary-id*
The identifier of the external dictionary. To get the identifier, call %LSAF_GETEXTDICTIONARYID.

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the studies. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETEXTDICTIONARYSTUDIES.

The data set contains a row for each study, sorted by the study name, and the following columns:

Column Name	Description
id	The unique study identifier.
name	The study name.
path	The path of the study.
typeId	The context type identifier.

%LSAF_GETSTANDARDSTUDIES Macro

Gets the metadata for all of the studies that are associated with a data standard. The metadata is stored in a SAS data set.

Category: StudyService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETSTANDARDSTUDIES(LSAF_STANDARDID=*standard-id*<, SAS_DSNAME=*name*>);

Required Argument

LSAF_STANDARDID=*standard-id*
The identifier of the standard. To get the identifier, call %LSAF_GETSTANDARDID.

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the studies. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETSTANDARDSTUDIES.

The data set contains a row for each study, sorted by the study identifier, and the following columns:

Column Name	Description
id	The unique study identifier.
name	The study name.
path	The path of the study.
typeId	The context type identifier.

%LSAF_GETSTDYCOMPONENTEDITEDBY Macro

Gets the name of the user who is the current editor for a study component.

Category: StudyService Module

Note: This macro sets the standard macro variables and _lsafStudyComponentEditedBy_, which is the user identifier of the editor for the study component.

Syntax

%LSAF_GETSTDYCOMPONENTEDITEDBY((LSAF_STUDYID=*study-id*, LSAF_STUDYCOMPONENT=*component*);

Required Arguments

LSAF_STUDYID=*study-id*
The identifier of the study. To get the identifier, call %LSAF_GETSTUDYID.

LSAF_STUDYCOMPONENT=*component*
The study component. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=StudyComponent`.

Details

Sets the value of the macro variable _lsafStudyComponentEditedBy_ to the user identifier of the editor for the study component. See “Value Macros” on page 5 for information about value macros.

%LSAF_GETSTUDYDICTIONARIES Macro

Gets the metadata for all of the controlled terminology dictionaries that are associated with a study. The metadata is stored in a SAS data set.

Category: StudyService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETSTUDYDICTIONARIES(LSAF_STUDYID=*study-id*<, SAS_DSNAME=*name*>);

Required Argument

LSAF_STUDYID=*study-id*
The identifier of the study. To get the identifier, call %LSAF_GETSTUDYID.

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the controlled terminology dictionaries. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETSTUDYDICTIONARIES.

The data set contains a row for each dictionary in the order that they are associated with the study, and columns with the following names:

Column Name	Description
studyId	The study identifier.
name	The dictionary name.
id	The unique dictionary identifier.
standard	The associated data standard.
description	The dictionary description.
isCheckedOut	Indicates whether the dictionary is checked out.

Column Name	Description
checkedOutBy	If the dictionary is checked out, the identifier of the user who has it checked out.
checkedOutDate	If the dictionary is checked out, the date on which it was checked out.
checkedOutDateSASFormat	If the dictionary is checked out, the date on which it was checked out (in SAS format).
createdDate	The date on which the dictionary was created.
createdDateSASFormat	The date on which the dictionary was created (in SAS format).
createdBy	The identifier of the user who created the dictionary.
state	The state of the dictionary.
publishedState	The published state of the dictionary.
currentVersion	The current published version of the dictionary.
associatedStudiesCount	The number of studies that are associated with the dictionary.
lastModifiedDate	The date on which the dictionary was last modified.
lastModifiedDateSASFormat	The date on which the dictionary was last modified (in SAS format).
lastModifiedBy	The identifier of the user of who last modified the dictionary.
releaseDate	The date on which the dictionary was released.
dictionaryStatus	The status of the dictionary.

Note: See “Format Date Values” on page 4 for information about date formats.

%LSAF_GETSTUDYEXTDICTIONARIES Macro

Gets the metadata for all of the external dictionaries that are associated with a study. The metadata is stored in a SAS data set.

Category: StudyService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETSTUDYEXTDICTIONARIES(LSAF_STUDYID=*study-id*<, SAS_DSNAME=*name*>);

Required Argument

LSAF_STUDYID=*study-id*
The identifier of the study. To get the identifier, call %LSAF_GETSTUDYID.

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the external dictionaries. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETSTUDYEXTDICTIONARIES.

The data set contains a row for each dictionary, sorted by the dictionary name, and the following columns:

Column Name	Description
studyId	The study identifier.
id	The unique external dictionary identifier.
name	The external dictionary name.
description	The external dictionary description.
dataType	The external dictionary data type. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=DataType` and `source=core`.

Column Name	Description
shortName	The short name of the external dictionary.
dictionary	The external dictionary.
version	The current published version of the external dictionary.
href	The HTTP reference for the external dictionary.
createdDate	The date on which the external dictionary was created.
createdDateSASFormat	The date on which the external dictionary was created (in SAS format).
createdBy	The identifier of the user who created the external dictionary.
associatedStudiesCount	The number of studies that are associated with the external dictionary.
lastModifiedDate	The date on which the external dictionary was last modified.
lastModifiedDateSASFormat	The date on which the external dictionary was last modified (in SAS format).
lastModifiedBy	The identifier of the user of who last modified the external dictionary.

Note: See “Format Date Values” on page 4 for information about date formats.

%LSAF_GETSTUDYID Macro

Gets the identifier for a study. You must have Read access permission to the study.

Category: StudyService Module

Note: This macro sets the standard macro variables and _lsafStudyId_, which is the study identifier.

Syntax

%LSAF_GETSTUDYID(LSAF_PATH=*path*);

Required Argument

LSAF_PATH=*path*
The case-sensitive path of the study.

Details

Sets the value of the macro variable _lsafStudyId_ to the study identifier. See “Value Macros” on page 5 for information about value macros.

You must have Read access permission to the study.

%LSAF_GETSTUDYSTANDARDS Macro

Gets the metadata for all of the standards that are associated with a study. The metadata is stored in a SAS data set.

Category: StudyService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETSTUDYSTANDARDS(LSAF_STUDYID=*study-id*<, SAS_DSNAME=*name*>);

Required Argument

LSAF_STUDYID=*study-id*
The identifier of the study. To get the identifier, call %LSAF_GETSTUDYID.

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the standards. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETSTUDYSTANDARDS.

The data set contains a row for each standard, sorted by the standard name, and the following columns:

Column Name	Description
studyId	The study identifier.
id	The unique standard identifier.
name	The data standard name.
baseStandardType	The base data standard type.
baseStandardName	The base data standard name.
baseStandardVersion	The base data standard version.
modelId	The model identifier.
description	The data standard description.
isCheckedOut	Indicates whether the data standard is checked out.
checkedOutBy	If the data standard is checked out, the identifier of the user who has it checked out.
checkedOutDate	If the data standard is checked out, the date on which it was checked out.
checkedOutDateSASFormat	If the data standard is checked out, the date on which it was checked out (in SAS format).
createdDate	The date on which the data standard was created.
createdDateSASFormat	The date on which the data standard was created (in SAS format).
createdBy	The identifier of the user who created the data standard.
state	The state of the data standard.
publishedState	The published state of the data standard.
currentVersion	The current published version of the data standard.
associatedStudiesCount	The number of studies that are associated with the data standard.

Column Name	Description
lastModifiedDate	The date on which the data standard was last modified.
lastModifiedDateSASFormat	The date on which the data standard was last modified (in SAS format).
lastModifiedBy	The identifier of the user who last modified the data standard.
standardStatus	The status of the data standard.

Note: See "Format Date Values" on page 4 for information about date formats.

%LSAF_IMPORTSTUDYCODELISTS Macro

Imports study codelists from the specified data set, and can create a validation report as a CSV file in the specified location.

Category: StudyService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_IMPORTSTUDYCODELISTS(LSAF_STUDYID=*study-id*, LSAF_IMPORTACTION=*update-action*, LSAF_LOCATION=WORKSPACE | REPOSITORY, LSAF_CODELISTSPATH=*path*<, LSAF_CODELISTSVERSION=*version*><, LSAF_REPORTLOCATION=REPOSITORY | WORKSPACE><, LSAF_REPORTPATH=*path*><, LSAF_REPORTOVERWRITE=0 | 1><, LSAF_REPORTENABLEVERSIONING=0 | 1><, LSAF_REPORTVERSIONTYPE=MAJOR | MINOR | CUSTOM><, LSAF_REPORTCUSTOMVERSION=*version*><, LSAF_REPORTCOMMENT=*comment*><, SAS_RESULT_DSNAME=*name*>);

Required Arguments

LSAF_STUDYID=*study-id*
The identifier of the study. To get the identifier, call %LSAF_GETSTUDYID.

LSAF_IMPORTACTION=*update-action*
The case-insensitive value to indicate the action to take by the import. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=UpdateAction`.

LSAF_LOCATION=WORKSPACE | REPOSITORY
The location of the import source file.

LSAF_CODELISTSPATH=*path*
The case-sensitive path of the data set to import code lists.

Optional Arguments

LSAF_CODELISTSVERSION=*version*
The version of the data set to import code lists. If not specified, the most recent version is used.

LSAF_REPORTLOCATION=REPOSITORY | WORKSPACE
The case-insensitive location in which to create the report. If the report location and report path are not specified, the import runs but no validation report is created.

LSAF_REPORTPATH=*path*
The case-sensitive path of the report, which includes the report file name. If the report location and report path are not specified, the import runs but no validation report is created.

LSAF_REPORTOVERWRITE=0 (Default) | 1
Indicates whether an existing unversioned repository or existing workspace report file is overwritten by a new report with a same path. See “LSAF_OVERWRITE=0 (Default) | 1” on page 6 for more information about the options for this argument.

LSAF_REPORTENABLEVERSIONING=0 (Default) | 1
Indicates whether to enable versioning for a new file in the repository. See “LSAF_ENABLEVERSIONING=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_REPORTVERSIONTYPE=MAJOR | MINOR | CUSTOM
The case-insensitive type for a versioned report in the repository. See “LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM” on page 6 for more information about this argument.

LSAF_REPORTCUSTOMVERSION=*version*
If LSAF_REPORTVERSIONTYPE is CUSTOM, the custom version value. See “LSAF_CUSTOMVERSION=version” on page 6 for more information about this argument.

LSAF_REPORTCOMMENT=*comment*
The comment to associate with the new repository file.

SAS_RESULT_DSNAME=*name*
The name of the output data set that contains the completion status information. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFIMPORTSTUDYCODELISTS.

The data set contains information that pertains to the import operation using the following columns:

Column Name	Description
studyId	The study identifier.
name	The name of the property.

Column Name	Description
value	The value of the property.

Details

Imports study codelists from the specified data set.

You can choose to create a validation report as a CSV file in the specified location.

Failure and Message Handling

The macro performs a series of preliminary validation procedures to determine whether the specified parameters are valid and required parameters are specified. If none of these procedures fail, the import is attempted. The success or failure, and any issues that are encountered during the import, are reported. For more information, see Table 1.4 on page 9.

%LSAF_IMPORTSTUDYDETAILS Macro

Imports study details from the specified SAS data set.

Category: StudyService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_IMPORTSTUDYDETAILS(LSAF_STUDYID=*study-id*, LSAF_LOCATION=WORKSPACE | REPOSITORY, LSAF_DETAILSPATH=*path*<, LSAF_DETAILSVERSION=*version*>);

Required Arguments

LSAF_STUDYID=*study-id*
The identifier of the study. To get the identifier, call %LSAF_GETSTUDYID.

LSAF_LOCATION=WORKSPACE | REPOSITORY
The location of the import source file.

LSAF_DETAILSPATH=*path*
The case-sensitive path of the data set to import the study details.

Optional Argument

LSAF_DETAILSVERSION=*version*
The version of the data set to import the details metadata. If not specified, the most recent version is used.

%LSAF_IMPORTSTUDYDOCS Macro

Imports study supporting documents from the specified data set and can create a validation report as a CSV file in the specified location.

Category: StudyService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_IMPORTSTUDYDOCS(LSAF_STUDYID=*study-id*, LSAF_BASESTANDARDNAME=*name*, LSAF_MODELID=*model-id*, LSAF_IMPORTACTION=*update-action*, LSAF_LOCATION=WORKSPACE | REPOSITORY, LSAF_DOCSPATH=*path*<, LSAF_DOCSVERSION=*version*><, LSAF_REPORTLOCATION=REPOSITORY | WORKSPACE><, LSAF_REPORTPATH=*path*><, LSAF_REPORTOVERWRITE=0 | 1><, LSAF_REPORTENABLEVERSIONING=0 | 1><, LSAF_REPORTVERSIONTYPE=MAJOR | MINOR | CUSTOM><, LSAF_REPORTCUSTOMVERSION=*version*><, LSAF_REPORTCOMMENT=*comment*><, SAS_RESULT_DSNAME=*name*>);

Required Arguments

LSAF_STUDYID=*study-id*
The identifier of the study. To get the identifier, call %LSAF_GETSTUDYID.

LSAF_BASESTANDARDNAME=*name*
The case-sensitive name of the base standard, such as SDTM and ADaM.

LSAF_MODELID=*model-id*
The identifier of the model. To get the identifier, call %LSAF_GETSTUDYSTANDARDS.

LSAF_IMPORTACTION=*update-action*
The case-insensitive value to indicate the action to take by the import. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=UpdateAction`.

LSAF_LOCATION=WORKSPACE | REPOSITORY
The location of the import source file.

LSAF_DOCSPATH=*path*
The case-sensitive path of the data set to import supporting documents.

Optional Arguments

LSAF_DOCSVERSION=*version*
The version of the data set to import the supporting documents. If not specified, the most recent version is used.

LSAF_REPORTLOCATION=REPOSITORY | WORKSPACE
The case-insensitive location in which to create the report. If the report location and report path are not specified, the import runs but no validation report is created.

LSAF_REPORTPATH=*path*
The case-sensitive path of the report, which includes the report file name. If the report location and report path are not specified, the import runs but no validation report is created.

LSAF_REPORTOVERWRITE=0 (Default) | 1
Indicates whether an existing unversioned repository or existing workspace report file is overwritten by a new report with a same path. See “LSAF_OVERWRITE=0 (Default) | 1” on page 6 for more information about the options for this argument.

LSAF_REPORTENABLEVERSIONING=0 (Default) | 1
Indicates whether to enable versioning for a new file in the repository. See “LSAF_ENABLEVERSIONING=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_REPORTVERSIONTYPE=MAJOR | MINOR | CUSTOM
The case-insensitive type for a versioned report in the repository. See “LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM” on page 6 for more information about this argument.

LSAF_REPORTCUSTOMVERSION=*version*
If LSAF_REPORTVERSIONTYPE is CUSTOM, the custom version value. See “LSAF_CUSTOMVERSION=version” on page 6 for more information about this argument.

LSAF_REPORTCOMMENT=*comment*
The comment to associate with the new repository file.

SAS_RESULT_DSNAME=*name*
The name of the output data set that contains the completion status information. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFIMPORTSTUDYCODELISTS.

The data set contains information that pertains to the import operation using the following columns:

Column Name	Description
studyId	The study identifier.
name	The name of the property.
value	The value of the property.

Details

Imports study supporting documents from the specified data set and can create a validation report as a CSV file in the specified location.

Failure and Message Handling

The macro performs a series of preliminary validation procedures to determine whether the specified parameters are valid and required parameters are specified. If none of these procedures fail, the import is attempted. The success or failure, and any issues that are encountered during the import, are reported. For more information, see Table 1.4 on page 9.

%LSAF_IMPORTSTUDYFROMDEFINEXML Macro

Imports metadata from a Define-XML file and updates study components based on the contents of the file. It can create a validation report as a CSV file in the specified location.

Category: StudyService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_IMPORTSTUDYFROMDEFINEXML(LSAF_STUDYID=*study-id*, LSAF_STANDARDID=*standard-id*, LSAF_CODELISTUPDATEACTION=*update-action*, LSAF_LOCATION=WORKSPACE | REPOSITORY, LSAF_DEFINEPATH=*path*<, LSAF_DEFINEVERSION=*version*><, LSAF_REPORTLOCATION=REPOSITORY | WORKSPACE><, LSAF_REPORTPATH=*path*><, LSAF_REPORTOVERWRITE=0 | 1><, LSAF_REPORTENABLEVERSIONING=0 | 1><, LSAF_REPORTVERSIONTYPE=MAJOR | MINOR | CUSTOM><, LSAF_REPORTCUSTOMVERSION=*version*><, LSAF_REPORTCOMMENT=*comment*><, SAS_RESULT_DSNAME=*name*>);

Required Arguments

LSAF_STUDYID=*study-id*
The identifier of the study. To get the identifier, call %LSAF_GETSTUDYID.

LSAF_STANDARDID=*standard-id*
The identifier of the data standard. To get the identifier, call %LSAF_GETSTANDARDID.

LSAF_CODELISTUPDATEACTION=*update-action*
The case-insensitive value to indicate the action to take to update the code lists in the study. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=UpdateAction`.

LSAF_LOCATION=WORKSPACE | REPOSITORY
The location of the import source file.

LSAF_DEFINEPATH=*path*
The case-sensitive path of the Define-XML file to import the study metadata.

Optional Arguments

LSAF_DEFINEVERSION=*version*
The version of the Define-XML to import the study metadata. If not specified, the most recent version is used.

LSAF_REPORTLOCATION=REPOSITORY | WORKSPACE
The case-insensitive location in which to create the report. If the report location and report path are not specified, the import runs but no validation report is created.

LSAF_REPORTPATH=*path*
The case-sensitive path of the report, which includes the report file name. If the report location and report path are not specified, the import runs but no validation report is created.

LSAF_REPORTOVERWRITE=0 (Default) | 1
Indicates whether an existing unversioned repository or existing workspace report file is overwritten by a new report with a same path. See “LSAF_OVERWRITE=0 (Default) | 1” on page 6 for more information about the options for this argument.

LSAF_REPORTENABLEVERSIONING=0 (Default) | 1
Indicates whether to enable versioning for a new file in the repository. See “LSAF_ENABLEVERSIONING=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_REPORTVERSIONTYPE=MAJOR | MINOR | CUSTOM
The case-insensitive type for a versioned report in the repository. See “LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM” on page 6 for more information about this argument.

LSAF_REPORTCUSTOMVERSION=*version*
If LSAF_REPORTVERSIONTYPE is CUSTOM the custom version value. See “LSAF_CUSTOMVERSION=version” on page 6 for more information about this argument.

LSAF_REPORTCOMMENT=*comment*
The comment to associate with the new repository file.

SAS_RESULT_DSNAME=*name*
The name of the output data set that contains the completion status information. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFIMPORTSTUDYCODELISTS.

The data set contains information that pertains to the import operation using the following columns:

Column Name	Description
studyId	The study identifier.
name	The name of the property.
value	The value of the property.

Details

Imports metadata from a Define-XML file and updates study components based on the contents of the file. It can create a validation report as a CSV file in the specified location.

Codelists for the study are updated based on the UpdateAction that is specified. All other study metadata is replaced.

For more information, see Table 1.4 on page 9.

%LSAF_IMPORTSTUDYTABLES Macro

Imports study tables from the specified data sets and can create a validation report as a CSV file in the specified location.

Category: StudyService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_IMPORTSTUDYTABLES(LSAF_STUDYID=*study-id*, LSAF_STANDARDID=*standard-id*, LSAF_IMPORTACTION=*update-action*, LSAF_LOCATION=WORKSPACE | REPOSITORY, LSAF_TABLESPATH=*path*, LSAF_COLUMNSPATH=*path*<, LSAF_TABLESVERSION=*version*><, LSAF_COLUMNSVERSION=*version*><, LSAF_REPORTLOCATION=REPOSITORY | WORKSPACE><, LSAF_REPORTPATH=*path*><, LSAF_REPORTOVERWRITE=0 | 1><, LSAF_REPORTENABLEVERSIONING=0 | 1><, LSAF_REPORTVERSIONTYPE=MAJOR | MINOR | CUSTOM><, LSAF_REPORTCUSTOMVERSION=*version*><, LSAF_REPORTCOMMENT=*comment*><, SAS_RESULT_DSNAME=*name*>);

Required Arguments

LSAF_STUDYID=*study-id*
The identifier of the study. To get the identifier, call %LSAF_GETSTUDYID.

LSAF_STANDARDID=*standard-id*
The identifier of the data standard. To get the identifier, call %LSAF_GETSTANDARDID.

LSAF_IMPORTACTION=*update-action*
The case-insensitive value to indicate the action to take by the import. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=UpdateAction`.

LSAF_LOCATION=WORKSPACE | REPOSITORY
The location of the import source file.

LSAF_TABLESPATH =*path*
The case-sensitive path of the data set to import the study tables.

LSAF_COLUMNSPATH =*path*
The case-sensitive path of the data set to import the study columns.

Optional Arguments

LSAF_TABLESVERSION=*version*
The version of the data set to import the study tables. If not specified, the most recent version is used.

LSAF_COLUMNSVERSION=*version*
The version of the data set to import the study columns. If not specified, the most recent version is used.

LSAF_REPORTLOCATION=REPOSITORY | WORKSPACE
The case-insensitive location in which to create the report. If the report location and report path are not specified, the import runs but no validation report is created.

LSAF_REPORTPATH=*path*
The case-sensitive path of the report, which includes the report file name. If the report location and report path are not specified, the import runs but no validation report is created.

LSAF_REPORTOVERWRITE=0 (Default) | 1
Indicates whether an existing unversioned repository or existing workspace report file is overwritten by a new report with a same path.See “LSAF_OVERWRITE=0 (Default) | 1” on page 6 for more information about the options for this argument.

LSAF_REPORTENABLEVERSIONING=0 (Default) | 1
Indicates whether to enable versioning for a new file in the repository. See “LSAF_ENABLEVERSIONING=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_REPORTVERSIONTYPE=MAJOR | MINOR | CUSTOM
The case-insensitive type for a versioned report in the repository. See “LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM” on page 6 for more information about this argument.

LSAF_REPORTCUSTOMVERSION=*version*
If LSAF_REPORTVERSIONTYPE is CUSTOM, the custom version value. See “LSAF_CUSTOMVERSION=version” on page 6 for more information about this argument.

LSAF_REPORTCOMMENT=*comment*
The comment to associate with the new repository file.

SAS_RESULT_DSNAME=*name*
The name of the output data set that contains the completion status information about the import. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFIMPORTSTUDYTABLES.

The data set contains information that pertains to the import operation using the following columns:

Column Name	Description
studyId	The study identifier.
name	The name of the property.
value	The value of the property.

Details

Imports study tables from the specified data sets and can create a validation report as a CSV file in the specified location.

Failure and Message Handling

The macro performs a series of preliminary validation procedures to determine whether the specified parameters are valid and required parameters are specified. If none of these procedures fail, the import is attempted. The success or failure, and any issues that are encountered during the import, are reported as follows.

Codelists for the study are updated based on the UpdateAction that is specified. All other study metadata is replaced.

For more information, see Table 1.4 on page 9.

%LSAF_IMPORTSTUDYTLFS Macro

Imports study tables, figures, and listings from the specified data sets and can create a validation report as a CSV file in the specified location.

Category: StudyService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_IMPORTSTUDYTLFS(LSAF_STUDYID=*study-id*, LSAF_BASESTANDARDNAME=*name*, LSAF_MODELID=*model-id*, LSAF_IMPORTACTION=*update-action*, LSAF_LOCATION=WORKSPACE | REPOSITORY, LSAF_SECTIONSPATH=*path*, LSAF_TLFSPATH=*path*<, LSAF_RESULTSPATH=*path*><, LSAF_DATASETSPATH=*path*><, LSAF_SECTIONSVERSION=*version*><, LSAF_TLFSVERSION=*version*><, LSAF_RESULTSVERSION=*version*><, LSAF_DATASETSVERSION=*version*><, LSAF_REPORTLOCATION=*location*><, LSAF_REPORTPATH=*path*><, LSAF_REPORTOVERWRITE=0 | 1><, LSAF_REPORTENABLEVERSIONING=0 | 1><, LSAF_REPORTVERSIONTYPE=MAJOR | MINOR | CUSTOM><, LSAF_REPORTCUSTOMVERSION=*version*><, LSAF_REPORTCOMMENT=*comment*><, SAS_RESULT_DSNAME=*name*>);

Required Arguments

LSAF_STUDYID=*study-id*
The identifier of the study. To get the identifier, call %LSAF_GETSTUDYID.

LSAF_BASESTANDARDNAME=*name*
The case-sensitive name of the base standard, such as SDTM and ADaM.

LSAF_MODELID=*model-id*
The identifier of the model. To get the identifier, call %LSAF_GETSTUDYSTANDARDS.

LSAF_IMPORTACTION=*update-action*
The case-insensitive value to indicate the action to take by the import. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=UpdateAction`.

LSAF_LOCATION=WORKSPACE | REPOSITORY
The location of the import source file.

LSAF_SECTIONSPATH=*path*
The case-sensitive path of the data set to import the TLF sections.

LSAF_TLFSPATH=*path*
The case-sensitive path of the data set to import the TLFs.

Optional Arguments

LSAF_RESULTSPATH=*path*
The case-sensitive path of the data set to import the TLF results.

LSAF_DATASETSPATH=*path*
The case-sensitive path of the data set to import the TLF data sets.

LSAF_SECTIONSVERSION=*version*
The version of the data set to import the TLF sections. If not specified, the most recent version is used.

LSAF_TLFSVERSION=*version*
The version of the data set to import the TLFs. If not specified, the most recent version is used.

LSAF_RESULTSVERSION=*version*
The version of the data set to import the TLF results. If not specified, the most recent version is used.

LSAF_DATASETSVERSION=*version*
The version of the data set to import the TLF data sets. If not specified, the most recent version is used.

LSAF_REPORTLOCATION=REPOSITORY | WORKSPACE
The case-insensitive location in which to create the report. If the report location and report path are not specified, the import runs but no validation report is created.

LSAF_REPORTPATH=*path*
The case-sensitive path of the report, which includes the report file name. If the report location and report path are not specified, the import runs but no validation report is created.

LSAF_REPORTOVERWRITE=0 (Default) | 1
Indicates whether an existing unversioned repository or existing workspace report file is overwritten by a new report with a same path. See

"LSAF_OVERWRITE=0 (Default) | 1" on page 6 for more information about the options for this argument.

LSAF_REPORTENABLEVERSIONING=0 (Default) | 1
Indicates whether to enable versioning for a new file in the repository. See "LSAF_ENABLEVERSIONING=0 (Default) | 1" on page 6 for more information about this argument.

LSAF_REPORTVERSIONTYPE=MAJOR | MINOR | CUSTOM
The case-insensitive type for a versioned report in the repository. See "LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM" on page 6 for more information about this argument.

LSAF_REPORTCUSTOMVERSION=*version*
If LSAF_REPORTVERSIONTYPE is CUSTOM, the custom version value. See "LSAF_CUSTOMVERSION=version" on page 6 for more information about this argument.

LSAF_REPORTCOMMENT=*comment*
The comment to associate with the new repository file.

SAS_RESULT_DSNAME=*name*
The name of the output data set that contains the completion status information about the import. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFIMPORTSTUDYTABLES.

The data set contains information that pertains to the import operation using the following columns:

Column Name	Description
studyId	The study identifier.
name	The name of the property.
value	The value of the property.

Details

Imports study tables, figures, and listings from the specified data sets and can create a validation report as a CSV file in the specified location.

Failure and Message Handling

The macro performs a series of preliminary validation procedures to determine whether the specified parameters are valid and required parameters are specified. If none of these procedures fail, the import is attempted. The success or failure, and any issues that are encountered during the import, are reported as follows.

Codelists for the study are updated based on the UpdateAction that is specified. All other study metadata is replaced.

For more information, see Table 1.4 on page 9.

%LSAF_IMPORTSTUDYVLM Macro

Imports study value level metadata from the specified data set and can create a validation report as a CSV file in the specified location.

Category: StudyService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_IMPORTSTUDYVLM(LSAF_STUDYID=*study-id*, LSAF_BASESTANDARDNAME=*name*, LSAF_MODELID=*model-id*, LSAF_IMPORTACTION=*update-action*, LSAF_LOCATION=WORKSPACE | REPOSITORY, LSAF_VLMPATH=*path*<, LSAF_VLMVERSION=*version*><, LSAF_REPORTLOCATION=*location*><, LSAF_REPORTPATH=*path*><, LSAF_REPORTOVERWRITE=0 | 1><, LSAF_REPORTENABLEVERSIONING=0 | 1><, LSAF_REPORTVERSIONTYPE=MAJOR | MINOR | CUSTOM><, LSAF_REPORTCUSTOMVERSION=*version*><, LSAF_REPORTCOMMENT=*comment*><, SAS_RESULT_DSNAME=*name*>);

Required Arguments

LSAF_STUDYID=*study-id*
The identifier of the study. To get the identifier, call %LSAF_GETSTUDYID.

LSAF_BASESTANDARDNAME=*name*
The case-sensitive name of the base standard, such as SDTM and ADaM.

LSAF_MODELID=*model-id*
Required - The identifier of the model. To get the identifier, call %LSAF_GETSTUDYSTANDARDS.

LSAF_IMPORTACTION=*update-action*
The case-insensitive value to indicate the action to take by the import. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=UpdateAction`.

LSAF_LOCATION=WORKSPACE | REPOSITORY
The location of the import source file.

LSAF_VLMPATH=*path*
The case-sensitive path of the data set to import the value level metadata.

Optional Arguments

LSAF_VLMVERSION=*version*
The version of the data set to import the value level metadata. If not specified, the most recent version is used.

LSAF_REPORTLOCATION=REPOSITORY | WORKSPACE
The case-insensitive location in which to create the report. If the report location and report path are not specified, the import runs but no validation report is created.

LSAF_REPORTPATH=*path*
The case-sensitive path of the report, which includes the report file name. If the report location and report path are not specified, the import runs but no validation report is created.

LSAF_REPORTOVERWRITE=0 (Default) | 1
Indicates whether an existing unversioned repository or existing workspace report file is overwritten by a new report with a same path. See “LSAF_OVERWRITE=0 (Default) | 1” on page 6 for more information about the options for this argument.

LSAF_REPORTENABLEVERSIONING=0 (Default) | 1
Indicates whether to enable versioning for a new file in the repository. See “LSAF_ENABLEVERSIONING=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_REPORTVERSIONTYPE=MAJOR | MINOR | CUSTOM
The case-insensitive type for a versioned report in the repository. See “LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM” on page 6 for more information about this argument.

LSAF_REPORTCUSTOMVERSION=*version*
If LSAF_REPORTVERSIONTYPE is CUSTOM, the custom version value. See “LSAF_CUSTOMVERSION=version” on page 6 for more information about this argument.

LSAF_REPORTCOMMENT=*comment*
The comment to associate with the new repository file.

SAS_RESULT_DSNAME=*name*
The name of the output data set that contains the completion status information about the import. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFIMPORTSTUDYTABLES.

The data set contains information that pertains to the import operation using the following columns:

Column Name	Description
studyId	The study identifier.
name	The name of the property.
value	The value of the property.

Details

Imports study value level metadata from the specified data set and can create a validation report as a CSV file in the specified location.

Failure and Message Handling

The macro performs a series of preliminary validation procedures to determine whether the specified parameters are valid and required parameters are specified. If none of these procedures fail, the import is attempted. The success or failure, and any issues that are encountered during the import, are reported as follows.

Codelists for the study are updated based on the UpdateAction that is specified. All other study metadata is replaced.

For more information, see Table 1.4 on page 9.

%LSAF_ISSTUDYDICTIONARY Macro

Indicates whether a controlled terminology dictionary is associated with a study.

Category: StudyService Module

Note: This macro sets the standard macro variables and _lsafIsStudyDictionary_, which indicates whether the controlled terminology dictionary is associated with the study.

Syntax

%LSAF_ISSTUDYDICTIONARY(LSAF_STUDYID=*study-id*, LSAF_DICTIONARYID=*dictionary-id*);

Required Arguments

LSAF_STUDYID=*study-id*
The identifier of the study. To get the identifier, call %LSAF_GETSTUDYID.

LSAF_DICTIONARYID=*dictionary-id*
The controlled terminology dictionary identifier. To get the identifier, call %LSAF_GETDICTIONARYID.

Details

Sets the Boolean macro variable _lsafIsStudyDictionary_, which indicates whether a controlled terminology dictionary is associated with a study. See “Boolean Macros” on page 4 for information about return values.

%LSAF_ISSTUDYEXTDICTIONARY Macro

Indicates whether an external dictionary is associated with a study.

Category: StudyService Module

Note: This macro sets the standard macro variables and _lsafIsStudyExtDictionary_, which indicates whether the external dictionary is associated with the study.

Syntax

%LSAF_ISSTUDYEXTDICTIONARY(LSAF_STUDYID=*study-id*, LSAF_EXTDICTIONARYID=*external-dictionary-id*);

Required Arguments

LSAF_STUDYID=*study-id*
The identifier of the study. To get the identifier, call %LSAF_GETSTUDYID.

LSAF_EXTDICTIONARYID=*external-dictionary-id*
The external dictionary identifier. To get the identifier, call %LSAF_GETEXTDICTIONARYID.

Details

Sets the Boolean macro variable _lsafIsStudyExtDictionary_, which indicates whether an external dictionary is associated with a study. See “Boolean Macros” on page 4 for information about return values.

%LSAF_ISSTUDYSTANDARD Macro

Indicates whether a standard is associated with a study.

Category: StudyService Module

Note: This macro sets the standard macro variables and _lsafIsStudyStandard_, which indicates whether the standard is associated with the study.

Syntax

%LSAF_ISSTUDYSTANDARD(LSAF_STUDYID=*study-id*, LSAF_STANDARDID=*standard-id*);

Required Arguments

LSAF_STUDYID=*study-id*
The identifier of the study. To get the identifier, call %LSAF_GETSTUDYID.

LSAF_STANDARDID=*standard-id*
The identifier of the standard. To get the identifier, call %LSAF_GETSTANDARDID.

Details

Sets the Boolean macro variable _lsafIsStudyStandard_, which indicates whether a standard is associated with a study. See “Boolean Macros” on page 4 for information about return values.

%LSAF_REMOVEALLSTUDYDICTIONARIES Macro

Removes all controlled terminology dictionaries from a study.

Category: StudyService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_REMOVEALLSTUDYDICTIONARIES(LSAF_STUDYID=*study-id*);

Required Argument

LSAF_STUDYID=*study-id*
The identifier of the study. To get the identifier, call %LSAF_GETSTUDYID.

%LSAF_REMOVESTUDYEXTDICTIONARY Macro

Removes an external dictionary from a study.

Category: StudyService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_REMOVESTUDYEXTDICTIONARY(LSAF_STUDYID=*study-id*, LSAF_EXTDICTIONARYID=*external-dictionary-id*);

Required Arguments

LSAF_STUDYID=*study-id*
The identifier of the study. To get the identifier, call %LSAF_GETSTUDYID.

LSAF_EXTDICTIONARYID=*external-dictionary-id*
The identifier of the external dictionary. To get the identifier, call %LSAF_GETEXTDICTIONARYID.

%LSAF_REMOVESTUDYSTANDARD Macro

Removes a standard from a study.

Category: StudyService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_REMOVESTUDYSTANDARD(LSAF_STUDYID=*study-id*, LSAF_STANDARDID=*standard-id*);

Required Arguments

LSAF_STUDYID=*study-id*
The identifier of the study. To get the identifier, call %LSAF_GETSTUDYID.

LSAF_STANDARDID=*standard-id*
The identifier of the standard. To get the identifier, call %LSAF_GETSTANDARDID.

%LSAF_SETSTDYCOMPONENTEDITSTATUS Macro

Sets the component edit status for a study.

Category: StudyService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_SETSTDYCOMPONENTEDITSTATUS(LSAF_STUDYID=*study-id*, LSAF_STUDYCOMPONENT=*component*<, LSAF_EDITCOMPONENT=0 | 1><, LSAF_FORCECLAIM=0 | 1>);

Required Arguments

LSAF_STUDYID=*study-id*
The identifier of the study. To get the identifier, call %LSAF_GETSTUDYID.

LSAF_STUDYCOMPONENT=*component*
The study component. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=StudyComponent`.

Optional Arguments

LSAF_EDITCOMPONENT=0 (Default) | 1
Indicates whether the component can be edited.

LSAF_FORCECLAIM=0 (Default) | 1
Indicates whether the operation is forced, even when you do not own the editability of the study component. You must have the user role privilege Manage Study.

%LSAF_UPDATESTUDYDICTIONARIES Macro

Updates the controlled terminology dictionaries for a study using a SAS data set as input.

Category: StudyService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_UPDATESTUDYDICTIONARIES(LSAF_STUDYID=*study-id*, SAS_DSNAME=*name*);

Required Arguments

LSAF_STUDYID=*study-id*
The identifier of the study. To get the identifier, call %LSAF_GETSTUDYID.

SAS_DSNAME=*name*
The name of a SAS data set that contains the dictionary identifiers for the study to update, specified as libref.dataset. The data set must contain at least the variable that is listed below. Additional variables are ignored.

The name of the data set that contains the dictionary identifiers for the study to update. See “Data Set Macros” on page 5 for information about specifying the name of the data set.

The data set must contain at least the following columns. Additional variables are ignored.

Column Name	Description
id	The identifier for the controlled terminology dictionary. The identifier can be retrieved by calling the lsaf_getdictionariesbystatus macro with the PUBLISHED option.

Details

Updates the controlled terminology dictionaries for a study using a SAS data set as input. The order of the dictionary identifiers in the input data set determines the order in which the dictionaries are associated to the study. This order is important and determines the precedence of duplicate code lists in the study. For example, if multiple dictionaries have the same named code list, only the code list from the first dictionary is associated with the study. All other duplicates are ignored.

Each dictionary identifier is represented as a single row in the input data set.

Use the output data set from the %LSAF_GETSTUDYDICTIONARIES macro to determine the controlled terminology dictionaries that are currently associated with a study. Consider running this macro first to have a record of the current associated controlled terminology dictionaries. If there are any issues with the updates, this data set can be used to reinstate the dictionary associations prior to running the updates.

The output data set from the %LSAF_GETSTUDYDICTIONARIES macro (published) can be used to determine dictionary identifiers that are available to associate with the study.

The dictionary identifiers that are represented in the input data set replace the list of dictionaries that are currently associated with the study.

Failure and Message Handling

The macro performs a series of validation procedures to determine whether the input data set has the necessary structure for processing. If any of these validation procedures fail, an appropriate failure message is written to the SAS log file and the macro stops additional processing.

If an invalid value is specified for a dictionary identifier, the updates are stopped and the appropriate error message is written to the SAS log file.

If duplicate valid dictionary identifiers are specified for the study, only the first value is maintained.

Column Name	Description
id	The identifier for the controlled terminology dictionary. To get the identifier, call %LSAF_GETDICTIONARIESBYSTATUS with the PUBLISHED option.

8

StudyTlfService Module

Overview

Macros to manipulate the Tables, Figures, and Listings (TLFs) of a study.

The properties for a TLF are configurable for each site. To determine the property metadata as configured at your site, call %LSAF_GETSTUDYTLFPROPERTIES.

Macro Summary Table

Category	Language Elements	Description
StudyTlf Module	%LSAF_DELETESTUDYTLF Macro (p. 107)	Deletes a study TLF.
	%LSAF_GETSTDYTLFSTITLESANDFOOTERS Macro (p. 107)	Gets the TITLES and FOOTNOTES properties of all or the specified study TLF. The data is stored in a SAS data set.
	%LSAF_GETSTUDYTLFANALYSISRESULTS Macro (p. 109)	Gets the analysis results for the specified TLF identifier.
	%LSAF_GETSTUDYTLFID Macro (p. 111)	Gets the alphanumeric id for a study TLF.
	%LSAF_GETSTUDYTLFPROPERTIES Macro (p. 111)	Gets all of the properties and their associated metadata of a study TLF. The properties are stored in a SAS data set.
	%LSAF_GETSTUDYTLFRESULTDATASETS Macro (p. 113)	Gets the analysis result data sets for the specified study TLF result identifier.
	%LSAF_GETSTUDYTLFSECTIONS Macro (p. 114)	Gets the metadata for the TLF sections for the specified study.
	%LSAF_GETSTUDYTLFSINFO Macro (p. 116)	Gets the non-configurable and non-derived metadata for the TLFs in the specified study.
	%LSAF_STUDYTLFAUTOFLOWENABLED Macro (p. 117)	Indicates whether a study TLF is autoflow-enabled.
	%LSAF_STUDYTLFEXISTS Macro (p. 118)	Indicates whether a TLF with the specified identifier exists within the specified study.
	%LSAF_UPDATESTDYTLFSTITLESANDFOOTERS Macro (p. 119)	Updates the TLF TITLES and FOOTNOTES using a SAS data set as input.
	%LSAF_UPDATESTUDYTLFPROPERTIES Macro (p. 121)	Updates the properties of a study TLF using a SAS data set as input.
	%LSAF_UPDATESTUDYTLFPROPERTY Macro (p. 123)	Updates a single property for a study TLF, using the property's SAS name.

Dictionary

%LSAF_DELETESTUDYTLF Macro

Deletes a study TLF.

Category: StudyTlf Module

Note: This macro sets the standard macro variables.

Syntax

LSAF_DELETESTUDYTLF(LSAF_STUDYID=*study-id*, LSAF_BASESTANDARDNAME=*name*, LSAF_MODELID=*model-id*, LSAF_TLFIDENTIFIER=*tlf-id*);

Required Arguments

LSAF_STUDYID=*study-id*
The identifier of the study. To get the identifier, call %LSAF_GETSTUDYID.

LSAF_BASESTANDARDNAME=*name*
The case-sensitive name of the base standard, such as SDTM and ADaM.

LSAF_MODELID=*model-id*
The identifier of the model. To get the identifier, call %LSAF_GETSTUDYSTANDARDS.

LSAF_TLFIDENTIFIER=*tlf-id*
The identifier of the TLF. To get the identifier, call %LSAF_GETSTUDYTLFSINFO.

%LSAF_GETSTDYTLFSTITLESANDFOOTERS Macro

Gets the TITLES and FOOTNOTES properties of all or the specified study TLF. The data is stored in a SAS data set.

Category: StudyTlf Module

Note: This macro sets the standard macro variables.

Syntax

LSAF_GETSTDYTLFSTITLESANDFOOTERS(LSAF_STUDYID=*study-id*, LSAF_BASESTANDARDNAME=*name*, LSAF_MODELID=*model-id*<, LSAF_TLFIDENTIFIER=*tlf-id*><, SAS_DSNAME=*name*>);

Required Arguments

LSAF_STUDYID=*study-id*
The identifier of the study. To get the identifier, call %LSAF_GETSTUDYID.

LSAF_BASESTANDARDNAME=*name*
The case-sensitive name of the base standard, such as SDTM and ADaM.

LSAF_MODELID=*model-id*
The identifier of the model. To get the identifier, call %LSAF_GETSTUDYSTANDARDS.

Optional Arguments

LSAF_TLFIDENTIFIER=*tlf-id*
The identifier of the TLF. To get the identifier, call %LSAF_GETSTUDYTLFSINFO. If omitted, the titles and footnotes for all study TLFs are returned.

SAS_DSNAME=*name*
The name of the output data set that contains the TITLE and FOOTNOTE properties for the TLF. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETSTDYTLFSTITLESANDFOOTERS.

The data set is sorted by the TLF identifier and contains the following columns:

Column Name	Description
studyId	The identifier of the study.
baseStandardName	The base data standard name.
modelId	The identifier of the model.
tlfIdentifier	The identifier of the TLF.
section	The identifier of the section in which the TLF is associated.
type	The type of the TLF (for example, Listing).
name	The name of the TLF.
TITLE1-TITLEx	The values for the TITLE properties that are defined.

Column Name	Description
FOOTNOTE1-FOOTNOTEx	The values for the FOOTNOTE properties that are defined.

Details

Gets the TITLES and FOOTNOTES properties of all of the specified study TLF. The data is stored in a SAS data set.

Note: This macro relies on the TLF model having attributes that start with “TITLE” or “FOOTNOTE”. Changes to the TLF model can impact the operations that are performed by this macro.

%LSAF_GETSTUDYTLFANALYSISRESULTS Macro

Gets the analysis results for the specified TLF identifier.

Category: StudyTlf Module

Note: This macro sets the standard macro variables.

Syntax

LSAF_GETSTUDYTLFANALYSISRESULTS(LSAF_STUDYID=*study-id*, LSAF_BASESTANDARDNAME=*name*, LSAF_MODELID=*model-id*, LSAF_TLFIDENTIFIER=*tlf-id*<, SAS_DSNAME=*name*>);

Required Arguments

LSAF_STUDYID=*study-id*
The identifier of the study. To get the identifier, call %LSAF_GETSTUDYID.

LSAF_BASESTANDARDNAME=*name*
The case-sensitive name of the base standard, such as SDTM and ADaM.

LSAF_MODELID=*model-id*
The identifier of the model. To get the identifier, call %LSAF_GETSTUDYSTANDARDS.

LSAF_TLFIDENTIFIER=*tlf-id*
The identifier of the TLF. To get the identifier, call %LSAF_GETSTUDYTLFSINFO.

Optional Argument

SAS_DSNAME=*name*

The name of the output data set that contains the list of TLF analysis results. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETSTUDYTLFANALYSISRESULTS.

The data set contains a row for each result and columns with the following names:

Column Name	Description
studyId	The identifier of the study.
baseStandardName	The base data standard name.
modelId	The identifier of the model.
tlfIdentifier	The identifier of the TLF to which this result is associated.
resultIdentifier	The unique identifier of the analysis result.
description	The description of the analysis result.
reason	The reason for the analysis result.
purpose	The purpose of the analysis result.
analysisParameter	The analysis parameter of the analysis result.
documentation	The documentation of the analysis result.
code	The code for the analysis result. If the text exceeds the maximum length (32767 characters) of a SAS data set variable, the code is truncated.
codeContext	The code context of the analysis result.
tableJoinComment	The comment for a table join of the analysis result.
order	The order number of the analysis result.
dataSetCount	The data set count for the analysis result.

%LSAF_GETSTUDYTLFID Macro

Gets the alphanumeric id for a study TLF.

Category: StudyTlf Module

Note: This macro sets the standard macro variables and _lsafStudyTlfId_, which is the unique alphanumeric ID of the study TLF.

Syntax

LSAF_GETSTUDYTLFID(LSAF_STUDYID=*study-id*, LSAF_BASESTANDARDNAME=*name*, LSAF_MODELID=*model-id*, LSAF_TLFIDENTIFIER=*tlf-id*);

Required Arguments

LSAF_STUDYID=*study-id*
The identifier of the study. To get the identifier, call %LSAF_GETSTUDYID.

LSAF_BASESTANDARDNAME=*name*
The case-sensitive name of the base standard, such as SDTM and ADaM.

LSAF_MODELID=*model-id*
The identifier of the model. To get the identifier, call %LSAF_GETSTUDYSTANDARDS.

LSAF_TLFIDENTIFIER=*tlf-id*
The identifier of the TLF. To get the identifier, call %LSAF_GETSTUDYTLFSINFO.

Details

Sets the value of the macro variable _lsafStudyTlfId_ to the unique alphanumeric id of the study TLF. See “Value Macros” on page 5 for information about value macros.

You must have Read access permission to the study.

%LSAF_GETSTUDYTLFPROPERTIES Macro

Gets all of the properties and their associated metadata of a study TLF. The properties are stored in a SAS data set.

Category: StudyTlf Module

Note: This macro sets the standard macro variables.

Syntax

LSAF_GETSTUDYTLFPROPERTIES(LSAF_STUDYID=*study-id*, LSAF_BASESTANDARDNAME=*name*, LSAF_MODELID=*model-id*, LSAF_TLFIDENTIFIER=*tlf-id*<, SAS_DSNAME=*name*>);

Required Arguments

LSAF_STUDYID=*study-id*
The identifier of the study. To get the identifier, call %LSAF_GETSTUDYID.

LSAF_BASESTANDARDNAME=*name*
The case-sensitive name of the base standard, such as SDTM and ADaM.

LSAF_MODELID=*model-id*
The identifier of the model. To get the identifier, call %LSAF_GETSTUDYSTANDARDS.

LSAF_TLFIDENTIFIER=*tlf-id*
The identifier of the TLF. To get the identifier, call %LSAF_GETSTUDYTLFSINFO.

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the properties for the TLF. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETSTUDYTLFPROPERTIES.

The data set is sorted by the ordering metadata element and contains the following columns:

Column Name	Description
studyId	The identifier of the study.
baseStandardName	The base data standard name.
modelId	The identifier of the model .
tlfIdentifier	The identifier of the TLF.
sasName	The SAS name of the property.
name	The property name.
description	The property description.
ordering	The display order of the property.
dataType	The data type of the property. For example, String or Numeric.

Column Name	Description
logicalType	The logical type of the property. For example, User, Date, or Folder.
dataLength	The maximum length of the property value.
isRequired	Indicates whether the property is required.
mustBeUnique	Indicates whether the property must be unique among TLFs within the modelId.
allowedValues	The comma-delimited list of valid values. For example, for property type: Listing, Table, or Figure.
value	The value of the property.
valueQualifier	Additional information that helps to define the property value. For example, file version or user group context.

%LSAF_GETSTUDYTLFRESULTDATASETS Macro

Gets the analysis result data sets for the specified study TLF result identifier.

Category: StudyTlf Module

Note: This macro sets the standard macro variables.

Syntax

LSAF_GETSTUDYTLFRESULTDATASETS(LSAF_STUDYID=*study-id*, LSAF_BASESTANDARDNAME=*name*, LSAF_MODELID=*model-id*, LSAF_RESULTIDENTIFIER=*result-id*<, SAS_DSNAME=*name*>);

Required Arguments

LSAF_STUDYID=*study-id*
The identifier of the study. To get the identifier, call %LSAF_GETSTUDYID.

LSAF_BASESTANDARDNAME=*name*
The case-sensitive name of the base standard, such as SDTM and ADaM.

LSAF_MODELID=*model-id*
The identifier of the model. To get the identifier, call %LSAF_GETSTUDYSTANDARDS.

LSAF_RESULTIDENTIFIER=*result-id*
The identifier of the analysis result. To get the identifier, call %LSAF_GETSTUDYTLFANALYSISRESULTS.

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the list of TLF analysis results. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETSTUDYTLFRESULTDATASETS.

The data set contains a row for each data set and columns with the following names:

Column Name	Description
studyId	The identifier of the study.
baseStandardName	The base data standard name.
modelId	The identifier of the model.
resultIdentifier	The identifier of the analysis to which this data set is associated.
name	The name of the analysis data set.
analysisColumns	The analysis columns of the data set.
whereClause	The WHERE clause of the data set.

%LSAF_GETSTUDYTLFSECTIONS Macro

Gets the metadata for the TLF sections for the specified study.

Category: StudyTlf Module

Note: This macro sets the standard macro variables.

Syntax

LSAF_GETSTUDYTLFSECTIONS(LSAF_STUDYID=*study-id*, LSAF_BASESTANDARDNAME=*name*, LSAF_MODELID=*model-id*<, SAS_DSNAME=*name*>);

Required Arguments

LSAF_STUDYID=*study-id*
The identifier of the study. To get the identifier, call %LSAF_GETSTUDYID.

LSAF_BASESTANDARDNAME=*name*
The case-sensitive name of the base standard, such as SDTM and ADaM.

LSAF_MODELID=*model-id*
The identifier of the model. To get the identifier, call %LSAF_GETSTUDYSTANDARDS.

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the sections. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETSTUDYTLFSECTIONS.

The data set contains a row for each section and columns with the following names:

Column Name	Description
studyId	The identifier of the study.
baseStandardName	The base data standard name.
modelId	The model identifier.
parentSectionId	The identifier of the section that is the immediate parent of the section represented in the observation.
sectionId	The id of the section.
sectionName	The name of the section.
sectionIdentifier	The identifier of the section.
sectionDescription	The description of the section.

%LSAF_GETSTUDYTLFSINFO Macro

Gets the non-configurable and non-derived metadata for the TLFs in the specified study.

Category: StudyTlf Module

Note: This macro sets the standard macro variables.

Syntax

LSAF_GETSTUDYTLFSINFO(LSAF_STUDYID=*study-id*, LSAF_BASESTANDARDNAME=*name*, LSAF_MODELID=*model-id*<, LSAF_SECTIONID=*section-id*><, LSAF_RECURSE=0 | 1 (Default)><, SAS_DSNAME=*name*>);

Required Arguments

LSAF_STUDYID=*study-id*
The identifier of the study. To get the identifier, call %LSAF_GETSTUDYID.

LSAF_BASESTANDARDNAME=*name*
The case-sensitive name of the base standard, such as SDTM and ADaM.

LSAF_MODELID=*model-id*
The identifier of the model. To get the identifier, call %LSAF_GETSTUDYSTANDARDS.

Optional Arguments

LSAF_SECTIONID=*section-id*
The identifier of the parent section for the TLFs. If not specified, the root section is used as the starting level. To get the sectionId, call this macro without specifying *section-id* or call %LSAF_GETSTUDYTLFSECTIONS.

LSAF_RECURSE=0 | 1 (Default)
Indicates whether the search recurses through child sections.

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the TLFs. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETSTUDYTLFSINFO.

The data set contains a row for each TLF and columns with the following names:

Column Name	Description
studyId	The identifier of the study.
baseStandardName	The base data standard name.

Column Name	Description
modelId	The model identifier.
sectionId	The id of the section.
tlfIdentifier	The identifier of the TLF.
tlfName	The name of the TLF.
tlfType	The type of the TLF.
tlfAutoflowEnabled	Indicates whether the TLF is autoflow enabled.

%LSAF_STUDYTLFAUTOFLOWENABLED Macro

Indicates whether a study TLF is autoflow-enabled.

Category: StudyTlf Module

Note: This macro sets the standard macro variables and _lsafStudyTlfAutoflowEnabled_, which indicates whether the TLF is autoflow enabled.

Syntax

LSAF_STUDYTLFAUTOFLOWENABLED(LSAF_STUDYID=*study-id*, LSAF_BASESTANDARDNAME=*name*, LSAF_MODELID=*model-id*, LSAF_TLFIDENTIFIER=*tlf-id*);

Required Arguments

LSAF_STUDYID=*study-id*
The identifier of the study. To get the identifier, call %LSAF_GETSTUDYID.

LSAF_BASESTANDARDNAME=*name*
The case-sensitive name of the base standard, such as SDTM and ADaM.

LSAF_MODELID=*model-identifier*
The identifier of the model. To get the identifier, call %LSAF_GETSTUDYSTANDARDS.

LSAF_TLFIDENTIFIER=*tlf-id*
The identifier of the TLF. To get the identifier, call %LSAF_GETSTUDYTLFSINFO.

Details

Sets the Boolean macro variable _lsafStudyTlfAutoflowEnabled_, which indicates whether a study TLF is autoflow-enabled. See “Boolean Macros” on page 4 for information about return values.

%LSAF_STUDYTLFEXISTS Macro

Indicates whether a TLF with the specified identifier exists within the specified study.

Category: StudyTlf Module

Note: This macro sets the standard macro variables and _lsafStudyTlfExists_, which indicates whether the TLF exists in the study.

Syntax

LSAF_STUDYTLFEXISTS(LSAF_STUDYID=*study-id*, LSAF_BASESTANDARDNAME=*name*, LSAF_MODELID=*model-id*, LSAF_TLFIDENTIFIER=*tlf-id*);

Required Arguments

LSAF_STUDYID=*study-id*
The identifier of the study. To get the identifier, call %LSAF_GETSTUDYID.

LSAF_BASESTANDARDNAME=*name*
The case-sensitive name of the base standard, such as SDTM and ADaM.

LSAF_MODELID=*model-id*
The identifier of the model. To get the identifier, call %LSAF_GETSTUDYSTANDARDS.

LSAF_TLFIDENTIFIER=*tlf-id*
The identifier of the TLF. To get the identifier, call %LSAF_GETSTUDYTLFSINFO.

Details

Sets the Boolean macro variable _lsafStudyTlfExists_, which indicates whether a TLF with the specified identifier exists within the specified study. See “Boolean Macros” on page 4 for information about return values.

%LSAF_UPDATESTDYTLFSTITLESANDFOOTERS Macro

Updates the TLF TITLES and FOOTNOTES using a SAS data set as input.

Category: StudyTlf Module

Note: This macro sets the standard macro variables.

Syntax

LSAF_UPDATESTDYTLFSTITLESANDFOOTERS(LSAF_STUDYID=*study-id*, LSAF_BASESTANDARDNAME=*name*, LSAF_MODELID=*model-id*, LSAF_TLFIDENTIFIER=*tlf-id*, SAS_DSNAME=*name*<, SAS_RESULTS_DSNAME=*name*>);

Required Arguments

LSAF_STUDYID=*study-id*
The identifier of the study. To get the identifier, call %LSAF_GETSTUDYID.

LSAF_BASESTANDARDNAME=*name*
The case-sensitive name of the base standard, such as SDTM and ADaM.

LSAF_MODELID=*model-id*
The identifier of the model. To get the identifier, call %LSAF_GETSTUDYSTANDARDS.

LSAF_TLFIDENTIFIER=*tlf-id*
The identifier of the TLF. To get the identifier, call %LSAF_GETSTUDYTLFSINFO.

SAS_DSNAME=*name*
The name of the data set that contains the information for all of the TLFs to modify. See “Data Set Macros” on page 5 for information about specifying the name of the data set.

The data set must contain at least the following columns. Additional variables are ignored.

Column Name	Description
tlfIdentifier	The identifier of the TLF to update.
<TITLE_1-TITLE_x, FOOTNOTE_1-FOOTNOTE_x>	Variables that indicate the properties to update. The variable names must match the SAS name of the property. At least one variable must be included.

Optional Argument

SAS_RESULTS_DSNAME=*name*

The name of the output data set that contains the results of the attempt to update each of the properties that were represented in the input data set. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFUPDATESTDYTLFSTITLESANDFOOTERS.

The data set contains a row for each property and columns with the following names:

Column Name	Description
studyId	The identifier of the study.
baseStandardName	The base data standard name.
modelId	The model identifier.
tlfIdentifier	The identifier of the TLF.
variable	The name of the variable in the input data set that was processed.
propertyName	The name of the TLF TITLE or FOOTNOTE property matched to the variable.
status	The result of the update operation. Valid values: ■ Property not found. ■ Property not applicable to the operation. ■ The property was processed.

Details

Updates the TLF TITLES and FOOTNOTES using a SAS data set as input.

Note: This macro relies on the TLF model having attributes that start with “TITLE” or “FOOTNOTE”. Changes to the TLF model can impact the operations performed by this macro.

The input data set is expected to have a variable for the TLF identifier and each TITLE or FOOTNOTE to update. The output data set from this macro contains the necessary variables and data needed as the input data set for this macro. Modify the data set with changes prior to calling this macro.

Variables in the input data set that do not correspond with a TITLE or FOOTNOTE property are ignored.

For other methods to update TLF properties, see %LSAF_UPDATESTUDYTLFPROPERTIES and %LSAF_UPDATESTUDYTLFPROPERTY.

Failure and Message Handling

This macro performs a series of validation procedures to determine that the input data set has the necessary structure for processing and that the output data set name is a valid SAS data set name. If any of these validation procedures fail, an appropriate failure message is printed to the SAS log file and the macro stops processing.

This macro sorts and performs a PROC TRANSPOSE on the input data set and stores the modified data in a temporary data set called WORK.__SASMACRO_STDYTLFSTITLESFOOTS__. In the event that a failure occurs during processing, this temporary data set is retained as a reference. If all updates are performed successfully, this data set is deleted from the work library. A message is printed to the SAS log file that indicates whether the data set was retained. Regardless of the success or failure of previous runs, this data set is deleted at the beginning of each new run of the macro. The data values and sort order of the original input data set are maintained.

All records in the temporary data set that pertain to the same TLF are processed as a single action by the macro. Messages are printed to the SAS log file as each TLF is processed.

If a failure occurs while processing a specific TLF, the macro stops processing, and a failure message is written to the SAS log file. Successful updates to TLFs made prior to the TLF that contains the failure are maintained in SAS Life Science Analytics Framework. Changes made to TITLE or FOOTNOTE properties that are part of the same TLF that contained the failure are lost.

%LSAF_UPDATESTUDYTLFPROPERTIES Macro

Updates the properties of a study TLF using a SAS data set as input.

Category: StudyTlf Module

Note: This macro sets the standard macro variables.

Syntax

LSAF_UPDATESTUDYTLFPROPERTIES(LSAF_STUDYID=*study-id*, LSAF_BASESTANDARDNAME=*name*, LSAF_MODELID=*model-id*, LSAF_TLFIDENTIFIER=*tlf-id*<, SAS_DSNAME=*name*>);

Required Arguments

LSAF_STUDYID=*study-id*
The identifier of the study. To get the identifier, call %LSAF_GETSTUDYID.

LSAF_BASESTANDARDNAME=*name*
The case-sensitive name of the base standard, such as SDTM and ADaM.

LSAF_MODELID=*model-id*
The identifier of the model. To get the identifier, call %LSAF_GETSTUDYSTANDARDS.

LSAF_TLFIDENTIFIER=*tlf-id*
The identifier of the TLF. To get the identifier, call %LSAF_GETSTUDYTLFSINFO.

Optional Argument

SAS_DSNAME=*name*
The name of the data set that contains the properties information for all of the notification element properties to modify. See “Data Set Macros” on page 5 for information about specifying the name of the data set.

The data set must contain at least the following columns. Additional variables are ignored.

Column Name	Description
sasName	The SasName of the property.
value	The value of the property.
valueQualifier	Additional information that is relevant to the property based on the logical type. A value is required when the logical type is a GROUP and must be the full path of the context in which the user group is defined. This value can be used when the logical type is FILE to indicate the version of the file. This value is ignored for other property types or when setting the value to blank.

Details

Updates the properties of a study TLF, using a SAS data set as input.

The output data set from the %LSAF_GETSTUDYTLFPROPERTIES macro contains the necessary variables and data needed as the input data set for this macro. Modify the data set with changes prior to calling this macro.

Data in the input data set that do not correspond to a property of the TLF to update are ignored.

Values that are specified for properties that are not editable are ignored.

To update TLF titles and footnotes, see %LSAF_UPDATESTDYTLFSTITLESANDFOOTERS on page 119. To update individual properties, see %LSAF_UPDATESTUDYTLFPROPERTY on page 123.

%LSAF_UPDATESTUDYTLFPROPERTY Macro

Updates a single property for a study TLF, using the property's SAS name.

Category: StudyTlf Module

Note: This macro sets the standard macro variables.

Syntax

LSAF_UPDATESTUDYTLFPROPERTY(LSAF_STUDYID=*study-id*, LSAF_BASESTANDARDNAME=*name*, LSAF_MODELID=*model-id*, LSAF_TLFIDENTIFIER=*tlf-id*, LSAF_SAS_NAME=*name*<, LSAF_VALUE=*value*><, LSAF_VALUEQUALIFIER=*qualifier*>);

Required Arguments

LSAF_STUDYID=*study-id*
The identifier of the study. To get the identifier, call %LSAF_GETSTUDYID.

LSAF_BASESTANDARDNAME=*name*
The case-sensitive name of the base standard, such as SDTM and ADaM.

LSAF_MODELID=*model-id*
The identifier of the model. To get the identifier, call %LSAF_GETSTUDYSTANDARDS.

LSAF_TLFIDENTIFIER=*tlf-id*
The identifier of the TLF. To get the identifier, call %LSAF_GETSTUDYTLFSINFO.

LSAF_SAS_NAME=*name*
The SAS name of the property.

Optional Arguments

LSAF_VALUE=*value*
The value of the property. If not specified, the property is set to blank.

LSAF_VALUEQUALIFIER=*qualifier*
Additional information that is relevant to the property, based on the logical type.

Required when the logical type is a GROUP and must be the full path of the context in which the user group is defined.

Can be used when logical type is FILE to indicate the version of the file.

The value is ignored for other property types or when setting the value to blank.

9

TLFTemplateService Module

Overview

Macros to manage the global Titles, Listings, and Figures (TLF) templates.

Macro Summary Table

Category	Language Elements	Description
TLFTemplateService Module	%LSAF_CREATEGLOBALTLFTEMPLATE Macro (p. 126)	Creates a new global TLF template.
	%LSAF_EXPORTGLOBALTLFS Macro (p. 127)	Exports the metadata for the Tables, Listings, and Figures (TLF) that are associated with the specified global TLF template to the specified area in the repository. The paths of the exported files are stored in SAS data sets.

Category	Language Elements	Description
	%LSAF_EXPORTGLOBALTLFSTOWORKSPACE Macro (p. 128)	Exports the metadata for the Tables, Listings, and Figures (TLF) that are associated with the specified global TLF template to the specified area in the workspace. The paths of the exported files are stored in SAS data sets.
	%LSAF_GETGLOBALTLFTEMPLATES Macro (p. 130)	Gets the metadata for the global Tables, Listing, and Figures templates. The metadata is stored in a SAS data set.
	%LSAF_GETTLFTEMPLATEID Macro (p. 131)	Gets the alphanumeric identifier of a Tables, Listings, and Figures (TLF) template.
	%LSAF_IMPORTGLOBALTLFS Macro (p. 131)	Imports Tables, Listings, and Figures (TLF) to the TLF template from the specified data sets.

Dictionary

%LSAF_CREATEGLOBALTLFTEMPLATE Macro

Creates a new global TLF template.

Category: TLFTemplateService Module

Note: This macro sets the standard macro variables and _lsaftlftemplateid_, which is the identifier of the new TLF template.

Syntax

%LSAF_CREATEGLOBALTLFTEMPLATE(LSAF_NAME=*name*<, LSAF_DESCRIPTION=*description*>);

Required Argument

LSAF_NAME=*name*
The name of the TLF template.

Optional Argument

LSAF_DESCRIPTION=*description*
The description of the TLF template.

Details

Sets the value of the macro variable _lsaftlftemplateid_ to the identifier of the new TLF template. See “Value Macros” on page 5 for information about value macros.

%LSAF_EXPORTGLOBALTLFS Macro

Exports the metadata for the Tables, Listings, and Figures (TLF) that are associated with the specified global TLF template to the specified area in the repository. The paths of the exported files are stored in SAS data sets.

Category: TLFTemplateService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_EXPORTGLOBALTLFS(LSAF_TLFTEMPLATEID=*template-id*, LSAF_PATH=*path*, LSAF_PREFIX=*prefix*<, LSAF_INCLUDERESULTS=0 | 1><, LSAF_OVERWRITE=0 | 1><, LSAF_ENABLEVERSIONING=0 | 1><, LSAF_VERSIONTYPEFORNEWFILES=MAJOR | MINOR | CUSTOM><, LSAF_CUSTOMVERSFORNEWFILES=*version*><, LSAF_VERSIONTYPEFOREXISTINGFILES=MAJOR | MINOR | CUSTOM><, LSAF_CUSTOMVERSFOREXISTINGFILES=*version*><, LSAF_COMMENT=*comment*><, SAS_RESULT_DSNAME=*name*>);

Required Arguments

LSAF_TLFTEMPLATEID=*template-id*
The identifier of the global TLF template. To get the identifier, call %LSAF_GETTLFTEMPLATEID.

LSAF_PATH=*path*
The case-sensitive output path for the exported files.

LSAF_PREFIX=*prefix*
The lowercase prefix of the SAS data sets to create for the exported tables.

Optional Arguments

LSAF_INCLUDERESULTS=0 (Default) | 1
Indicates whether to include the results data sets.

LSAF_OVERWRITE=0 (Default) | 1
Indicates whether an existing unversioned repository or existing workspace file is overwritten by a new file with a same path. See “LSAF_OVERWRITE=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_ENABLEVERSIONING=0 (Default) | 1
Indicates whether to enable versioning for a new file in the repository. See “LSAF_ENABLEVERSIONING=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_VERSIONTYPEFORNEWFILES=MAJOR | MINOR | CUSTOM
Indicates the version type to apply to new files. See “LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM” on page 6 for more information about this argument.

LSAF_CUSTOMVERSFORNEWFILES=*version*
Specifies the version number to use to create a customized version of a new versioned file. See “LSAF_CUSTOMVERSION=version” on page 6 for more information about this argument.

LSAF_VERSIONTYPEFOREXISTINGFILES=MAJOR | MINOR | CUSTOM
Indicates the version type to apply to a new version of an existing versioned repository file. See “LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM” on page 6 for more information about this argument.

LSAF_CUSTOMVERSFOREXISTINGFILES=*version*
Specifies the version number to use to create a customized version of an existing versioned file. See “LSAF_CUSTOMVERSION=version” on page 6 for more information about this argument.

LSAF_COMMENT=*comment*
The check-in comment to associate with the action of writing a file to the repository. When writing to the workspace, this argument is ignored.

SAS_RESULT_DSNAME=*name*
The name of the output data set that contains the exported TLFs. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFEXPORTGLOBALTLFS.

The data set contains a row for each exported data set, sorted by path, and the following columns:

Column Name	Description
location	The location of the exported file (REPOSITORY).
name	The name of the exported file.
path	The path of the exported file.

%LSAF_EXPORTGLOBALTLFSTOWORKSPACE Macro

Exports the metadata for the Tables, Listings, and Figures (TLF) that are associated with the specified global TLF template to the specified area in the workspace. The paths of the exported files are stored in SAS data sets.

Category: TLFTemplateService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_EXPORTGLOBALTLFSTOWORKSPACE(LSAF_TLFTEMPLATEID=*template-id*, LSAF_PATH=*path*, LSAF_PREFIX=*prefix*<, LSAF_INCLUDERESULTS=0 | 1><, LSAF_OVERWRITE=0 | 1><, SAS_RESULT_DSNAME=*name*>);

Required Arguments

LSAF_TLFTEMPLATEID=*template-id*
The identifier of the global TLF template. To get the identifier, call %LSAF_GETTLFTEMPLATEID.

LSAF_PATH=*path*
The case-sensitive output path for the exported files.

LSAF_PREFIX=*prefix*
The lowercase prefix of the SAS data set names that are created for the exported tables.

Optional Arguments

LSAF_INCLUDERESULTS=0 | 1 (Default)
Indicates whether to include the results data sets.

LSAF_OVERWRITE=0 (Default) | 1
Indicates whether an existing unversioned repository or existing workspace file is overwritten by a new file with the same path. See “LSAF_OVERWRITE=0 (Default) | 1” on page 6 for more information about this argument.

SAS_RESULT_DSNAME=*name*
The name of the output data set that contains the exported TLFs. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFEXPORTGLOBALTLFSTOWORKSPACE.

The data set contains a row for each exported data set, sorted by path, and the following columns:

Column Name	Description
Location	The location of the exported file (WORKSPACE).
Name	The name of the exported file.
Path	The path of the exported file.

%LSAF_GETGLOBALTLFTEMPLATES Macro

Gets the metadata for the global Tables, Listing, and Figures templates. The metadata is stored in a SAS data set.

Category: TLFTemplateService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETGLOBALTLFTEMPLATES(< SAS_DSNAME=*name*>);

Optional Argument

SAS_DSNAME=*name*

The name of the output data set that contains the TLF template metadata. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETGLOBALTLFTEMPLATES.

The data set contains a row for each template and columns with the following names:

Column Name	Description
id	The identifier of the TLF template.
name	The name of the TLF template.
description	The description of the TLF template.
createdBy	The identifier of the user who created the TLF template.
createdDate	The date on which the TLF template was created.
createdDateSASFormat	The date on which the TLF template was created, in SAS format.
lastModifiedBy	The identifier of the user who last modified the TLF template.
lastModifiedDate	The date on which the TLF template was last modified.

Column Name	Description
lastModifiedDateSASFormat	The date on which the TLF template was last modified, in SAS format.

Note: See "Format Date Values" on page 4 for information about date formats.

%LSAF_GETTLFTEMPLATEID Macro

Gets the alphanumeric identifier of a Tables, Listings, and Figures (TLF) template.

Category: TLFTemplateService Module

Note: This macro sets the standard macro variables and _lsafTlfTemplateId_, which is the TLF template identifier.

Syntax

%LSAF_GETTLFTEMPLATEID(LSAF_NAME=*name*);

Required Argument

LSAF_NAME=*name*
The name of the TLF template.

Details

Sets the value of the macro variable _lsafTlfTemplateId_ to the TLF template identifier. See "Value Macros" on page 5 for information about value macros.

%LSAF_IMPORTGLOBALTLFS Macro

Imports Tables, Listings, and Figures (TLF) to the TLF template from the specified data sets.

Category: TLFTemplateService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_IMPORTGLOBALTLFS(LSAF_TLFTEMPLATEID=*template-id*, LSAF_IMPORTACTION=*update-action*, LSAF_LOCATION=WORKSPACE | REPOSITORY, LSAF_SECTIONSPATH=*path*, LSAF_TLFSPATH=*path*<, LSAF_SECTIONSVERSION=*version*><, LSAF_TLFSVERSION=*version*><, LSAF_RESULTSPATH=*path*><, LSAF_RESULTSVERSION=*version*><, LSAF_DATASETSPATH=*path*><, LSAF_DATASETSVERSION=*version*><, LSAF_REPORTLOCATION=*location*><, LSAF_REPORTPATH=*path*><, LSAF_REPORTOVERWRITE=0 | 1><, LSAF_REPORTENABLEVERSIONING=0 | 1><, LSAF_REPORTVERSIONTYPE=MAJOR | MINOR | CUSTOM><, LSAF_REPORTCUSTOMVERSION=*version*><, LSAF_REPORTCOMMENT=*comment*><, SAS_RESULT_DSNAME=*name*>);

Required Arguments

LSAF_TLFTEMPLATEID=*template-id*
The identifier of the global TLF template. To get the identifier, call %LSAF_GETTLFTEMPLATEID.

LSAF_IMPORTACTION=*update-action*
The case-insensitive value to indicate the action to take by the import. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=UpdateAction`.

LSAF_LOCATION=WORKSPACE | REPOSITORY
The location of the import source files.

LSAF_SECTIONSPATH=*path*
The path of the data set to import the TLF sections.

LSAF_TLFSPATH=*path*
The path of the data set to import the TLFs.

Optional Arguments

LSAF_SECTIONSVERSION=*version*
The version of the data set to import the TLF sections.

LSAF_TLFSVERSION=*version*
The version of the data set to import the TLFs.

LSAF_RESULTSPATH=*path*
The path of the data set to import the TLF results. This argument is required when a LSAF_DATASETSPATH is specified.

LSAF_RESULTSVERSION=*version*
The version of the data set to import the TLF results.

LSAF_DATASETSPATH=*path*
The path of the data set to import the TLF data sets.

LSAF_DATASETSVERSION=*version*
The version of the data set to import the TLF data sets.

LSAF_REPORTLOCATION=*location*
The location to create the report. If the report location and report path are not specified, the import runs but no validation report is created.

LSAF_REPORTPATH=*path*
The path of the report, which includes the report file name. If the report location and report path are not specified, the import runs but no validation report is created.

LSAF_REPORTOVERWRITE=0 (Default) | 1
Indicates whether an existing unversioned repository or existing workspace file is overwritten by a new file with a same path. See “LSAF_OVERWRITE=0 (Default) | 1” on page 6 for more information about the options for this argument.

LSAF_REPORTENABLEVERSIONING=0 (Default) | 1
Indicates whether to enable versioning for a new file in the repository. See “LSAF_ENABLEVERSIONING=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_REPORTVERSIONTYPE=MAJOR | MINOR | CUSTOM
Indicates the version type to apply to new files or a new version of an existing versioned repository file. See “LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM” on page 6 for more information about this argument.

LSAF_REPORTCUSTOMVERSION=*version*
Specifies the version number to use to create a customized version of a new or existing versioned file. See “LSAF_CUSTOMVERSION=version” on page 6 for more information about this argument.

LSAF_REPORTCOMMENT=*comment*
The check-in comment to associate with the action of writing a file to the repository. When writing to the workspace, this argument is ignored.

SAS_RESULT_DSNAME=*name*
The name of the output data set that contains the completion status information about the import. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFIMPORTGLOBALTLFS.

The data set contains information pertaining to the import operation using the following columns:

Column Name	Description
tlfTemplateId	The TLF template identifier.
name	The name of the property.
value	The value of the property.

Details

Imports Tables, Listings, and Figures (TLF) from the specified data sets.

You can choose to create a validation report as a CSV file in the specified location.

Failure and Message Handling

This macro performs a series of preliminary validation procedures to determine whether the specified parameters are valid and required parameters are specified. If

none of these procedures fail, the import is attempted. The success or failure, and any issues that are encountered during the import, are reported. For more information, see Table 1.4 on page 9.

PART 3

Content Macros

10

BatchRepositoryService Module

Overview

Macros to perform batch operations in the repository.

Note: These macros are supported only from a PC SAS session. Calling a macro from within an active SAS Life Science Analytics Framework session results in a failure.

Macro Summary Table

Category	Language Elements	Description
BatchRepository Service Module	%LSAF_DOWNLOADASZIP Macro (p. 138)	Downloads a file or folder along with its content to a local computer as a ZIP file.
	%LSAF_UPLOADANDEXPAND Macro (p. 139)	Uploads and expands the contents of a ZIP file into the specified container location in the repository.

Dictionary

%LSAF_DOWNLOADASZIP Macro

Downloads a file or folder along with its content to a local computer as a ZIP file.

Category: BatchRepositoryService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_DOWNLOADASZIP(LSAF_PATH=*lsaf-path*, LOCAL_PATH=*local-path*<, LSAF_OVERWRITE=1 | 0>);

Required Arguments

LSAF_PATH=*lsaf-path*
The path of the item to download.

LOCAL_PATH=*local-path*
The full path and name of the file to create (or overwrite, if applicable) on the local computer. File extensions are entered on the file as specified. If no extension is specified, the ZIP file has no extension.

Optional Argument

LSAF_OVERWRITE=1 | 0 (Default)
Indicates whether to overwrite the current ZIP file on the local computer. See “LSAF_OVERWRITE=0 (Default) | 1” on page 6 for more information about this argument.

Details

Downloads a file or folder along with its content to a local computer as a ZIP file.

If the local directory does not exist, it is created. If the local file does not exist, it is created. If the item to download is a container, the contents of the container (child contexts, subfolders, and files) are downloaded and the hierarchy is maintained.

Note: This macro is supported only from a PC SAS session. Calling this macro within an active SAS Life Science Analytics Framework session results in a failure.

%LSAF_UPLOADANDEXPAND Macro

Uploads and expands the contents of a ZIP file into the specified container location in the repository.

Category: BatchRepositoryService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_UPLOADANDEXPAND(LOCAL_PATH=*local-path*, LSAF_PATH=*lsaf-path*<, LSAF_ENABLEVERSIONING=0 | 1><, LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM><, LSAF_CUSTOMVERSION=*version*><, LSAF_COMMENT=*comment*>);

Required Arguments

LOCAL_PATH=*local-path*
The absolute path and name of the ZIP file on the local computer.

LSAF_PATH=*lsaf-path*
The path of the target container.

Optional Arguments

LSAF_ENABLEVERSIONING=0 (Default) | 1
Indicates whether the new files to create are versioned. See "LSAF_ENABLEVERSIONING=0 (Default) | 1" on page 6 for more information about this argument.

LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM
The case-insensitive type for a versioned file. See "LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM" on page 6 for more information about this argument.

LSAF_CUSTOMVERSION=*version*
If LSAF_VERSIONTYPE is CUSTOM, the specified version value. See "LSAF_CUSTOMVERSION=version" on page 6 for more information about this argument.

LSAF_COMMENT=*comment*
The comment to associate with the new file or file version.

Details

Uploads and expands the contents of a ZIP file into the specified container location in the repository.

Note: This macro is supported only from a PC SAS session. Calling this macro within an active SAS Life Science Analytics Framework session results in a failure.

Failure and Message Handling

The action status for each file that is uploaded and each folder that is created is printed in the SAS log file.

11

BatchWorkspaceService Module

Overview

Macros to perform batch operations in the workspace.

Note: These macros are supported only from a PC SAS session. Calling a macro from within an active SAS Life Science Analytics Framework session results in a failure.

Macro Summary Table

Category	Language Elements	Description
BatchWorkspace Service Module	%LSAF_DOWNLOADASZIPFROMWORKSPACE Macro (p. 142)	Downloads a workspace file or folder along with its content to a local computer as a ZIP file.
	%LSAF_UPLOADANDEXPANDINWORKSPACE Macro (p. 143)	Uploads and expands the contents of a ZIP file into the specified container location in the workspace.

Dictionary

%LSAF_DOWNLOADASZIPFROMWORKSPACE Macro

Downloads a workspace file or folder along with its content to a local computer as a ZIP file.

Category: BatchWorkspaceService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_DOWNLOADASZIPFROMWORKSPACE(LSAF_PATH=*lsaf-path*, LOCAL_PATH=*local-path*<, LSAF_OVERWRITE=1 | 0>);

Required Arguments

LSAF_PATH=*lsaf-path*
The path of the item to download.

LOCAL_PATH=*local-path*
The full path and name of the file to create (or overwrite, if applicable) on the local computer. File extensions are entered on the file as specified. If no extension is specified, the ZIP file has no extension.

Optional Argument

LSAF_OVERWRITE=1 | 0 (Default)
Indicates whether to overwrite the current ZIP file on the local computer. See “LSAF_OVERWRITE=0 (Default) | 1” on page 6 for more information about this argument.

Details

Downloads a workspace file or folder along with its content to a local computer as a ZIP file.

If the local directory does not exist, it is created. If the local file does not exist, it is created. If the item to download is a folder, the contents of the folder are downloaded and the hierarchy is maintained.

Note: This macro is supported only from a PC SAS session. Calling this macro within an active SAS Life Science Analytics Framework session results in a failure.

%LSAF_UPLOADANDEXPANDINWORKSPACE Macro

Uploads and expands the contents of a ZIP file into the specified container location in the workspace.

Category: BatchWorkspaceService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_UPLOADANDEXPANDINWORKSPACE(LSAF_PATH=*lsaf-path*, LOCAL_PATH=*local-path*);

Required Arguments

LSAF_PATH=*lsaf-path*
The path of the target container.

LOCAL_PATH=*local-path*
The absolute path and name of the ZIP file on the local computer.

Details

Uploads and expands the contents of a ZIP file into the specified container location in the workspace. Existing files are overwritten.

The action status for each file that is uploaded and each folder that is created is printed in the SAS log file.

Note: This macro is supported only from a PC SAS session. Calling this macro within an active SAS Life Science Analytics Framework session results in a failure.

12

RecycleBinService Module

Overview

Macros to retrieve and permanently delete items in the recycle bin.

Macro Summary Table

Category	Language Elements	Description
RecycleBinService Module	%LSAF_DELETERBITEMSBYQUERY Macro (p. 146)	Permanently deletes items from the recycle bin that match the specified criteria. The paths of the items that were permanently deleted are returned in a SAS data set.
	%LSAF_DELETERECYCLEBINITEMS Macro (p. 147)	Permanently deletes items from the recycle bin, using a SAS data set as input. Returns a SAS data set that contains the paths for the items that were permanently deleted from the recycle bin.
	%LSAF_GETRECYCLEBINITEMS Macro (p. 148)	Gets the metadata for the items in the recycle bin that match the specified search criteria. The metadata is stored in a SAS data set.

Dictionary

%LSAF_DELETERBITEMSBYQUERY Macro

Permanently deletes items from the recycle bin that match the specified criteria. The paths of the items that were permanently deleted are returned in a SAS data set.

Category: RecycleBinService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_DELETERBITEMSBYQUERY(<LSAF_LOCATION=*path*><, LSAF_NAME=*name*><, LSAF_DELETEDBY=*user-id*><, LSAF_FROM=*date*><, LSAF_TO=*date*><, LSAF_INCLUDESUBFOLDERS=0 | 1><, SAS_RESULT_DSNAME=*name*>);

Optional Arguments

LSAF_LOCATION=*path*
The path of the container that is the parent of the deleted items. Items from child containers are also returned.

LSAF_NAME=*name*
The name of the item to return by the search. If set to * (such as LSAF_NAME=*) or is omitted, items of any name are returned. Wildcards can be used. For example, LSAF_NAME=*.sas returns all of the items with a .sas extension.

LSAF_DELETEDBY=*user-id*
The identifier of the user who deleted the item. If omitted, set to blank, or * (such as LSAF_DELETEDBY=*), all items that the current user has the privilege to retrieve, regardless of who deleted the items, are returned. If the current user lacks the privileges, only current user items are returned.

LSAF_FROM=*date*
The beginning of the date range for items that were deleted within a specific time period. See “Format Date Values” on page 4 for information about date formats.

LSAF_TO=*date*
The end of the date range for items that were deleted within a specific time period. See “Format Date Values” on page 4 for information about date formats.

LSAF_INCLUDESUBFOLDERS=0 | 1 (Default)
Indicates whether the search includes containers within the specified location.

SAS_RESULT_DSNAME=*name*
The name of the output data set that contains the deleted paths. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFDELETERBITEMSBYQUERY.

The data set contains a row for each deleted path, sorted by path, and the following columns:

Column Name	Description
name	The name of the item.
path	The path of the item.

Details

Permanently deletes items from the recycle bin that match the specified criteria. At least one search criteria must be specified. The paths of the items that were permanently deleted are returned in a SAS data set.

To delete items put in the recycle bin by other users, you must have the Permanently Delete Items privilege.

%LSAF_DELETERECYCLEBINITEMS Macro

Permanently deletes items from the recycle bin, using a SAS data set as input. Returns a SAS data set that contains the paths for the items that were permanently deleted from the recycle bin.

Category: RecycleBinService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_DELETERECYCLEBINITEMS(SAS_DSNAME=name<, SAS_RESULT_DSNAME=name>);

Required Argument

SAS_DSNAME=name
The name of the data set that contains the identifiers for the items to delete from the recycle bin. See “Data Set Macros” on page 5 for information about specifying the name of the data set.

The data set must contain at least the following columns. Additional variables are ignored.

Column Name	Description
identifier	The unique identifier for the item.

Optional Argument

SAS_RESULT_DSNAME=name

The name of the output data set that contains the deleted paths. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFDELETERECYCLEBINITEMS.

The data set contains a row for each deleted path, sorted by path, and the following columns:

Column Name	Description
name	The name of the item.
path	The path of the item.

Details

Permanently deletes items from the recycle bin, using a SAS data set as input. Returns a SAS data set that contains the paths for the items that were permanently deleted from the recycle bin.

The items listed in the input data set are permanently removed from the recycle bin, provided that you have the Permanently Delete Items role privilege for the parent context of the deleted item. No error message is provided for items that cannot be permanently deleted.

A path that is created and deleted multiple times from the repository appear multiple times in the recycle bin. Therefore, the items must be referenced by their unique identifier. To get the identifier,call %LSAF_GETRECYCLEBINITEMS.

The output data set contains one row for each location that is permanently deleted from the recycle bin. If multiple items with the same path are deleted, the path is listed only once. The locations for each item that are deleted from a container (such as a folder) are listed separately in the output data set.

%LSAF_GETRECYCLEBINITEMS Macro

Gets the metadata for the items in the recycle bin that match the specified search criteria. The metadata is stored in a SAS data set.

Category: RecycleBinService Module

Note: This macro sets the standard macro variables and _lsafRBSearchLimitReached_, which is the search limit reached flag.

Syntax

%LSAF_GETRECYCLEBINITEMS(<LSAF_LOCATION=*path*><, LSAF_NAME=*name*><, LSAF_DELETEDBY=*user-id*><, LSAF_FROM=date><, LSAF_TO=*date*><, LSAF_INCLUDESUBFOLDERS=0 | 1><, SAS_DSNAME=*name*><, LSAF_PRINTLIMITREACHEDWARNING=0 (NOTE) | 1 (WARNING)>);

Optional Arguments

LSAF_LOCATION=*path*
The path of the container that is the parent of the deleted items. Items from child containers are also returned.

LSAF_NAME=*name*
The name of the item to return by the search. If set to * (such as LSAF_NAME=*) or is omitted, items of any name are returned. Wildcard notation can be used. For example, LSAF_NAME=*.sas returns all of the items with a .sas extension.

LSAF_DELETEDBY=*user-id*
The identifier of the user who deleted the item. If omitted, set to blank, or * (such as LSAF_DELETEDBY=*), all items that the current user has the privilege to retrieve, regardless of who deleted the items, are returned. If the current user lacks the privileges, only current user items are returned.

LSAF_FROM=*date*
The beginning of the date range for items deleted within a specific time period. See “Format Date Values” on page 4 for information about date formats.

LSAF_TO=*date*
The end of the date range for items that were deleted within a specific time period. See “Format Date Values” on page 4 for information about date formats.

LSAF_INCLUDESUBFOLDERS=0 | 1 (Default)
Indicates whether to include containers beyond the specified location.

SAS_RESULT_DSNAME=*name*
The name of the output data set that contains the metadata for the items. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETRECYCLEBINITEMS.

The data set contains a row for each item in the list, sorted by path, and the following columns:

Column Name	Description
name	The name of the item.
path	The path of the item.

Column Name	Description
type	Type of the item (such as folder or data set)
identifier	The unique identifier of the item.
size	The size (in bytes) of the item. The size is 0 for a container.
deletedBy	The identifier of the user who deleted the item.
deletedOn	The full date and time on which the item was deleted, represented as a String.
dateDeleted	The date on which the item was deleted, specified as a SAS Datetime format.
lastModifiedBy	The identifier of the user who last modified the item.
lastModifiedOn	The date on which the item was last modified, represented as a String.
dateLastModified	The date on which the item was last modified, represented as a SAS Datetime format.

Note: See “Format Date Values” on page 4 for information about date formats.

LSAF_PRINTLIMITREACHEDWARNING=0 (NOTE) | 1 (WARNING) (Default)
Indicates whether to print the message about the server query limit being reached as a WARNING or a NOTE.

Details

Gets the metadata for the items in the recycle bin that match the specified search criteria. The metadata is stored in a SAS data set.

Sets the Boolean macro variable _lsafRBSearchLimitReached_, which indicates whether the maximum limit on the number of items that can be returned by a query is reached. If the limit is reached, there might be more items in the recycle bin that meet the specified criteria. A more refined search is needed to retrieve those records. See “Boolean Macros” on page 4 for information about return values.

Users must have the Permanently Delete Items or Restore Items Deleted by Others role privilege at the specified location to retrieve items put in the recycle bin by other users.

13

RepositoryService Module

Overview

Macros to manage content in the repository.

Macro Summary Table

Category	Language Elements	Description
RepositoryService Module	%LSAF_CHECKIN Macro (p. 154)	Checks in a repository file.
	%LSAF_CHECKOUT Macro (p. 155)	Checks out a repository file.
	%LSAF_COPY Macro (p. 156)	Copies a file or folder to the specified location in the repository.
	%LSAF_CREATECONTEXT Macro (p. 156)	Creates a context in the repository.
	%LSAF_CREATEFILE Macro (p. 157)	Uploads a file from the local computer to the repository.
	%LSAF_CREATEFOLDER Macro (p. 158)	Creates a folder in the repository.
	%LSAF_DELETE Macro (p. 159)	Logically deletes an item. Deleted items are moved to the recycle bin.
	%LSAF_DELETEVERSION Macro (p. 159)	Permanently deletes the specified version of a repository item. If the version specified is the last version of the file, the macro fails.
	%LSAF_DISABLEVERSIONING Macro (p. 160)	Disables versioning for a repository file.
	%LSAF_DOWNLOADFILE Macro (p. 160)	Downloads a file to a local computer from the repository.

Category	Language Elements	Description
	%LSAF_ENABLEVERSIONING Macro (p. 161)	Enables versioning for a repository file.
	%LSAF_EXISTS Macro (p. 162)	Indicates whether a repository item exists.
	%LSAF_GETCHECKEDOUTFILES Macro (p. 162)	Gets the metadata for the files that are checked out of the repository. The information is stored in a SAS data set.
	%LSAF_GETCHILDREN Macro (p. 164)	Gets the metadata for the items that are within a container in the repository. The metadata is stored in a SAS data set.
	%LSAF_GETCONTEXTSTATE Macro (p. 167)	Gets the state value for the specified context in the repository.
	%LSAF_GETCOPYTOWORKSPACESTATUS Macro (p. 167)	Gets the value for the Copy to Workspace status of a repository item.
	%LSAF_GETITEMID Macro (p. 168)	Gets the identifier for a repository item.
	%LSAF_GETPARENTMEMBERSHIPCONTEXT Macro (p. 169)	Gets the nearest parent context with the membership capability for the specified path.
	%LSAF_GETPROPERTIES Macro (p. 169)	Gets the properties of an item in the repository. The properties are stored in a SAS data set.
	%LSAF_GETSYNCINFO Macro (p. 170)	Gets the synchronization information for a repository file. The information is stored in a SAS data set.
	%LSAF_GETTOPLEVELCONTEXTS Macro (p. 171)	Gets the metadata for the contexts at the top hierarchical structure in the repository to which you have Read access permission. The metadata is stored in a SAS data set.
	%LSAF_GETTYPE Macro (p. 173)	Gets the type of a repository item.
	%LSAF_GETVERSIONS Macro (p. 174)	Gets the metadata for all of the versions of a repository file. The metadata is stored in a SAS data set.
	%LSAF_ISCHECKEDOUT Macro (p. 175)	Indicates whether a repository file is checked out.
	%LSAF_ISCONTAINER Macro (p. 175)	Indicates whether a repository item is a container. Containers include folders and contexts.
	%LSAF_ISFILELOCKED Macro (p. 176)	Indicates whether a repository file is locked.
	%LSAF_ISVERSIONED Macro (p. 177)	Indicates whether a repository file is versioned.

Category	Language Elements	Description
	%LSAF_LOCKFILE Macro (p. 177)	Locks a repository file.
	%LSAF_MOVE Macro (p. 178)	Moves a repository file or folder to the specified target location in the repository.
	%LSAF_PERMANENTLYDELETE Macro (p. 178)	Permanently deletes a repository item. Deleted items are not moved to the recycle bin and cannot be restored.
	%LSAF_RENAME Macro (p. 179)	Renames an item in the repository.
	%LSAF_SEARCH Macro (p. 180)	Searches for repository items that meet the specified criteria.
	%LSAF_SYNCFILETOWORKSPACE Macro (p. 181)	Copies the version of the repository file to the same location in the workspace of the current user.
	%LSAF_UNDOCHECKOUT Macro (p. 182)	Undoes the checkout of a repository file.
	%LSAF_UNLOCKFILE Macro (p. 182)	Unlocks a repository file.
	%LSAF_UPDATECONTEXTSTATE Macro (p. 183)	Updates the state of a context in the repository.
	%LSAF_UPDATECOPYTOWORKSPACESTATUS Macro (p. 184)	Updates the value for the Copy To Workspace status of a repository item.
	%LSAF_UPDATEFILE Macro (p. 184)	Updates the contents of a file in the repository by uploading a new version of the file from the local computer.
	%LSAF_UPDATEPROPERTIES Macro (p. 185)	Updates the editable properties of a repository item using a SAS data set as input.
	%LSAF_UPDATEVERSIONLIMITS Macro (p. 186)	Sets the version limits on files and default version limits on repository containers.

Dictionary

%LSAF_CHECKIN Macro

Checks in a repository file.

Category: RepositoryService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_CHECKIN(LSAF_PATH=*path*<, LSAF_COMMENT=*comment*><, LSAF_ENABLEVERSIONINGFORNEWFILES=0 | 1><, LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM><, LSAF_CUSTOMVERSION=*version*>);

Required Argument

LSAF_PATH=*path*
The path of the file in the repository.

Optional Arguments

LSAF_COMMENT=*comment*
The comment to associate with the check in.

LSAF_ENABLEVERSIONINGFORNEWFILES=0 (Default) | 1
Indicates whether to enable versioning when checking in a new file to the repository. See “LSAF_ENABLEVERSIONING=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM
The version type to create by the check-in action. See “LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM” on page 6 for more information about this argument.

LSAF_CUSTOMVERSION=*version*
The specific version to create by the check-in action. See “LSAF_CUSTOMVERSION=version” on page 6 for more information about this argument.

%LSAF_CHECKOUT Macro

Checks out a repository file.

Category: RepositoryService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_CHECKOUT(LSAF_PATH=*path*);

Required Argument

LSAF_PATH=*path*
The path of the file in the repository.

%LSAF_COPY Macro

Copies a file or folder to the specified location in the repository.

Category: RepositoryService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_COPY(LSAF_SOURCE=*path*, LSAF_TARGET=*path*);

Required Arguments

LSAF_SOURCE=*path*
The path of the item.

LSAF_TARGET=*path*
The path (which includes the name, and, for files, the extension) for the copied item. The required parent folders, if they do not exist, are created for the copied file.

Details

Copies a file or folder to the specified location.

The copy of a versioned file is not versioned. The children of a copied folder are also copied to the new location and maintain their original directory structure.

%LSAF_CREATECONTEXT Macro

Creates a context in the repository.

Category: RepositoryService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_CREATECONTEXT(LSAF_PATH=*path*, LSAF_TYPEID=*type-id*);

Required Arguments

LSAF_PATH=*path*
The path of the context.

LSAF_TYPEID=*type-id*
The type identifier of the context. To determine the valid context types, call %LSAF_GETCONTEXTTYPES.

Details

Creates a context in the repository.

%LSAF_CREATEFILE Macro

Uploads a file from the local computer to the repository.

Category: RepositoryService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_CREATEFILE(LOCAL_PATH=*path*, LSAF_PATH=*path*<, LSAF_ENABLEVERSIONING=0 | 1><, LSAF_VERSION=*version*><, LSAF_COMMENT=*comment*>);

Required Arguments

LOCAL_PATH=*path*
The absolute path and name of the file on the local computer.

LSAF_PATH=*path*
The path and name of the file to create. Files can be created only in areas that support folder and file creation.

Optional Arguments

LSAF_ENABLEVERSIONING=0 (Default) | 1
Indicates whether the file to create is versioned. See “LSAF_ENABLEVERSIONING=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_VERSION=*version*
The version number to assign to the file. See “LSAF_VERSION=version” on page 6 for more information about this argument.

LSAF_COMMENT=*comment*
The audit history comment to associate with checking the versioned file into the repository.

Details

Uploads a file from the local computer to the repository.

All parent folders that do not exist are created.

Note: This macro is supported only from a PC SAS session. Calling this macro within an active SAS Life Science Analytics Framework session results in a failure.

%LSAF_CREATEFOLDER Macro

Creates a folder in the repository.

Category: RepositoryService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_CREATEFOLDER(LSAF_PATH=*path*);

Required Argument

LSAF_PATH=*path*
The path of the folder.

Details

Creates a folder in the repository.

All parent folders that do not exist are created.

%LSAF_DELETE Macro

Logically deletes an item. Deleted items are moved to the recycle bin.

Category: RepositoryService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_DELETE(LSAF_PATH=*path*);

Required Argument

LSAF_PATH=*path*
The path of the item.

Details

Logically deletes an item. Deleted items are moved to the recycle bin.

Note: Context items cannot be logically deleted. Attempting to delete a context results in a failure.

%LSAF_DELETEVERSION Macro

Permanently deletes the specified version of a repository item. If the version specified is the last version of the file, the macro fails.

Category: RepositoryService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_DELETEVERSION(LSAF_PATH=*path*, LSAF_VERSION=*version*);

Required Arguments

LSAF_PATH=*path*
The path of the repository item.

LSAF_VERSION=*version*
The version of the item. See “LSAF_VERSION=version” on page 6 for more information about this argument.

%LSAF_DISABLEVERSIONING Macro

Disables versioning for a repository file.

Category: RepositoryService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_DISABLEVERSIONING(LSAF_PATH=*path*<, LSAF_COMMENT=*comment*>);

Required Argument

LSAF_PATH=*path*
The path of the file.

Optional Argument

LSAF_COMMENT=*comment*
The comment to associate with the action. The comment is visible in the audit trail.

%LSAF_DOWNLOADFILE Macro

Downloads a file to a local computer from the repository.

Category: RepositoryService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_DOWNLOADFILE(LSAF_PATH=*path*, LOCAL_PATH=*path*<, LSAF_VERSION=*version*>);

Required Arguments

LSAF_PATH=*path*
The path of the file in the repository.

LOCAL_PATH=*path*
The full path and name of the file on the local computer.

Optional Argument

LSAF_VERSION=*version*
The version of the file to download. See “LSAF_VERSION=version” on page 6 for more information about this argument.

Details

Downloads a file to a local computer from the repository.

If the local folder does not exist, it is created. If the local file exists, it is overwritten.

Note: This macro is supported only from a PC SAS session. Calling this macro within an active SAS Life Science Analytics Framework session results in a failure.

%LSAF_ENABLEVERSIONING Macro

Enables versioning for a repository file.

Category: RepositoryService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_ENABLEVERSIONING(LSAF_PATH=*path*<, LSAF_VERSION=*version*><, LSAF_COMMENT=*comment*>);

Required Argument

LSAF_PATH=*path*
The path of the file in the repository.

Optional Arguments

LSAF_VERSION=*version*
The starting version number of the file. See “LSAF_VERSION=version” on page 6 for more information about this argument.

LSAF_COMMENT=*comment*
The comment to associate with the action. This value appears in the audit trail.

%LSAF_EXISTS Macro

Indicates whether a repository item exists.

Category: RepositoryService Module

Note: This macro sets the standard macro variables and _lsafExists_, which indicates whether the item exists.

Syntax

%LSAF_EXISTS(LSAF_PATH=*path*);

Required Argument

LSAF_PATH=*path*
The path of the item.

Details

Sets the Boolean macro variable _lsafExists_, which indicates whether a repository item exists. See “Boolean Macros” on page 4 for information about return values.

%LSAF_GETCHECKEDOUTFILES Macro

Gets the metadata for the files that are checked out of the repository. The information is stored in a SAS data set.

Category: RepositoryService Module

Note: This macro sets the standard macro variables and _lsafCOFSearchLimitReached_, which is the search limit reached flag.

Syntax

%LSAF_GETCHECKEDOUTFILES(<LSAF_CHECKEDOUTBY=*user-id*><, LSAF_NAME=*name*><, LSAF_LOCATION=*path*><, SAS_DSNAME=*name*><, LSAF_PRINTLIMITREACHEDWARNING=0 (NOTE) | 1 (WARNING)>);

Optional Arguments

LSAF_CHECKEDOUTBY=*user-id*
The identifier of the user who checked out the files. If set to * (such as LSAF_CHECKEDOUTBY=*) or omitted, the files that are checked out by any user are returned. Wildcards can be used. For example, LSAF_CHECKEDOUTBY=ge* returns the files that are checked out by users whose account name starts with ge.

LSAF_NAME=*name*
The name of the file. If set to * (such as LSAF_NAME=*) or omitted, all files are returned. Wildcards can be used. For example, LSAF_NAME=*.sas returns all file names with a .sas extension.

LSAF_LOCATION=*path*
The path of the container that is the parent of the checked out file. If set to * (such as LSAF_LOCATION=*) or omitted, files from any container are returned. Wildcards can be used. For example, LSAF_LOCATION=/SAS/File* returns all information with a file path that starts with /SAS/File.

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the checked out files. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETCHECKEDOUTFILES.

The output data set contains a single row for each checked out file and contains the following columns:

Column Name	Description
name	The name of the file.
location	The location (parent container) of the checked out file.
size	The size (in bytes) of the file.
formattedsize	The size of the file, expressed in units such as KB or MB.
checkedoutby	The user who checked out the file.

Column Name	Description
modifieddate	The date on which the file was last modified. See “Format Date Values” on page 4 for information about date formats.
version	The current version of the file that is checked out.
status	The workspace status of the file. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=CheckoutStatus`.

LSAF_PRINTLIMITREACHEDWARNING=0 (NOTE) | 1 (WARNING) (Default)
Indicates whether to print the message about the server query limit being reached as a WARNING or a NOTE.

This argument applies only when the _lsafCOFSearchLimitReached_ macro variable is 1.

Details

Gets the checked out file information for the specified user, file, or location within the repository. The information is stored in a SAS data set.

This macro requires that the logged on user have the administrative privilege PRIVILEGE_MANAGE_CHECKEDOUT_FILES.

Sets the Boolean macro variable _lsafCOFSearchLimitReached_, which indicates whether the maximum limit on the number of items that can be returned by a query is reached. If the limit is reached, there might be checked out files in the repository that meet the specified search criteria. A more refined search is needed to retrieve those records. See “Boolean Macros” on page 4 for information about return values.

%LSAF_GETCHILDREN Macro

Gets the metadata for the items that are within a container in the repository. The metadata is stored in a SAS data set.

Category: RepositoryService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETCHILDREN(LSAF_PATH=*path*<, SAS_DSNAME=*name*><, LSAF_RECURSIVE=*level*>);

Required Argument

LSAF_PATH=*path*
The path of the container.

Optional Arguments

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the items. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETCHILDREN.

The data set contains a row for each item in the list, sorted by path, and the following columns:

Column Name	Description
path	The path of the item.
name	The name of the item.
itemType	The type of the item (such as folder or data set).
isContainer	Indicates whether the item is a container.
description	The description of the item.
version	The version that is associated with this specific item.
isVersioned	Indicates whether the item is currently versioned.
isCheckedOut	Indicates whether the item is currently checked out.
isLocked	Indicates whether the item is currently locked.
signatureStatus	Indicates whether the item has electronic signatures attached. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=SigningStatus`.

Column Name	Description
size	The size (in bytes) of the item. This value is set to missing (.) for containers.
formattedSize	The formatted size (in bytes, KB, MB, or TB).
createdBy	The identifier of the user who created the item.
created	The full date and time on which the item was created, represented as a String.
dateCreated	The date on which the item was created, represented as a SAS datetime format.
lastModifiedBy	The identifier of the user who last modified the item.
lastModified	The date on which the item was last modified, represented as a String.
dateLastModified	The date on which the item was last modified, represented as a SAS datetime format.
propertiesLastModifiedBy	The identifier of the user who last modified the properties of the item.
propertiesLastModified	The date on which the item properties were last modified, represented as a String.
datePropertiesLastModified	The date on which the item properties were last modified, represented as a SAS datetime format.
state	If the item is a context, the state of the item. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=State`.

Note: See “Format Date Values” on page 4 for information about date formats.

LSAF_RECURSIVE=*level*

The level of recursion. Valid values:

- 0 (Default): Returns the container only.

- 1 - 99: (inclusive) Returns the container and traverses the tree with the number of levels indicated.

Details

Gets the metadata for the items that are within a container in the repository. The metadata is stored in a SAS data set.

To retrieve the children of the root level, call %LSAF_GETTOPLEVELCONTEXTS.

%LSAF_GETCONTEXTSTATE Macro

Gets the state value for the specified context in the repository.

Category: RepositoryService Module

Note: This macro sets the standard macro variables and _lsafContextState_, which is the state of the context.

Syntax

%LSAF_GETCONTEXTSTATE(LSAF_PATH=*path*);

Required Argument

LSAF_PATH=*path*
The path to the context.

Details

Sets the value of the macro variable _lsafContextState_ to the state of the context. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=State`. See “Value Macros” on page 5 for information about value macros.

%LSAF_GETCOPYTOWORKSPACESTATUS Macro

Gets the value for the Copy to Workspace status of a repository item.

Category: RepositoryService Module

Note: This macro sets the standard macro variables. and _lsafCopyToWSStatus_, which is the Copy to Workspace status.

Syntax

%LSAF_GETCOPYTOWORKSPACESTATUS(LSAF_PATH=*path*);

Required Argument

LSAF_PATH=*path*
The path of the repository file.

Details

Sets the value of the macro variable _lsafCopyToWSStatus_ to the Copy to Workspace status of a repository item. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=Syncable`. See "Value Macros" on page 5 for information about value macros.

%LSAF_GETITEMID Macro

Gets the identifier for a repository item.

Category: RepositoryService Module

Note: This macro sets the standard macro variables and _lsafItemId_, which is the item ID.

Syntax

%LSAF_GETITEMID(LSAF_PATH=*path*);

Required Argument

LSAF_PATH=*path*
The path of the item.

Details

Sets the value of the macro variable _lsafItemId_ to the identifier for a repository item. You must have Read access permission to the item. See "Value Macros" on page 5 for information about value macros.

%LSAF_GETPARENTMEMBERSHIPCONTEXT Macro

Gets the nearest parent context with the membership capability for the specified path.

Category: RepositoryService Module

Note: This macro sets the standard macro variables and _lsafParentMembershipContext_, which is the parent membership context.

Syntax

%LSAF_GETPARENTMEMBERSHIPCONTEXT(LSAF_PATH=*path*);

Required Argument

LSAF_PATH=*path*
The path to the repository item.

Details

Gets the nearest parent context with the membership capability for the specified path and sets the value of the macro variable _lsafParentMembershipContext_ to the path of the parent membership context value. See “Value Macros” on page 5 for information about value macros.

%LSAF_GETPROPERTIES Macro

Gets the properties of an item in the repository. The properties are stored in a SAS data set.

Category: RepositoryService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETPROPERTIES(LSAF_PATH=*path*<, LSAF_VERSION=*version*><, SAS_DSNAME=*name*>);

Required Argument

LSAF_PATH=*path*
The path of the item in the repository.

Optional Arguments

LSAF_VERSION=*version*
The specific file version. See “LSAF_VERSION=version” on page 6 for more information about this argument.

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the item. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETPROPERTIES.

The data set contains the following columns that represent each element of metadata for the item, sorted by the NAME variable:

Column Name	Description
path	The path of the item
name	The name of the property.
value	The value of the property.
displayName	The name of the property as displayed as a label within SAS Life Science Analytics Framework.
type	The type of the property. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=AttributeType`.
isReadonly	Indicates whether the property cannot be modified.
isExtended	Indicates whether the property is an extended attribute.

Note: Extended attributes are included, but the names represent the system identifier. The recognizable label is shown in the display name variable.

%LSAF_GETSYNCINFO Macro

Gets the synchronization information for a repository file. The information is stored in a SAS data set.

Category: RepositoryService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETSYNCINFO(LSAF_PATH=*path*<, SAS_DSNAME=*name*>);

Required Argument

LSAF_PATH=*path*
The path of the file.

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the synchronization information. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETSYNCINFO.

The data set contains the following variables that represent each element of metadata for the file:

Column Name	Description
path	The path of the item.
name	The name of the property.
value	The value of the property.
description	The description of what the property represents.

%LSAF_GETTOPLEVELCONTEXTS Macro

Gets the metadata for the contexts at the top hierarchical structure in the repository to which you have Read access permission. The metadata is stored in a SAS data set.

Category: RepositoryService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETTOPLEVELCONTEXTS(<SAS_DSNAME=*name*>);

Optional Argument

SAS_DSNAME=*name*

The name of the output data set that contains the metadata for the contexts. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETTOPLEVELCONTEXTS.

The data set contains a row for each context in the list, sorted by path, and the following columns:

Column Name	Description
name	The name of the item.
path	The path of the item.
name	The name of the item.
itemType	Type of the item (such as sas:organization or businessunit).
description	The description of the item.
createdBy	The identifier of the user who created the item.
created	The full date and time on which the item was created, represented as a String.
dateCreated	The date on which the item was created, represented as a SAS datetime format.
lastModifiedBy	The identifier of the user who last modified the item.
lastModified	The date on which the item was last modified, represented as a String.
dateLastModified	The date on which the item was last modified, represented as a SAS datetime format.
propertiesLastModifiedBy	The identifier of the user who last modified the properties of the item.

Column Name	Description
propertiesLastModified	The date on which the item properties were last modified, represented as a String.
datePropertiesLastModified	The date on which the item properties were last modified, represented as a SAS datetime format.
state	The state of the item. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=State`.

Note: See "Format Date Values" on page 4 for information about date formats.

%LSAF_GETTYPE Macro

Gets the type of a repository item.

Category: RepositoryService Module

Note: This macro sets the standard macro variables and _lsafType_, which is the item type.

Syntax

%LSAF_GETTYPE(LSAF_PATH=*path*);

Required Argument

LSAF_PATH=*path*
The path of the item.

Details

Sets the value of the macro variable _lsafType_ to the repository item type. Any of the valid context types can also be returned as possible item type values. See "Value Macros" on page 5 for information about value macros.

%LSAF_GETVERSIONS Macro

Gets the metadata for all of the versions of a repository file. The metadata is stored in a SAS data set.

Category: RepositoryService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETVERSIONS(LSAF_PATH=*path*<, SAS_DSNAME=*name*>);

Required Argument

LSAF_PATH=*path*
The path of the item in the repository.

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the versions. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETVERSIONS.

The data set contains a row for each version of the item and columns with the following names, sorted by descending version:

Column Name	Description
path	The path of the item.
version	The version that is associated with this specific item.
isSigned	Indicates whether the item has electronic signatures attached to it.
isLatest	Indicates whether the item is the latest version.
size	The size (in bytes) of the item. This is 0 for a container.
createdBy	The user identifier that created the item.
created	The full date and time on which the item was created, represented as a String.

Column Name	Description
dateCreated	The date on which the item was created, represented as a SAS datetime format.
comment	The comment that was added when the version was created.

Note: See "Format Date Values" on page 4 for information about date formats.

%LSAF_ISCHECKEDOUT Macro

Indicates whether a repository file is checked out.

Category: RepositoryService Module

Note: This macro sets the standard macro variables and _lsafIsCheckedOut_, which indicates whether the file is checked out.

Syntax

%LSAF_ISCHECKEDOUT(LSAF_PATH=*path*);

Required Argument

LSAF_PATH=*path*
The path of the file.

Details

Sets the Boolean macro variable _lsafIsCheckedOut_, which indicates whether a repository file is checked out. See "Boolean Macros" on page 4 for information about return values.

%LSAF_ISCONTAINER Macro

Indicates whether a repository item is a container. Containers include folders and contexts.

Category: RepositoryService Module

Note: This macro sets the standard macro variables and _lsafIsContainer_,which indicates whether the item is a container.

Syntax

%LSAF_ISCONTAINER(LSAF_PATH=*path*);

Required Argument

LSAF_PATH=*path*
The path of the item in the repository.

Details

Sets the Boolean macro variable _lsafIsContainer_, which indicates whether a repository item is a container. Containers include folders and contexts. See “Boolean Macros” on page 4 for information about return values.

%LSAF_ISFILELOCKED Macro

Indicates whether a repository file is locked.

Category: RepositoryService Module

Note: This macro sets the standard macro variables and _lsafIsFileLocked_, which indicates whether the file is locked.

Syntax

%LSAF_ISFILELOCKED(LSAF_PATH=*path*);

Required Argument

LSAF_PATH=*path*
The path of the file.

Details

Sets the Boolean macro variable _lsafIsFileLocked_, which indicates whether a repository file is locked. See “Boolean Macros” on page 4 for information about return values.

%LSAF_ISVERSIONED Macro

Indicates whether a repository file is versioned.

Category: RepositoryService Module

Note: This macro sets the standard macro variables and _lsafIsVersioned_, which indicates whether the file is versioned.

Syntax

%LSAF_ISVERSIONED(LSAF_PATH=*path*);

Required Argument

LSAF_PATH=*path*
The path to the file.

Details

Sets the Boolean macro variable _lsafIsVersioned_, which indicates whether a repository file is versioned. See “Boolean Macros” on page 4 for information about return values.

%LSAF_LOCKFILE Macro

Locks a repository file.

Category: RepositoryService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_LOCKFILE(LSAF_PATH=*path*);

Required Argument

LSAF_PATH=*path*
The path of the file.

Details

Locks a repository file.

You must have the Manage Locking role privilege.

%LSAF_MOVE Macro

Moves a repository file or folder to the specified target location in the repository.

Category: RepositoryService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_MOVE(LSAF_SOURCE=*path*, LSAF_TARGET=*path*);

Required Arguments

LSAF_SOURCE=*path*
The path of the file or folder.

LSAF_TARGET=*path*
The full path (which includes the name and file extension) for the moved file or folder. The target location must exist. The path cannot end with a slash (/).

Details

Moves a repository file or folder to the specified target location in the repository. Contexts cannot be moved.

The item to move must not exist in the target location. The parent folders for the target item must exist. The children of a moved folder are also moved to the target location, keeping their directory structure and version history (if any) intact.

%LSAF_PERMANENTLYDELETE Macro

Permanently deletes a repository item. Deleted items are not moved to the recycle bin and cannot be restored.

Category: RepositoryService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_PERMANENTLYDELETE(LSAF_PATH=*path*);

Required Argument

LSAF_PATH=*path*
The path of the item.

Details

Permanently deletes a repository item. Deleted items are not moved to the recycle bin and cannot be restored.

To move an item to the recycle bin, call %LSAF_DELETE.

%LSAF_RENAME Macro

Renames an item in the repository.

Category: RepositoryService Module

Note: This macro sets the standard macro variables and _lsafRenamedItem_, which is the full path of the renamed item.

Syntax

%LSAF_RENAME(LSAF_SOURCE=*path*, LSAF_NEWNAME=*name*);

Required Arguments

LSAF_SOURCE=*path*
The full path of the item to be renamed.

LSAF_NEWNAME=*name*
The new name.

Details

Renames an item in the repository and sets the value of the macro variable _lsafRenamedItem_ to the full path of the renamed item. The new name for a file must include the file extension, otherwise the renaming occurs without the file having an extension. See “Value Macros” on page 5 for information about value macros.

%LSAF_SEARCH Macro

Searches for repository items that meet the specified criteria.

Category: RepositoryService Module

Note: This macro sets the standard macro variables and _lsafSearchLimitReached_, which is the search limit reached flag.

Syntax

%LSAF_SEARCH(<LSAF_LOCATION=*path*><, LSAF_NAME=*name*><, LSAF_TYPEID=*type-id*><, LSAF_OWNER=*user-id*><, LSAF_LASTMODIFIEDBY=*user-id*><, LSAF_CREATEDBY=*user-id*><, LSAF_FROMCREATED=*date*><, LSAF_TOCREATED=*date*><, LSAF_FROMLASTMODIFIED=*date*><, LSAF_TOLASTMODIFIED=*date*><, LSAF_CONTENT=*content*><, LSAF_COPYTOWORKSPACESTATUS=*syncable*><, SAS_DSNAME=*name*><, LSAF_PRINTLIMITREACHEDWARNING=0 (NOTE) | 1 (WARNING)>);

Optional Arguments

LSAF_LOCATION=*path*
The path of the parent container in which to search. Child containers and their content are also returned.

LSAF_NAME=*name*
The name of the item.

LSAF_TYPEID=*type-id*
The type of the item. To get the identifiers, call %LSAF_GETSEARCHABLETYPES.

LSAF_OWNER=*user-id*
The identifier of the user who owns the item.

LSAF_LASTMODIFIEDBY=*user-id*
The identifier of the user who last modified the item.

LSAF_CREATEDBY=*user-id*
The identifier of the user who created the item.

LSAF_FROMCREATED=*date*
The earliest date on which the item was created. See “Format Date Values” on page 4 for information about date formats.

LSAF_TOCREATED=*date*
The latest date on which the item was created. See “Format Date Values” on page 4 for information about date formats.

LSAF_FROMLASTMODIFIED=*date*
The earliest date on which the item was last modified. See “Format Date Values” on page 4 for information about date formats.

LSAF_TOLASTMODIFIED=*date*
The latest date on which the item was last modified. See “Format Date Values” on page 4 for information about date formats.

LSAF_CONTENT=*content*
The text that is expected to appear in the file. Because job files contain XML content, the search finds only text that is in attribute values.

LSAF_COPYTOWORKSPACESTATUS=*syncable*
The status of the Copy to Workspace property. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=Syncable`.

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the items. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFSEARCH.

The data set contains a row for each item in the list, sorted by path.For more information, see the table for %LSAF_GETCHILDREN on page 165.

LSAF_PRINTLIMITREACHEDWARNING=0 (NOTE) | 1 (WARNING) (Default)
Indicates whether to write to the SAS log file the message about the server query limit being reached as a WARNING or a NOTE.

Details

Searches for repository items that meet the specified criteria. Although all arguments are optional, at least one argument value must be specified to start a search.

A search is not case sensitive. For example, “AAA” finds items with “aaa”.

Sets the Boolean macro variable _lsafSearchLimitReached_, which indicates whether the maximum limit on the number of items that can be returned by a query is reached. If the limit is reached, there might be additional items in the repository that meet the search criteria. A more refined search is needed to retrieve those records. See “Boolean Macros” on page 4 for information about return values.

With the exception of the location and datetime parameters, wildcards can be used in the search values. For example:

- LSAF_NAME=user* searches for an occurrence that starts with 'user'.
- LSAF_NAME=*use* searches for an occurrence that contains 'use'.
- LSAF_NAME=us?r searches for an occurrence that starts with 'us', then any 1 character, and then ending in 'r'.

See “Format Date Values” on page 4 for information about date formats.

%LSAF_SYNCFILETOWORKSPACE Macro

Copies the version of the repository file to the same location in the workspace of the current user.

Category: RepositoryService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_SYNCFILETOWORKSPACE(LSAF_PATH=*path*<, LSAF_VERSION=*version*>);

Required Argument

LSAF_PATH=*path*
The path of the repository file.

Optional Argument

LSAF_VERSION=*version*
The version of the file to synchronize. See “LSAF_VERSION=version” on page 6 for more information about this argument.

%LSAF_UNDOCHECKOUT Macro

Undoes the checkout of a repository file.

Category: RepositoryService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_UNDOCHECKOUT(LSAF_PATH=*path*);

Required Argument

LSAF_PATH=*path*
The path of the file.

%LSAF_UNLOCKFILE Macro

Unlocks a repository file.

Category: RepositoryService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_UNLOCKFILE(LSAF_PATH=*path*);

Required Argument

LSAF_PATH=*path*
The path of the file.

Details

Unlocks a repository file. You must have the role privilege Manage Locking.

%LSAF_UPDATECONTEXTSTATE Macro

Updates the state of a context in the repository.

Category: RepositoryService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_UPDATECONTEXTSTATE(LSAF_PATH=*path*, LSAF_STATE=*state*<, LSAF_COMMENT=*comment*><, LSAF_RECURSIVE=0 | 1>);

Required Arguments

LSAF_PATH=*path*
The path of the context.

LSAF_STATE=*state*
The state to set. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=State`.

Optional Arguments

LSAF_COMMENT=*comment*
The reason for the state change.

LSAF_RECURSIVE=0 (Default) | 1
Indicates whether the state of the child contexts, if any exist, are updated. This argument applies only when reactivating a context. The behavior to close a context is to automatically close all children.

Details

Updates the state of a context in the repository. If the context is already in the specified state, no error is generated.

To determine whether the state of a context can change, the capabilities column from the output data set that is created by calling %LSAF_GETCONTEXTTYPES contains the value STATE.

%LSAF_UPDATECOPYTOWORKSPACESTATUS Macro

Updates the value for the Copy To Workspace status of a repository item.

Category: RepositoryService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_UPDATECOPYTOWORKSPACESTATUS(LSAF_PATH=*path*<, LSAF_STATUS=*status*>);

Required Argument

LSAF_PATH=*path*
The path to the file.

Optional Argument

LSAF_STATUS=*status*
The Copy To Workspace status. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=Syncable`.

%LSAF_UPDATEFILE Macro

Updates the contents of a file in the repository by uploading a new version of the file from the local computer.

Category: RepositoryService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_UPDATEFILE(LOCAL_PATH=*path*, LSAF_PATH=*path*<, LSAF_COMMENT=*comment*><, LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM><, LSAF_CUSTOMVERSION=*version*>);

Required Arguments

LOCAL_PATH=*path*
The absolute path to the file on the local computer.

LSAF_PATH=*path*
The path of the file in the repository to update. The file must exist and must not be a container.

Optional Arguments

LSAF_COMMENT=*comment*
The comment to associate with updating the file.

LSAF_VERSIONTYPE=MINOR | MAJOR | CUSTOM
If the file to update is versioned, the type of version to create. See "LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM" on page 6 for more information about this argument.

LSAF_CUSTOMVERSION=*version*
The specific version to create. See "LSAF_CUSTOMVERSION=version" on page 6 for more information about this argument.

Details

Updates the contents of a file in the repository by uploading a new version of the file from the local computer.

If the file is versioned, a new version is created. If the file is not versioned, the current contents are overwritten.

Note: This macro is supported only from a PC SAS session. Calling this macro within an active SAS Life Science Analytics Framework session results in a failure.

%LSAF_UPDATEPROPERTIES Macro

Updates the editable properties of a repository item using a SAS data set as input.

Category: RepositoryService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_UPDATEPROPERTIES(LSAF_PATH=*path*, SAS_DSNAME=*name*);

Required Arguments

LSAF_PATH=*path*
The path of the repository item.

SAS_DSNAME=*name*
The name of the data set that contains the property values to set on the repository item. See "Data Set Macros" on page 5 for information about specifying the name of the data set.

The data set must contain at least the following columns. Additional variables are ignored.

Column Name	Description
name	The name of the property.
value	The value.

Details

Updates the editable properties of a repository item using a SAS data set as input.

The output data set from this macro contains the necessary variables and data needed as the input data set for the %LSAF_UPDATEPROPERTIES macro. The column isReadonly indicates whether the field can be updated: 0 permits updates, 1 does not. Modify the data set with changes prior to calling this macro.

- Data in the input data set that do not correspond to a property of the item being updated are ignored.
- Values that are specified for properties that are not editable are ignored.
- An input data value that is the same as the current property value is ignored and does not update the property. No audit record is created for the property.

Not all properties can be updated by calling %LSAF_UPDATEPROPERTIES.

- Call %LSAF_UPDATECONTEXTSTATE to update a context state.
- Call %LSAF_UPDATECOPYTOWORKSPACESTATUS to update the Copy to Workspace property.

%LSAF_UPDATEVERSIONLIMITS Macro

Sets the version limits on files and default version limits on repository containers.

Category: RepositoryService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_UPDATEVERSIONLIMITS(LSAF_PATH=*path*<, LSAF_VERSIONMAJOR=*version*><, LSAF_VERSIONMINOR=*version*>);

Required Argument

LSAF_PATH=*path*
The path of the repository item.

Optional Arguments

LSAF_VERSIONMAJOR=*version*
The major version limit. If this argument is omitted, the limit is not modified. If set to null (such as LSAF_VERSIONMAJOR=), the version limit is to No Limit. Valid values are 1 through 10.

LSAF_VERSIONMINOR=*version*
The minor version limit. If this argument is omitted, the limit is not modified. If set to null (such as LSAF_VERSIONMINOR=), the version limit is to No Limit. Valid values are 1 through 10.

Details

Sets the version limits on files and default version limits on repository containers.

The %LSAF_GETPROPERTIES macro returns the values for the minor and major version limits of an object.

14

ResourceUtilizationService Module

Overview

Macros to get information about resource utilization.

The macros in this service collect and perform calculations on potentially large amounts of information. The operation might take several minutes. For macros that take a recursion argument, it is advised that the path and recursion be limited to avoid time-out issues.

Macro Summary Table

Category	Language Elements	Description
ResourceUtilization Module	%LSAF_GETADMINREPOUTILIZATION Macro (p. 190)	Gets the dashboard resource utilization information for child containers with the calculated sizes for the repository.
	%LSAF_GETADMINTRANSUTILIZATION Macro (p. 192)	Gets the dashboard summary utilization information for each user's existing transient workspaces with existing sessions, if any. You must have the View Administrative Dashboard Tiles user account privilege to run this macro.
	%LSAF_GETADMINWSUTILIZATION Macro (p. 192)	Gets the dashboard summary utilization information for each user's workspace. You must have the View Administrative Dashboard Tiles user account privilege to run this macro.
	%LSAF_GETRESOURCEUTILIZATION Macro (p. 193)	Gets the dashboard resource utilization summary data.
	%LSAF_GETUSERREPOUTILIZATION Macro (p. 194)	Gets the dashboard resource utilization information for child containers with calculated sizes for the current user's repository /Users folder.
	%LSAF_GETUSERTRANSUTILIZATION Macro (p. 195)	Gets the dashboard utilization information for the current user's transient workspaces with sessions, if any exist.
	%LSAF_GETUSERWSUTILIZATION Macro (p. 196)	Gets the dashboard resource utilization information for child folders with calculated sizes for the current user's workspace.

Dictionary

%LSAF_GETADMINREPOUTILIZATION Macro

Gets the dashboard resource utilization information for child containers with the calculated sizes for the repository.

Category: ResourceUtilization Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETADMINREPOUTILIZATION(<LSAF_PATH=*path*><, LSAF_RECURSIVE=*number*><, SAS_DSNAME=*name*>);

Optional Arguments

LSAF_PATH=*path*
The full path of the repository container. If not specified, the top level (/) is used.

LSAF_RECURSIVE=*number*
The number of levels of subfolders to recurse. Valid values:

- 1 (Default): The top-level folders.
- 2 - 99: The levels of subfolders.

SAS_DSNAME=*name*
The name of the output data set that contains the resource utilization data. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETADMINREPOUTILIZATION.

The data set contains a row for each summary data value, sorted by path, and the following columns:

Column Name	Description
type	The type of the container.
path	The full path to the container.
name	The name of the container.
size	The numeric total size (in bytes) of the folder, formatted with SAS format SIZEKMG.

Details

Gets the dashboard resource utilization information for child containers with the calculated sizes for the repository. System (hidden) folders, which include the recycle bin, are ignored.

You must have the View Administrative Dashboard Tiles user account privilege to run this macro.

%LSAF_GETADMINTRANSUTILIZATION Macro

Gets the dashboard summary utilization information for each user's existing transient workspaces with existing sessions, if any. You must have the View Administrative Dashboard Tiles user account privilege to run this macro.

Category: ResourceUtilization Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETADMINTRANSUTILIZATION(< SAS_DSNAME=*name*>);

Optional Argument

SAS_DSNAME=*name*

The name of the output data set that contains the resource utilization data. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETADMINTRANSUTILIZATION.

The data set contains a row for each summary data value, sorted by descending size, and the following columns:

Column Name	Description
owner	The user identifier for the owner of the transient workspace.
count	The number of transient workspaces owned by the user.
size	The numeric total size (in bytes) of the user's transient workspace, formatted with SAS format SIZEKMG.

%LSAF_GETADMINWSUTILIZATION Macro

Gets the dashboard summary utilization information for each user's workspace. You must have the View Administrative Dashboard Tiles user account privilege to run this macro.

Category: ResourceUtilization Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETADMINWSUTILIZATION(< SAS_DSNAME=*name*>);

Optional Argument

SAS_DSNAME=*name*

The name of the output data set that contains the resource utilization data. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETADMINWSUTILIZATION.

The data set contains a row for each summary data value, sorted by descending size, and the following columns:

Column Name	Description
owner	The user identifier for the owner of the workspace.
size	The numeric total size (in bytes) of the user's workspace, formatted with SAS format SIZEKMG.

%LSAF_GETRESOURCEUTILIZATION Macro

Gets the dashboard resource utilization summary data.

Category: ResourceUtilization Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETRESOURCEUTILIZATION(< SAS_DSNAME=*name*>);

Optional Argument

SAS_DSNAME=*name*

The name of the output data set that contains the resource utilization data. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETRESOURCEUTILIZATION.

The data set contains a row for each summary data value and columns with the following names:

Column Name	Description
type	The data type. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=UtilizationType`.
workarea	The work area referenced by the data. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=UtilizationArea`.
size	The numeric size (in bytes) of the data, formatted with SAS format SIZEKMG.

Details

Gets the dashboard resource utilization summary data. If you do not have the user account privilege View Administrative Dashboard Tiles only the data for the current user's workspace and transient workspace are returned.

This macro collects multiple pieces of information and might require several seconds to run.

%LSAF_GETUSERREPOUTILIZATION Macro

Gets the dashboard resource utilization information for child containers with calculated sizes for the current user's repository /Users folder.

Category: ResourceUtilization Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETUSERREPOUTILIZATION(<LSAF_PATH=*path*><, LSAF_RECURSIVE=*number*><, SAS_DSNAME=*name*>);

Optional Arguments

LSAF_PATH=*path*
The full path of the repository container. If not specified, the top level (`/Users/currentUserId`) is used.

LSAF_RECURSIVE=*number*
The number of levels of subfolders to recurse. Valid values:

- 1 (Default): The top-level folders.

- 2 - 99: The levels of subfolders.

SAS_DSNAME=*name*
The name of the output data set that contains the resource utilization data. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETUSERREPOUTILIZATION.

The data set contains a row for each summary data value, sorted by path, and the following columns:

Column Name	Description
path	The full path to the container.
name	The name of the container.
size	The numeric total size (in bytes) of the folder, formatted with SAS format SIZEKMG.

Details

Gets the dashboard resource utilization information for child containers with calculated sizes for the current user's repository /Users folder. System (hidden) folders, which include the recycle bin, are ignored.

%LSAF_GETUSERTRANSUTILIZATION Macro

Gets the dashboard utilization information for the current user's transient workspaces with sessions, if any exist.

Category: ResourceUtilization Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETUSERTRANSUTILIZATION(< SAS_DSNAME=*name*>);

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the resource utilization data. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETUSERTRANSUTILIZATION.

The data set contains a row for each summary data value, sorted by descending size, and the following columns:

Column Name	Description
owner	The user identifier for the owner of the transient workspace.
sessionState	The state of the session.
clientId	The incremental number assigned to the session.
sessionLabel	The label assigned to the session.
size	The numeric total size (in bytes) of the session, formatted with SAS format SIZEKMG.

%LSAF_GETUSERWSUTILIZATION Macro

Gets the dashboard resource utilization information for child folders with calculated sizes for the current user's workspace.

Category: ResourceUtilization Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETUSERWSUTILIZATION(<LSAF_PATH=*path*><, LSAF_RECURSIVE=*number*><, SAS_DSNAME=*name*>);

Optional Arguments

LSAF_PATH=*path*
The full path of the workspace container. If not specified, the top level (/) is used.

LSAF_RECURSIVE=*number*
The number of levels of subfolders to recurse. Valid values:

- 1 (Default): The top-level folders.
- 2 - 99: The levels of subfolders.

SAS_DSNAME=*name*
The name of the output data set that contains the resource utilization data. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETUSERWSUTILIZATION.

The data set contains a row for each summary data value, sorted by path, and the following columns:

Column Name	Description
path	The full path of the container.
name	The name of the container.
size	The numeric total size (in bytes) of the folder, formatted with SAS format SIZEKMG.

Details

Gets the dashboard resource utilization information for child folders with calculated sizes for the current user's workspace.

All system (hidden) folders are aggregated into a single system folder that represents all hidden content in a single folder representation.

15

SignatureService Module

Overview

Macros to retrieve electronic signature information for files that are in the repository.

Macro Summary Table

Category	Language Elements	Description
SignatureService Module	%LSAF_GETSIGNATURES Macro (p. 200)	Gets the metadata for all of the signatures of a file or a specific version of the file. The metadata is stored in a SAS data set.
	%LSAF_GETSIGNINGSTATUS Macro (p. 201)	Gets the status of a signature for a repository file.
	%LSAF_ISVERSIONSIGNED Macro (p. 202)	Indicates whether the specified version of a file in the repository is signed.

Dictionary

%LSAF_GETSIGNATURES Macro

Gets the metadata for all of the signatures of a file or a specific version of the file. The metadata is stored in a SAS data set.

Category: SignatureService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETSIGNATURES(LSAF_PATH=*path*<, LSAF_VERSION=*version*><, SAS_DSNAME=*name*>);

Required Argument

LSAF_PATH=*path*
The path of the file in the repository.

Optional Arguments

LSAF_VERSION=*version*
The specific version number of the file. See “LSAF_VERSION=version” on page 6 for more information about this argument.

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the signatures. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETSIGNATURES.

The data set contains a row for each signature for file and version specified, sorted by descending date of signature, and the following columns:

Column Name	Description
name	The name of the file.
path	The path of the file.
itemType	The type of file (such as data set or program).

Column Name	Description
version	The version that is associated with this specific item.
userId	The unique login identifier for the signer of the file.
dateTimeStamp	The date and time of the signature, represented as a string.
dateSigned	The date and time of the signature, represented as a SAS Datetime format.
reason	The reason specified when the file was signed. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=Signature Reason`.
role	The role specified when the file was signed. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=Signature Role`.
comment	The comment that was entered when the file was signed.

Note: See "Format Date Values" on page 4 for information about date formats.

%LSAF_GETSIGNINGSTATUS Macro

Gets the status of a signature for a repository file.

Category: SignatureService Module

Note: This macro sets the standard macro variables and _lsafSigningStatus_, which is the signature status.

Syntax

%LSAF_GETSIGNINGSTATUS(LSAF_PATH=*path*);

Required Argument

LSAF_PATH=*path*
The path of the repository file.

Details

Sets the value of the macro variable _lsafSigningStatus_ to the signature status. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=SigningStatus`. See "Value Macros" on page 5 for information about value macros.

%LSAF_ISVERSIONSIGNED Macro

Indicates whether the specified version of a file in the repository is signed.

Category: SignatureService Module

Note: This macro sets the standard macro variables and _lsafIsVersionSigned_, which indicates whether the version of the file is signed.

Syntax

%LSAF_ISVERSIONSIGNED(LSAF_PATH=*path*, LSAF_VERSION=*version*);

Required Arguments

LSAF_PATH=*path*
The path of the repository file.

LSAF_VERSION=*version*
The version of the file. See "LSAF_VERSION=version" on page 6 for more information about this argument.

Details

Sets the Boolean macro variable _lsafIsVersionSigned_, which indicates whether the specified version of a file in the repository is signed. See "Boolean Macros" on page 4 for information about return values.

16

WorkspaceService Module

Overview

Macros to manage content in the workspace.

Macro Summary Table

Category	Language Elements	Description
WorkspaceService Module	%LSAF_COPYWORKSPACEITEM Macro (p. 204)	Copies a workspace file or folder to the specified location.

Category	Language Elements	Description
	%LSAF_CREATEWORKSPACEFILE Macro (p. 205)	Uploads a file from the local computer to the workspace.
	%LSAF_CREATEWORKSPACEFOLDER Macro (p. 206)	Creates a folder in the workspace.
	%LSAF_DELETEWORKSPACEITEM Macro (p. 206)	Deletes a workspace item.
	%LSAF_DOWNLOADWORKSPACEFILE Macro (p. 207)	Downloads a workspace file to a local computer.
	%LSAF_EXISTSINWORKSPACE Macro (p. 207)	Indicates whether an item exists in the workspace.
	%LSAF_GETWORKSPACECHECKOUTSTATUS Macro (p. 208)	Gets the checkout status of the workspace file.
	%LSAF_GETWORKSPACECHILDREN Macro (p. 209)	Gets the metadata for items in a folder in the workspace of the current user. The metadata is stored in a SAS data set.
	%LSAF_MARKWORKSPACEITEMFORADD Macro (p. 210)	Marks a workspace file for addition to the repository.
	%LSAF_MOVEWORKSPACEITEM Macro (p. 210)	Moves a workspace file or folder to the specified target location in the workspace.
	%LSAF_RENAMEWORKSPACEITEM Macro (p. 211)	Renames an item in the workspace.
	%LSAF_UPDATEWORKSPACEFILE Macro (p. 212)	Updates the contents of a file in the workspace by uploading a new version of the file from the local computer.

Dictionary

%LSAF_COPYWORKSPACEITEM Macro

Copies a workspace file or folder to the specified location.

Category: WorkspaceService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_COPYWORKSPACEITEM(LSAF_SOURCE=*source-path*, LSAF_TARGET=*target-path*);

Required Arguments

LSAF_SOURCE=*source-path*
The case-sensitive workspace path of the file or folder to copy.

LSAF_TARGET=*target-path*
The full path (which includes the name and, for files, the extension) for the copied item.

Details

Copies a workspace file or folder to the specified location. The workspace item to copy must not exist in the specified location. The children of a copied folder are also copied to the new location and maintain their original directory structure.

If the parent folders needed to create the target path do not exist, they are created.

%LSAF_CREATEWORKSPACEFILE Macro

Uploads a file from the local computer to the workspace.

Category: WorkspaceService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_CREATEWORKSPACEFILE(LOCAL_PATH=*local-path*, LSAF_PATH=*workspace-path*);

Required Arguments

LOCAL_PATH=*local-path*
The absolute path and name of the file on the local computer.

LSAF_PATH=*workspace-path*
The path and name of the file to create in the workspace.

Details

Uploads a file from the local computer to the workspace. All parent folders that do not exist are created.

Note: This macro is supported only from a PC SAS session. Calling this macro within an active SAS Life Science Analytics Framework session results in a failure.

%LSAF_CREATEWORKSPACEFOLDER Macro

Creates a folder in the workspace.

Category: WorkspaceService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_CREATEWORKSPACEFOLDER(LSAF_PATH=*path*);

Required Argument

LSAF_PATH=*path*
The path to the folder.

Details

Creates a folder in the workspace. All parent folders that do not exists are created.

%LSAF_DELETEWORKSPACEITEM Macro

Deletes a workspace item.

Category: WorkspaceService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_DELETEWORKSPACEITEM(LSAF_PATH=*path*);

Required Argument

LSAF_PATH=*path*
The path to the item.

%LSAF_DOWNLOADWORKSPACEFILE Macro

Downloads a workspace file to a local computer.

Category: WorkspaceService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_DOWNLOADWORKSPACEFILE(LSAF_PATH=*lsaf-path*, LOCAL_PATH=*local-path*);

Required Arguments

LSAF_PATH=*lsaf-path*
The path of the workspace to download.

LOCAL_PATH=*local-path*
The full path and name of the file on the local computer.

Details

Downloads a workspace file to a local computer.

If the local directory does not exist, it is created. If the local file exists, it is overwritten.

Note: This macro is supported only from a PC SAS session. Calling this macro from within an active SAS Life Science Analytics Framework session results in a failure.

%LSAF_EXISTSINWORKSPACE Macro

Indicates whether an item exists in the workspace.

Category: WorkspaceService Module

Note: This macro sets the standard macro variables and _lsafExistsInWorkspace_, which indicates whether the item exists.

Syntax

%LSAF_EXISTSINWORKSPACE(LSAF_PATH=*path*);

Required Argument

LSAF_PATH=*path*
The path to the item.

Details

Sets the Boolean macro variable _lsafExistsInWorkspace_, which indicates whether an item exists in the workspace. See “Boolean Macros” on page 4 for information about return values.

%LSAF_GETWORKSPACECHECKOUTSTATUS Macro

Gets the checkout status of the workspace file.

Category: WorkspaceService Module

Note: This macro sets the standard macro variables and _lsafWorkspaceCheckoutStatus_, which is the checkout status of the workspace file.

Syntax

%LSAF_GETWORKSPACECHECKOUTSTATUS(LSAF_PATH=*path*);

Required Argument

LSAF_PATH=*path*
The path of the workspace file.

Details

Gets the checkout status of the workspace file. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=CheckoutStatus`.

%LSAF_GETWORKSPACECHILDREN Macro

Gets the metadata for items in a folder in the workspace of the current user. The metadata is stored in a SAS data set.

Category: WorkspaceService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETWORKSPACECHILDREN(LSAF_PATH=*path*<, LSAF_RECURSIVE=*level*><, SAS_DSNAME=*name*>);

Required Argument

LSAF_PATH=*path*
The path of the folder in the workspace.

Optional Arguments

LSAF_RECURSIVE=*level*
The level of recursion. Valid values:

- 0 (Default): Returns the folder only.
- 1 - 99: (inclusive) Returns the container and traverses the tree with the number of levels indicated.

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the items. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETWORKSPACECHILDREN.

The data set contains a row for each item in the list, sorted by path, and the following columns:

Column Name	Description
name	The name of the item.
path	The path of the item.
itemType	The type of the item.

Column Name	Description
isFolder	Indicates whether the item is a folder.
size	The size (in bytes) of the item. This value is set to missing (.) for containers.
formattedSize	The formatted size (in bytes, KB, MB, or TB).
lastModified	The date on which the item was last modified, represented as a String.
dateLastModified	The date on which the item was last modified, represented as a SAS Datetime format.

Note: See "Format Date Values" on page 4 for information about date formats.

%LSAF_MARKWORKSPACEITEMFORADD Macro

Marks a workspace file for addition to the repository.

Category: WorkspaceService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_MARKWORKSPACEITEMFORADD(LSAF_PATH=*path*);

Required Argument

LSAF_PATH=*path*
The path of the workspace file.

%LSAF_MOVEWORKSPACEITEM Macro

Moves a workspace file or folder to the specified target location in the workspace.

Category: WorkspaceService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_MOVEWORKSPACEITEM(LSAF_SOURCE=*source-path*, LSAF_TARGET=*target-path*);

Required Arguments

LSAF_SOURCE=*source-path*
The case-sensitive workspace path of the file or folder to move.

LSAF_TARGET=*target-path*
The full path (which includes the name, and, for files, the extension) for the moved item. The path cannot end with a slash (/).

Details

Moves a workspace file or folder to the specified target location. The workspace item to move must not exist in the target location. The children of a moved folder are also moved to the target location, keeping their directory structure intact.

Example

```
LSAF_MOVEWORKSPACEITEM(LSAF_SOURCE=%STR(/workspace/srcFolder/
scrFile.txt), LSAF_TARGET=%STR(/workspace/destFolder/scrFile.txt));
```

%LSAF_RENAMEWORKSPACEITEM Macro

Renames an item in the workspace.

Category: WorkspaceService Module

Note: This macro sets the standard macro variables and _lsafRenamedWorkspaceItem_, which is the full path of the renamed item.

Syntax

%LSAF_RENAMEWORKSPACEITEM(LSAF_SOURCE=*source-path*, LSAF_NEWNAME=*name*);

Required Arguments

LSAF_SOURCE=*source-path*
The case-sensitive workspace path of the file or folder to rename.

LSAF_NEWNAME=*name*
The new name of the file or folder. Naming conventions for files and folders are enforced, as is case sensitivity.

Details

Renames an item in the workspace. The new name for a file must include the file extension, otherwise the renaming occurs without the file having an extension.

Sets the value of the macro variable _lsafRenamedWorkspaceItem_ to the full path of the renamed item. See “Value Macros” on page 5 for information about value macros.

%LSAF_UPDATEWORKSPACEFILE Macro

Updates the contents of a file in the workspace by uploading a new version of the file from the local computer.

Category: WorkspaceService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_UPDATEWORKSPACEFILE(LOCAL_PATH=*local-path*, LSAF_PATH=*workspace-path*);

Required Arguments

LOCAL_PATH=*local-path*
The absolute path to the file on the local computer.

LSAF_PATH=*workspace-path*
The path of the file in the workspace to update.

Details

Updates the contents of a file in the workspace by uploading a new version of the file from the local computer.

Note: This macro is supported only from a PC SAS session. Calling this macro within an active SAS Life Science Analytics Framework session results in a failure.

PART 4

Core Macros

17

SystemService Module

Overview

Macros to report the configuration of the system.

Macro Summary Table

Category	Language Elements	Description
SystemService Module	%LSAF_GETAPIVERSIONS Macro (p. 216)	Writes to the SAS log file the current version numbers for the SAS Life Science Analytics Framework API Client and the SAS macros.
	%LSAF_GETSYSTEMCONSTANTS Macro (p. 216)	Gets the list of constants that are used as both macro input parameter values and values in returned data sets. The constants are stored in a SAS data set.

Dictionary

%LSAF_GETAPIVERSIONS Macro

Writes to the SAS log file the current version numbers for the SAS Life Science Analytics Framework API Client and the SAS macros.

Category: SystemService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETAPIVERSIONS;

%LSAF_GETSYSTEMCONSTANTS Macro

Gets the list of constants that are used as both macro input parameter values and values in returned data sets. The constants are stored in a SAS data set.

Category: SystemService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETSYSTEMCONSTANTS(<LSAF_NAME=*name*><, SAS_DSNAME=*name*>);

Optional Arguments

LSAF_NAME=*name*
The name of the constant. If omitted, the full list is returned. Otherwise, just the values for constants with the specified name are returned. If no constants exist with the specified name, an empty data set is returned.

SAS_DSNAME=*name*
The name of the output data set that contains the constants. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETSYSTEMCONSTANTS.

The data set contains a row for each subscription, sorted by name and source, and the following columns:

Column Name	Description
macroApiVersion	The version of the SAS Macro API that generated the list.
source	The source of the constant. This is usually a Java ENUM class package.
name	The name of the constant.
value	A value of the constant.
isDefaultValue	When the constant is used as an optional input parameter value, indicates whether the value for this row is always the default value.
details	Supporting information for the constant. This might be blank for some types.

Details

Gets the list of constants that are used as both macro input parameter values and values in returned data sets. The constants are stored in a SAS data set.

The names of constants can be repeated across sources.

Constant values are case-insensitive when used as input parameters for other macros.

18

TypeService Module

Overview

Macros to describe the item type definitions.

Macro Summary Table

Category	Language Elements	Description
TypeService Module	%LSAF_GETALLTYPES Macro (p. 220)	Gets the metadata for all of the object types. The metadata is stored in a SAS data set.

Category	Language Elements	Description
	%LSAF_GETCONTEXTTYPES Macro (p. 221)	Gets the metadata for all of the context types that are defined in the repository. The metadata is stored in a SAS data set.
	%LSAF_GETSEARCHABLETYPES Macro (p. 222)	Gets the metadata for all of the searchable types that are defined in the repository. The metadata is stored in a SAS data set.
	%LSAF_GETTYPEALLOWABLECHILDREN Macro (p. 223)	Gets the metadata for the item types that can be created as immediate children of the specified context type in the repository. The metadata is stored in a SAS data set.
	%LSAF_GETTYPESBYCAPABILITY Macro (p. 224)	Gets the metadata for the context types based on capability. The metadata is stored in a SAS data set.
	%LSAF_GETTYPEEXTENDEDATTRIBUTES Macro (p. 226)	Gets the metadata for the extended attributes for the specified type. The metadata is stored in a SAS data set.
	%LSAF_ISCONTEXTTYPE Macro (p. 227)	Indicates whether the specified type identifier is a context.
	%LSAF_TYPEEXISTS Macro (p. 227)	Indicates whether a type exists.
	%LSAF_TYPEHASCAPABILITY Macro (p. 228)	Indicates whether the context type has the specified capability in the repository.

Dictionary

%LSAF_GETALLTYPES Macro

Gets the metadata for all of the object types. The metadata is stored in a SAS data set.

Category: TypeService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETALLTYPES(<SAS_DSNAME=*name*>);

Optional Argument

SAS_DSNAME=*name*

The name of the output data set that contains the metadata for the types. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETALLTYPES.

The data set contains a row for each type, sorted by the identifier, and the following columns:

Column Name	Description
listtype	The list type (in this case, All).
id	The unique identifier of the type.
name	The display name of the type.
description	The description of the type.
lsaftype	The object type. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=LsafType`.
allowablechildren	The identifiers of the immediate child types of container types.
capabilities	The functionality that is enabled for the type. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=Capability`.

%LSAF_GETCONTEXTTYPES Macro

Gets the metadata for all of the context types that are defined in the repository. The metadata is stored in a SAS data set.

Category: TypeService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETCONTEXTTYPES(<SAS_DSNAME=*name*>);

Optional Argument

SAS_DSNAME=*name*

The name of the output data set that contains the metadata for the context types. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETCONTEXTTYPES.

The data set contains a row for each context, sorted by the context identifier, and the following columns:

Column Name	Description
listtype	The list type (in this case, Context).
id	The unique identifier for the type.
name	The display name for the type.
description	The description for the type.
lsaftype	The object type (in this case, Context).
allowablechildren	The identifiers for the immediate children of the context.
capabilities	The functionality enabled by the context. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=Capability`.

%LSAF_GETSEARCHABLETYPES Macro

Gets the metadata for all of the searchable types that are defined in the repository. The metadata is stored in a SAS data set.

Category: TypeService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETSEARCHABLETYPES(<SAS_DSNAME=*name*>);

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the searchable types. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETSEARCHABLETYPES.

The data set contains a row for each searchable type, sorted by the identifier, and the following columns:

Column Name	Description
listtype	The list type (in this case, Searchable).
id	The unique identifier for the type.
name	The display name for the type.
description	The description for the type.
lsaftype	The object type. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=LsafType`.

%LSAF_GETTYPEALLOWABLECHILDREN Macro

Gets the metadata for the item types that can be created as immediate children of the specified context type in the repository. The metadata is stored in a SAS data set.

Category: TypeService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETTYPEALLOWABLECHILDREN(LSAF_TYPEID=*type-id*<, SAS_DSNAME=*name*>);

Required Argument

LSAF_TYPEID=*type-id*
The type identifier of the context item in the repository.

Optional Argument

SAS_DSNAME=*name*

The name of the output data set that contains the metadata for the types. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETTYPEALLOWABLECHILDREN.

The data set contains a row for each item in the list, sorted by the identifier, and the following columns:

Column Name	Description
listtype	The list type (in this case Type allowable children).
typeId	The type identifier. To get a list of valid context types, run %LSAF_GETCONTEXTTYPES.
id	The identifier of the child type.
name	The name of the type.
description	The description of the type.
lsaftype	The object type.
allowablechildren	The allowable child types for this type.
capabilities	The capabilities of this child. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=Capability`.

Details

Gets the metadata for the item types that can be created as immediate children of the specified context type in the repository. The metadata is stored in a SAS data set.

%LSAF_GETTYPESBYCAPABILITY Macro

Gets the metadata for the context types based on capability. The metadata is stored in a SAS data set.

Category: TypeService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETTYPESBYCAPABILITY(LSAF_CAPABILITY=*capability*<, SAS_DSNAME=*name*>);

Required Argument

LSAF_CAPABILITY=*capability*
The name of the context capability to retrieve. To get the list of valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=Capability`.

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the context types. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETCONTEXTTYPESBYCAPABILITY.

The data set contains a row for each type, sorted by the identifier, and the following columns:

Column Name	Description
listtype	The type of list (in this case, Types with the specified capability).
id	The unique identifier for the type.
name	The display name for the type.
description	The description for the type.
lsaftype	The object type.
allowablechildren	The identifiers for the immediate child types.
capabilities	The functionality enabled by the type. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=Capability`.

%LSAF_GETTYPEEXTENDEDATTRIBUTES Macro

Gets the metadata for the extended attributes for the specified type. The metadata is stored in a SAS data set.

Category: TypeService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETTYPEEXTENDEDATTRIBUTES(<LSAF_TYPEID=*type-id*><, SAS_DSNAME=*name*>);

Optional Arguments

LSAF_TYPEID=*type-id*
The identifier of the type. If omitted, the metadata for the extended attributes for all types is returned. To get a list of valid type IDs, run %LSAF_GETALLTYPES.

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the extended attributes. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETTYPEEXTENDEDATTRIBUTES.

The data set contains a row for each extended attribute, sorted by the identifier and attribute name, and the following columns:

Column Name	Description
typeId	The unique identifier for the type.
typeName	The display name for the type.
attrId	The unique identifier for the extended attribute.
attrName	The display name for the extended attribute.
attrType	The type of the extended attribute.
attrIsInherited	Indicates whether the extended attribute is inherited from a parent type.

%LSAF_ISCONTEXTTYPE Macro

Indicates whether the specified type identifier is a context.

Category: TypeService Module

Note: This macro sets the standard macro variables and _lsafIsContextType_, which indicates whether the value is a context type.

Syntax

%LSAF_ISCONTEXTTYPE(LSAF_TYPEID=*type-id*);

Required Argument

LSAF_TYPEID=*type-id*
The identifier of the type. To get a list of valid context types, run %LSAF_GETCONTEXTTYPES.

Details

Sets the Boolean macro variable _lsafIsContextType_, which indicates whether the specified type identifier is a context. See “Boolean Macros” on page 4 for information about return values.

%LSAF_TYPEEXISTS Macro

Indicates whether a type exists.

Category: TypeService Module

Note: This macro sets the standard macro variables and _lsafTypeExists, which indicates whether the type exists.

Syntax

%LSAF_TYPEEXISTS(LSAF_TYPEID=*type-id*);

Required Argument

LSAF_TYPEID=*type-id*
The identifier of the type.

Details

Sets the Boolean macro variable _lsafTypeExists_, which indicates whether a type exists. See “Boolean Macros” on page 4 for information about return values.

%LSAF_TYPEHASCAPABILITY Macro

Indicates whether the context type has the specified capability in the repository.

Category: TypeService Module

Note: This macro sets the standard macro variables and _lsafTypeHasCapability_, which indicates whether the type has the capability.

Syntax

%LSAF_TYPEHASCAPABILITY(LSAF_TYPEID=*type-id*, LSAF_CAPABILITY=*capability*);

Required Arguments

LSAF_TYPEID=*type-id*
The identifier of the type. To get a list of valid context types, run %LSAF_GETCONTEXTTYPES.

LSAF_CAPABILITY=*capability*
The capability of the type. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=Capability`.

Details

Sets the Boolean macro variable _lsafTypeHasCapability_, which indicates whether the type has the specified capability in the repository. See “Boolean Macros” on page 4 for information about return values.

19

UtilityService Module

Overview

Macros to derive relative and absolute paths.

Macro Summary Table

Category	Language Elements	Description
UtilityService Module	%LSAF_GETABSOLUTEPATH Macro (p. 230)	Gets an absolute path for an item relative to the specified base path.
	%LSAF_GETRELATIVEPATH Macro (p. 231)	Gets a relative path for an item relative to the specified base path.

Dictionary

%LSAF_GETABSOLUTEPATH Macro

Gets an absolute path for an item relative to the specified base path.

Category: UtilityService Module

Note: This macro sets the standard macro variables and _lsafAbsolutePath_, which is the absolute path.

Syntax

%LSAF_GETABSOLUTEPATH(LSAF_BASE_PATH=*path*, LSAF_ABSOLUTE_PATH=*path*);

Required Arguments

LSAF_BASE_PATH=*path*
The absolute path to the base container location.

LSAF_ABSOLUTE_PATH=*path*
The path of the file relative to the base path. A value of . returns the full base path. A value of .. returns the parent of the base path. For example, if the base path is /SAS/Folder1/Folder2/Folder3:

- A value of ./File4 returns an absolute path value of /SAS/Folder1/Folder2/Folder3/Folder4.
- A value of ../Folder4 returns an absolute path value of /SAS/Folder1/Folder2/Folder4.
- A value of ../../Folder4 returns an absolute path value of /SAS/Folder1/Folder4.

Details

Gets an absolute path for an item relative to the specified base path. Neither path is validated for existence or for valid characters. Logging in to SAS Life Science Analytics Framework is not required to perform this operation.

Sets the value of the macro variable _lsafAbsolutePath_ to the absolute path. See “Value Macros” on page 5 for information about value macros.

%LSAF_GETRELATIVEPATH Macro

Gets a relative path for an item relative to the specified base path.

Category: UtilityService Module

Note: This macro sets the standard macro variables and _lsafRelativePath_, which is the relative path.

Syntax

%LSAF_GETRELATIVEPATH(LSAF_BASE_PATH=*path*, LSAF_ABSOLUTE_PATH=*path*);

Required Arguments

LSAF_BASE_PATH=*path*
The absolute path to the base container location.

LSAF_ABSOLUTE_PATH=*path*
The absolute path of the file relative to the base path.

Details

Gets a relative path for an item relative to the specified base path. Neither path is validated for existence or for valid characters. Logging in to SAS Life Science Analytics Framework is not required to perform this operation.

Sets the value of the macro variable _lsafRelativePath_ to the relative path. See “Value Macros” on page 5 for information about value macros.

PART 5

Execution Macros

20

JobManifestService Module

Overview

Macros to retrieve the information from a job manifest file.

Macro Summary Table

Category	Language Elements	Description
JobManifestService Module	%LSAF_GETJOBMANIFESTEXTERNALREFS Macro (p. 236)	Gets the metadata for the external references for the specified job manifest file. The metadata is stored in a SAS data set.

Category	Language Elements	Description
	%LSAF_GETJOBMANIFESTINFO Macro (p. 238)	Gets the information for a job manifest file.
	%LSAF_GETJOBMANIFESTINPUTS Macro (p. 239)	Gets the metadata for the input files for the specified job manifest file. The metadata is stored in a SAS data set.
	%LSAF_GETJOBMANIFESTLOGLOCS Macro (p. 240)	Gets the log location information for a job manifest file. The information is stored in a SAS data set.
	%LSAF_GETJOBMANIFESTMETRICS Macro (p. 242)	Gets the metrics for the specified job manifest file. The data is stored in a SAS data set.
	%LSAF_GETJOBMANIFESTOUTPUTS Macro (p. 243)	Gets the metadata for the output files for the specified job manifest file. The metadata is stored in a SAS data set.
	%LSAF_GETJOBMANIFESTPARAMETERS Macro (p. 244)	Gets the metadata for the parameters for the specified job manifest file. The metadata is stored in a SAS data set.
	%LSAF_GETJOBMANIFESTPROGRAMS Macro (p. 245)	Gets the metadata for the programs for the specified job manifest file. The metadata is stored in a SAS data set.
	%LSAF_GETJOBMANIFESTRESULTLOCS Macro (p. 247)	Gets the result location information for a job manifest file that is. The information is stored in a SAS data set.

Dictionary

%LSAF_GETJOBMANIFESTEXTERNALREFS Macro

Gets the metadata for the external references for the specified job manifest file. The metadata is stored in a SAS data set.

Category: JobManifestService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETJOBMANIFESTEXTERNALREFS(LSAF_LOCATION=REPOSITORY | WORKSPACE, LSAF_PATH=*path*<, LSAF_VERSION=*version*><, LSAF_TYPE=*reference-type*><, SAS_DSNAME=*name*>);

Required Arguments

LSAF_LOCATION=REPOSITORY | WORKSPACE
The case-insensitive location of the manifest file.

LSAF_PATH=*path*
The case-sensitive path of the job manifest file.

Optional Arguments

LSAF_VERSION=*version*
The specific version number of the job manifest file. See “LSAF_VERSION=version” on page 6 for more information about this argument.

LSAF_TYPE=*reference-type*
The type of the external references to retrieve. A blank or missing argument value retrieves both types.

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the external references. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETJOBMANIFESTEXTERNALREFS.

The data set contains a row for each job manifest file external reference, and columns with the following names. The data set retains the order of the external references.

Column Name	Description
jobManifestLocation	The job manifest source location: REPOSITORY or WORKSPACE.
jobManifestPath	The path of the job manifest file.
jobManifestVersion	The job manifest file version.
path	The path of the external reference.
type	The type of the external reference. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=ManifestExtRefType`.
referenceProgram	The referenced SAS program file.
referenceType	The type of the external reference.

%LSAF_GETJOBMANIFESTINFO Macro

Gets the information for a job manifest file.

Category: JobManifestService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETJOBMANIFESTINFO(LSAF_LOCATION=REPOSITORY | WORKSPACE, LSAF_PATH=*path*<, LSAF_VERSION=*version*><, SAS_DSNAME=*name*>);

Required Arguments

LSAF_LOCATION=REPOSITORY | WORKSPACE
The case-insensitive location of the manifest file.

LSAF_PATH=*path*
The case-sensitive path of the job manifest file.

Optional Arguments

LSAF_VERSION=*version*
The specific version number of the job manifest file. See “LSAF_VERSION=version” on page 6 for more information about this argument.

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the manifest file. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETJOBMANIFESTINFO.

The data set contains the following variables that represent each element of the metadata for the job manifest file, sorted by the NAME variable:

Column Name	Description
jobManifestLocation	The job manifest source location: REPOSITORY or WORKSPACE.
jobManifestPath	The path of the job manifest file.
jobManifestVersion	The version number that is associated with the job manifest file.
name	The name of the property or attribute.

Column Name	Description
value	The value of the property or attribute.

Details

Gets the information for a job manifest file. The information is stored in a SAS data set. The output data set can be used as an input data set for %LSAF_CREATEWORKSPACEJOB.

This is the information that is displayed on the **Details** tab of the job manifest file and the information that concerns the versioning of new output files.

%LSAF_GETJOBMANIFESTINPUTS Macro

Gets the metadata for the input files for the specified job manifest file. The metadata is stored in a SAS data set.

Category: JobManifestService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETJOBMANIFESTINPUTS(LSAF_LOCATION=REPOSITORY | WORKSPACE, LSAF_PATH=*path*<, LSAF_VERSION=*version*><, SAS_DSNAME=*name*>);

Required Arguments

LSAF_LOCATION=REPOSITORY | WORKSPACE
The case-insensitive location of the manifest file.

LSAF_PATH=*path*
The case-sensitive path of the job manifest file.

Optional Arguments

LSAF_VERSION=*version*
The specific version number of the job manifest file. See “LSAF_VERSION=version” on page 6 for more information about this argument.

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the inputs. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETJOBMANIFESTINPUTS.

The data set contains a row for each job manifest file input and columns with the following names. The data set retains the order of the inputs.

Column Name	Description
jobManifestLocation	The job manifest source location: REPOSITORY or WORKSPACE.
jobManifestPath	The path of the job manifest file.
jobManifestVersion	The job manifest file version.
jobLocation	The job source location.
path	The path of the input file.
version	The version of the input file that is used in the job submission, or if the file is not versioned, blank .
size	The size (in bytes) of the input file.
sizeFormatted	The formatted size (in bytes, KB, MB, or TB) of the input file.
referenceProgram	The referenced SAS program file.

%LSAF_GETJOBMANIFESTLOGLOCS Macro

Gets the log location information for a job manifest file. The information is stored in a SAS data set.

Category: JobManifestService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETJOBMANIFESTLOGLOCS(LSAF_LOCATION=REPOSITORY | WORKSPACE, LSAF_PATH=*path*<, LSAF_VERSION=*version*><, SAS_DSNAME=*name*>);

Required Arguments

LSAF_LOCATION=REPOSITORY | WORKSPACE
The case-insensitive location of the manifest file.

LSAF_PATH=*path*
The case-sensitive path of the job manifest file.

Optional Arguments

LSAF_VERSION=*version*
The specific version number of the job manifest file. See "LSAF_VERSION=version" on page 6 for more information about this argument.

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the manifest log locations. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETJOBMANIFESTLOGLOCS.

The data set contains a row for each log location, sorted by the log name, and the following columns:

Column Name	Description
jobManifestLocation	The job manifest source location: REPOSITORY or WORKSPACE.
jobManifestPath	The path of the job manifest file.
listType	The type of the property or attribute (LOGS).
jobManifestVersion	The version number that is associated with the job manifest file.
logPath	The path of the log file.
logName	The name of the log file.
logVersion	The version of the log file.
engineType	The type of system that generated the file.
dateModified	The date on which the log file was last modified. See "Format Date Values" on page 4 for information about date formats.
srcLocation	The source location (REPOSITORY or WORKSPACE).
fileSize	The size (in bytes) of the file.
fileSizeFmt	The formatted size of the file (for example, 32 KB).

%LSAF_GETJOBMANIFESTMETRICS Macro

Gets the metrics for the specified job manifest file. The data is stored in a SAS data set.

Category: JobManifestService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETJOBMANIFESTMETRICS(LSAF_LOCATION=REPOSITORY | WORKSPACE, LSAF_PATH=*path*<, LSAF_VERSION=*version*><, SAS_DSNAME=*name*>);

Required Arguments

LSAF_LOCATION=REPOSITORY | WORKSPACE
The case-insensitive location of the manifest file.

LSAF_PATH=*path*
The path of the job manifest file.

Optional Arguments

LSAF_VERSION=*version*
The specific version number of the job manifest file. See “LSAF_VERSION=version” on page 6 for more information about this argument.

SAS_DSNAME=*name*
The name of the output data set that contains the manifest file metrics. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETJOBMANIFESTMETRICS.

The data set contains a row for each job manifest metric and columns with the following names:

Column Name	Description
jobManifestLocation	The job manifest source location: REPOSITORY or WORKSPACE.
jobManifestPath	The path of the job manifest file.
jobManifestVersion	The job manifest file version.
type	The type of metric.
total	The total for the metric.

Column Name	Description
size	The size of the metric.
formattedsize	The formatted size (in bytes, KB, MB, or TB).
time	The time or interval of the metric.
formattedtime	The formatted time or interval of the metric, which is displayed as h:m:s:ms.

%LSAF_GETJOBMANIFESTOUTPUTS Macro

Gets the metadata for the output files for the specified job manifest file. The metadata is stored in a SAS data set.

Category: JobManifestService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETJOBMANIFESTOUTPUTS(LSAF_LOCATION=REPOSITORY | WORKSPACE, LSAF_PATH=*path*<, LSAF_VERSION=*version*><, SAS_DSNAME=*name*>);

Required Arguments

LSAF_LOCATION=REPOSITORY | WORKSPACE
The case-insensitive location of the manifest file.

LSAF_PATH=*path*
The case-sensitive path of the job manifest file.

Optional Arguments

LSAF_VERSION=*version*
The specific version number of the job manifest file. See “LSAF_VERSION=version” on page 6 for more information about this argument.

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the outputs. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETJOBMANIFESTOUTPUTS.

The data set contains a row for each job manifest file output and columns with the following names. The data set retains the order of the inputs.

Column Name	Description
jobManifestLocation	The job manifest source location: REPOSITORY or WORKSPACE.
jobManifestPath	The path of the job manifest file.
jobManifestVersion	The job manifest file version.
jobLocation	The job source location.
path	The path of the output file.
version	The version of the output file that is used in the job submission, or if the output file is not versioned, blank.
size	The size (in bytes) of the output file.
sizeFormatted	The formatted size (in bytes, KB, MB, or TB) of the output file.
referenceProgram	The referenced SAS program file.

%LSAF_GETJOBMANIFESTPARAMETERS Macro

Gets the metadata for the parameters for the specified job manifest file. The metadata is stored in a SAS data set.

Category: JobManifestService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETJOBMANIFESTPARAMETERS(LSAF_LOCATION=REPOSITORY | WORKSPACE, LSAF_PATH=*path*<, LSAF_VERSION=*version*><, SAS_DSNAME=*name*>);

Required Arguments

LSAF_LOCATION=REPOSITORY | WORKSPACE
The case-insensitive location of the manifest file.

LSAF_PATH=*path*
The case-sensitive path of the job manifest file.

Optional Arguments

LSAF_VERSION=*version*
The specific version number of the job manifest file. See “LSAF_VERSION=version” on page 6 for more information about this argument.

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the parameters. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETJOBMANIFESTPARAMETERS.

The data set contains a row for each job manifest file parameter and columns with the following names:

Column Name	Description
jobManifestLocation	The job manifest source location: REPOSITORY or WORKSPACE.
jobManifestPath	The path of the job manifest file.
jobManifestVersion	The job manifest file version.
jobLocation	The job source location.
name	The name of the parameter.
type	The type of the parameter.
label	The label of the parameter.
value	The value of the parameter.
version	The version of the file that is used in the job submission, or if file is not versioned, blank.
includeSubfolders	Indicates whether to include subfolders.

%LSAF_GETJOBMANIFESTPROGRAMS Macro

Gets the metadata for the programs for the specified job manifest file. The metadata is stored in a SAS data set.

Category: JobManifestService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETJOBMANIFESTPROGRAMS(LSAF_LOCATION=REPOSITORY | WORKSPACE, LSAF_PATH=*path*<, LSAF_VERSION=*version*><, SAS_DSNAME=*name*>);

Required Arguments

LSAF_LOCATION=REPOSITORY | WORKSPACE
The case-insensitive location of the manifest file.

LSAF_PATH=*path*
The case-sensitive path of the job manifest file.

Optional Arguments

LSAF_VERSION=*version*
The specific version number of the job manifest file. See "LSAF_VERSION=version" on page 6 for more information about this argument.

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the programs. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETJOBMANIFESTPROGRAMS.

The data set contains a row for each job manifest file program and columns with the following names. The data set retains the order of the programs.

Column Name	Description
jobManifestLocation	The job manifest source location: REPOSITORY or WORKSPACE.
jobManifestPath	The path of the job manifest file.
jobManifestVersion	The job manifest file version.
jobLocation	The job source location.
path	The path of the program file.
version	The version of the program file that is used in the job submission, or if the program file is not versioned, blank.
size	The size (in bytes) of the program.
sizeFormatted	The formatted size (in bytes, KB, MB, or TB) of the input file.

%LSAF_GETJOBMANIFESTRESULTLOCS Macro

Gets the result location information for a job manifest file that is. The information is stored in a SAS data set.

Category: JobManifestService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETJOBMANIFESTRESULTLOCS(LSAF_LOCATION=REPOSITORY | WORKSPACE, LSAF_PATH=*path*<, LSAF_VERSION=*version*><, SAS_DSNAME=*name*>);

Required Arguments

LSAF_LOCATION=REPOSITORY | WORKSPACE
The case-insensitive location of the manifest file.

LSAF_PATH=*path*
The case-sensitive path of the job manifest file.

Optional Arguments

LSAF_VERSION=*version*
The specific version number of the job manifest file. See “LSAF_VERSION=version” on page 6 for more information about this argument.

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the manifest file result locations. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETJOBMANIFESTRESULTLOCS.

The data set contains the following variables that represent each element of the metadata for the result locations, sorted by the RESULTNAME variable:

Column Name	Description
jobManifestLocation	The job manifest source location: REPOSITORY or WORKSPACE.
jobManifestPath	The path of the job manifest file.
listType	The type of the property or attribute (RESULTS).

Column Name	Description
jobManifestVersion	The version number that is associated with the job manifest file.
resultPath	The path of the result file.
resultName	The name of the result file.
resultVersion	The version of the result file.
engineType	The type of system that generated the file.
dateModified	The date on which the result file was last modified. See “Format Date Values” on page 4 for information about date formats.
srcLocation	The source location (REPOSITORY or WORKSPACE).
fileSize	The size (in bytes) of the file.
fileSizeFmt	The formatted size of the file (such as 32 KB).

21

JobService Module

Overview

Macros to create, update, and retrieve information for jobs.

Macro Summary Table

Category	Language Elements	Description
JobService Module	%LSAF_CREATEWORKSPACEJOB Macro (p. 251)	Creates a job in the workspace of the current user.
	%LSAF_GETJOBCHECKINLOCATIONS Macro (p. 255)	Gets the metadata for the checkin locations for the specified job in the repository. The metadata is stored in a SAS data set.
	%LSAF_GETJOBINFO Macro (p. 256)	Gets the information for a job in the repository. The information is stored in a SAS data set.
	%LSAF_GETJOBPARAMETERS Macro (p. 257)	Gets the job parameter metadata for the specified job. The metadata is stored in a SAS data set.
	%LSAF_GETJOBPROGRAMS Macro (p. 259)	Gets the metadata for the programs for the specified job in the repository. The metadata is stored in a SAS data set.
	%LSAF_GETJOBRUNASOWNER Macro (p. 260)	Gets the value of the property Run as owner for a repository job.
	%LSAF_GETJOBSETUP Macro (p. 260)	Gets the metadata for the setup for the specified job in the repository. The metadata is stored in a SAS data set. The list does not include the setup programs.
	%LSAF_GETWORKSPACEJOBCHECKINLOCS Macro (p. 261)	Gets the metadata for the checkin locations for the specified job in the workspace of the current user. The metadata is stored in a SAS data set.
	%LSAF_GETWORKSPACEJOBINFO Macro (p. 262)	Gets the information for a job in the workspace of the current user. The information is stored in a SAS data set.
	%LSAF_GETWORKSPACEJOBPARAMETERS Macro (p. 264)	Gets the job parameter metadata for the specified job in the workspace of the current user. The metadata is stored in a SAS data set.
	%LSAF_GETWORKSPACEJOBPROGRAMS Macro (p. 265)	Gets the metadata for the programs for the specified job in the workspace of the current user. The metadata is stored in a SAS data set.
	%LSAF_GETWORKSPACEJOBSETUP Macro (p. 266)	Gets the metadata for the setup items for the specified job in the workspace of the current user. The metadata is stored in a SAS data set. The list does not include the setup programs.
	%LSAF_SETJOBRUNASOWNER Macro (p. 267)	Sets the job property Run as owner. Only the owner can change the property.

Category	Language Elements	Description
	%LSAF_UPDATEWORKSPACEJOB Macro (p. 268)	Updates a job in workspace of the current user.

Dictionary

%LSAF_CREATEWORKSPACEJOB Macro

Creates a job in the workspace of the current user.

Category: JobService Module

Note: This macro sets the standard macro variables and _lsafWorkspaceJobPath_, which is the full path of the new job.

Syntax

%LSAF_CREATEWORKSPACEJOB(LSAF_PATH=*path*<, LSAF_USERELATIVEPATHS=0 | 1><, SAS_DSNAME_INFO=*name*><, SAS_DSNAME_PROGRAMS=*name*><, SAS_DSNAME_PARAMETERS=*name*><, SAS_DSNAME_SETUP=*name*><, SAS_DSNAME_CHECKIN=*name*>);

Required Argument

LSAF_PATH=*path*
The path for the job to create in the workspace of the current user.

Optional Arguments

LSAF_USERELATIVEPATHS=0 | 1 (Default)
Indicates whether to generate setup and checkin locations as absolute or relative paths.

SAS_DSNAME_INFO=*name*
The name of the output data set that contains the job information. See “Data Set Macros” on page 5 for information about specifying the name of the data set. If this argument is not included in the call, the default job information is set.

The data set must contain at least the following columns. Additional variables are ignored.

Column Name	Description
name	The name of the job property. The information that can be specified is DOCUMENTATION, LOGSLOCATION, RESULTSLOCATION, MANIFESTLOCATION, NEWOUTPUTS_CUSTOMVERSION, NEWOUTPUTS_ENABLEVERSION, and NEWOUTPUTS_VERSIONTYPE. All other values listed in the data set are ignored. All data are case-insensitive. If a name is listed more than once, the last value processed is used. If a name is omitted, the default value is used.
value	The value of the property.

SAS_DSNAME_PROGRAMS=*name*

The name of the output data set that contains the list of paths to the SAS programs to run. See "Data Set Macros" on page 5 for information about specifying the name of the data set. If this argument is not included in the call, no programs are set.

The data set must contain at least the following columns. Additional variables are ignored.

Column Name	Description
programPath	The path of the SAS program file.
programVersion	The version of the SAS program file. Blank values specify the latest file version.

SAS_DSNAME_PARAMETERS=*name*

The name of the output data set that contains the job parameter information. See "Data Set Macros" on page 5 for information about specifying the name of the data set. If this argument is not included in the call, no parameters are set.

The data set must contain at least the following columns. Additional variables are ignored.

Column Name	Description
name	The name of the parameter.
label	The label for the parameter.
type	The parameter type. Valid values: ■ CHARACTER

Column Name	Description
	■ CHARACTER_MASKED ■ NUMERIC ■ DATE ■ FILE ■ FOLDER
defaultValue	The default value for the parameter. See “Format Date Values” on page 4 for information about date formats.
fileVersion	If the parameter is type FILE, the file version. Blank values specify the latest file version.
includeSubFolders (numeric)	Indicates whether the folder includes subfolders. Valid values: ■ 0: No. ■ 1: Yes.

Data set rows with duplicate values for parameter name result in a data set validation error.

SAS_DSNAME_SETUP=*name*

The name of the output data set that contains the list of the job setup items. See “Data Set Macros” on page 5 for information about specifying the name of the data set. If this argument is not included in the call, no setup items are set.

The data set must contain at least the following columns. Additional variables are ignored.

Column Name	Description
setupPath (character)	The path of the setup file or container.
setupType (character)	The type of setup item. Valid values: ■ FILE ■ CONTAINER
setupVersion (character)	The version of the FILE item. Blank values specify the latest file version. The value is ignored for setup items of type CONTAINER.
setupIncludeSubFolders (numeric)	Indicates whether the setup folder includes subfolders. Valid values:

Column Name	Description
	■ 0: No. ■ 1: Yes. The value is ignored for setup items of type FILE.
setupIncludeFiles (numeric)	Indicates whether the setup includes files of type CONTAINER. Valid values: ■ 0: No. ■ 1: Yes.

Data set rows with duplicate values for setupPath result in a data set validation error.

SAS_DSNAME_CHECKIN=*name*

The name of the output data set that contains the list of the job checkin locations. See “Data Set Macros” on page 5 for information about specifying the name of the data set. If this argument is not included in the call, no checkin locations are set.

The data set must contain at least the following columns. Additional variables are ignored.

Column Name	Description
checkinPath (character)	The checkin location path of the container.
checkinIncludeSubFolders (numeric)	Indicates whether the checkin location includes subfolders. Valid values: ■ 0: No. ■ 1: Yes.

Data set rows with duplicate values for checkinPath result in a data set validation error.

Details

Creates a job in the workspace of the current user. Sets the value of the macro variable _lsafWorkspaceJobPath_ to the full path of the created job, which includes the file extension. See “Value Macros” on page 5 for information about value macros.

Data sets that represent the job data to set at creation time can be specified in the call to %LSAF_CREATEWORKSPACEJOB. If data sets are not specified, the default values for all job data are used.

The output data sets from the following macros contain all of the data that is relevant to the job properties that can be specified when the job is created:

- "%LSAF_GETJOBINFO Macro"
- "%LSAF_GETWORKSPACEJOBINFO Macro"
- "%LSAF_GETJOBPROGRAMS Macro"
- "%LSAF_GETWORKSPACEJOBPROGRAMS Macro"
- "%LSAF_GETJOBPARAMETERS Macro"
- "%LSAF_GETWORKSPACEJOBPARAMETERS Macro"
- "%LSAF_GETJOBSETUP Macro"
- "%LSAF_GETWORKSPACEJOBSETUP Macro"
- "%LSAF_GETJOBCHECKINLOCATIONS Macro"
- "%LSAF_GETWORKSPACEJOBCHECKINLOCS Macro"

The output data sets from the following job manifest file macros contain all of the data that is relevant to the job properties that can be specified when the job is created:

- "%LSAF_GETJOBMANIFESTINFO Macro"
- "%LSAF_GETJOBMANIFESTPROGRAMS Macro"
- "%LSAF_GETJOBMANIFESTPARAMETERS Macro"
- "%LSAF_GETJOBMANIFESTOUTPUTS Macro"

%LSAF_GETJOBCHECKINLOCATIONS Macro

Gets the metadata for the checkin locations for the specified job in the repository. The metadata is stored in a SAS data set.

Category: JobService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETJOBCHECKINLOCATIONS(LSAF_PATH=*path*<, LSAF_VERSION=*version*><, SAS_DSNAME=*name*>);

Required Argument

LSAF_PATH=*path*
The path of the job.

Optional Arguments

LSAF_VERSION=*version*
The version of the job. See “LSAF_VERSION=version” on page 6 for more information about this argument.

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the checkin locations. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETJOBCHECKINLOCATIONS.

The data set contains a row for each job checkin location and columns with the following names. The data set retains the order of the checkin locations.

Column Name	Description
jobPath	The path of the job.
jobVersion	The job version.
checkinPath	The path of the checkin location.
checkinIncludeSubFolders	Indicates whether the subfolders of the checkin location container are included during job execution.

%LSAF_GETJOBINFO Macro

Gets the information for a job in the repository. The information is stored in a SAS data set.

Category: JobService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETJOBINFO(LSAF_PATH=*path*<, LSAF_VERSION=*version*><, SAS_DSNAME=*name*>);

Required Argument

LSAF_PATH=*path*
The path of the job.

Optional Arguments

LSAF_VERSION=*version*
The specific version number of the job. See “LSAF_VERSION=version” on page 6 for more information about this argument.

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the job. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETJOBINFO.

The data set contains the following variables that represent each element of the metadata for the job, sorted by the NAME variable:

Column Name	Description
path	The path of the job.
version	The version number that is associated with the job.
name	The name of the property or attribute.
value	The value of the property or attribute.

Details

Gets the information for a job in the repository. The information is stored in a SAS data set.

This is the information that is displayed in the **Details** area and the **Check-in** area of a job in SAS Life Science Analytics Framework. For more information, see “Edit a Job” in *SAS Life Science Analytics Framework: User’s Guide*.

%LSAF_GETJOBPARAMETERS Macro

Gets the job parameter metadata for the specified job. The metadata is stored in a SAS data set.

Category: JobService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETJOBPARAMETERS(LSAF_PATH=*path*<, LSAF_VERSION=*version*><, SAS_DSNAME=*name*>);

Required Argument

LSAF_PATH=*path*
The path of the job.

Optional Arguments

LSAF_VERSION=*version*
The version of the job. See “LSAF_VERSION=version” on page 6 for more information about this argument.

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the programs. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETJOBPROGRAMS.

The data set contains a row for each job program and columns with the following names. The data set retains the order of the programs.

Column Name	Description
jobPath	The path of the job.
jobVersion	The job version.
name	The name of the parameter.
label	The parameter label.
type	The parameter type. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=ParameterType`.
defaultValue	The default value for the parameter. Dates are returned in SAS datetime format and are in GMT format. See “Format Date Values” on page 4 for information about date formats.
fileVersion	If the parameter is type FILE, the file version to use. A blank value specifies that either the file is not versioned or the file is versioned but no version was specified.
includeSubFolders	Indicates whether the subfolders of a container are included during job execution.

%LSAF_GETJOBPROGRAMS Macro

Gets the metadata for the programs for the specified job in the repository. The metadata is stored in a SAS data set.

Category: JobService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETJOBPROGRAMS(LSAF_PATH=*path*<, LSAF_VERSION=*version*><, SAS_DSNAME=*name*>);

Required Argument

LSAF_PATH=*path*
The path of the job.

Optional Arguments

LSAF_VERSION=*version*
The version of the job. See “LSAF_VERSION=version” on page 6 for more information about this argument.

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the parameters. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETJOBPARAMETERS.

The data set contains a row for each job program and columns with the following names. The data set retains the job program order.

Column Name	Description
jobPath	The path of the job.
jobVersion	The job version.
programPath	The path of the program file.
programVersion	The version of the program file. A blank value specifies the latest version of the program file is used by the job.

%LSAF_GETJOBRUNASOWNER Macro

Gets the value of the property Run as owner for a repository job.

Category: JobService Module

Note: This macro sets the standard macro variables and _lsafJobRunAsOwner_, which is the value of the property.

Syntax

%LSAF_GETJOBRUNASOWNER(LSAF_PATH=*path*);

Required Argument

LSAF_PATH=*path*
The case-sensitive path of the job.

Details

Sets the value of the macro variable _lsafJobRunAsOwner_ to the value of the property. See “Value Macros” on page 5 for information about value macros.

%LSAF_GETJOBSETUP Macro

Gets the metadata for the setup for the specified job in the repository. The metadata is stored in a SAS data set. The list does not include the setup programs.

Category: JobService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETJOBSETUP(LSAF_PATH=*path*<, LSAF_VERSION=*version*><, SAS_DSNAME=*name*>);

Required Argument

LSAF_PATH=*path*
The path of the job.

Optional Arguments

LSAF_VERSION=*version*
The version of the job. See “LSAF_VERSION=version” on page 6 for more information about this argument.

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the job setup. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETJOBETUP.

The data set contains a row for each job setup and columns with the following names. The data set retains the order of the inputs.

Column Name	Description
jobPath	The path of the job.
jobVersion	The job version.
setupPath	The path of the setup file.
setupType	The item type for the setup. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=ItemType`.
setupVersion	The version of the setup file. A blank value specifies the latest version of the setup file is used by the job.
setupIncludeSubFolders	Indicates whether the subfolders of a container setup are included during job execution.
setupIncludeFiles	Indicates whether the files of a container setup are included during job execution.

%LSAF_GETWORKSPACEJOBCHECKINLOCS Macro

Gets the metadata for the checkin locations for the specified job in the workspace of the current user. The metadata is stored in a SAS data set.

Category: JobService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETWORKSPACEJOBCHECKINLOCS(LSAF_PATH=*path*<, SAS_DSNAME=*name*>);

Required Argument

LSAF_PATH=*path*
The path of the job.

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the checkin locations. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETWORKSPACEJOBCHECKINLOCS.

The data set contains a row for each job checkin location and columns with the following names. The data set retains the order of the checkin locations.

Column Name	Description
jobPath	The path of the job.
jobVersion	Versions are not relevant in the workspace. This variable is included for consistency with the data set that is returned by the repository macro %LSAF_GETJOBCHECKINLOCATIONS. The value from this macro call is WORKSPACE.
checkinPath	The path of the checkin location.
checkinIncludeSubFolders	Indicates whether the subfolders of a checkin location container are included during job execution.

%LSAF_GETWORKSPACEJOBINFO Macro

Gets the information for a job in the workspace of the current user. The information is stored in a SAS data set.

Category: JobService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETWORKSPACEJOBINFO(LSAF_PATH=*path*<, SAS_DSNAME=*name*>);

Required Argument

LSAF_PATH=*path*
The path of the job in the workspace of the current user.

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the job in the workspace. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETWORKSPACEJOBINFO.

The data set contains the following columns that represent each element of the metadata for the job, sorted by the NAME variable:

Column Name	Description
path	The path of the job.
version	Versions are not relevant in the workspace. This variable is included for consistency with the data set that is returned by %LSAF_GETJOBINFO. The value from this macro call is WORKSPACE.
name	The name of the property or attribute.
value	The value of the property or attribute.

Details

Gets the information for a job in the workspace of the current user. The information is stored in a SAS data set.

This is the information that is displayed in the **Details** area and the **Check-in** area of a job in SAS Life Science Analytics Framework. For more information, see "Edit a Job" in *SAS Life Science Analytics Framework: User's Guide*.

%LSAF_GETWORKSPACEJOBPARAMETERS Macro

Gets the job parameter metadata for the specified job in the workspace of the current user. The metadata is stored in a SAS data set.

Category: JobService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETWORKSPACEJOBPARAMETERS(LSAF_PATH=*path*<, SAS_DSNAME=*name*>);

Required Argument

LSAF_PATH=*path*
The path of the job.

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the parameters. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETWORKSPACEJOBPARAMETERS.

The data set contains a row for each job parameter and columns with the following names. The data set retains the job parameter order.

Column Name	Description
jobPath	The path of the job.
jobVersion	Versions are not relevant in the workspace. This variable is included for consistency with the data set returned by the %LSAF_GETJOBPARAMETERS macro. The value from this macro call is WORKSPACE.
name	The name of the parameter.
label	The parameter label.

Column Name	Description
type	The parameter type. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=ParameterType`.
defaultValue	The default value for the parameter. Dates are returned in SAS datetime format and are in GMT format. See "Format Date Values" on page 4 for information about date formats.
fileVersion	If the parameter is type FILE, the file version to use. A blank value specifies that either the file is not versioned or the file is versioned but no version was specified.
includeSubFolders	Indicates whether the subfolders of a container are included during job execution.

%LSAF_GETWORKSPACEJOBPROGRAMS Macro

Gets the metadata for the programs for the specified job in the workspace of the current user. The metadata is stored in a SAS data set.

Category: JobService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETWORKSPACEJOBPROGRAMS(LSAF_PATH=*path*<, SAS_DSNAME=*name*>);

Required Argument

LSAF_PATH=*path*
The path of the job.

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the programs. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETWORKSPACEJOBPROGRAMS.

The data set contains a row for each job program and columns with the following names. The data set retains the order of the programs.

Column Name	Description
jobPath	The path of the job.
jobVersion	Versions are not relevant in the workspace. This variable is included for consistency with the data set that is returned by the %LSAF_GETJOBPROGRAMS macro. The value from this macro call is WORKSPACE.
programPath	The path of the program file.
programVersion	The version of the program file. A blank value specifies the latest version of the program file is used by the job.

%LSAF_GETWORKSPACEJOBSETUP Macro

Gets the metadata for the setup items for the specified job in the workspace of the current user. The metadata is stored in a SAS data set. The list does not include the setup programs.

Category: JobService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETWORKSPACEJOBSETUP(LSAF_PATH=*path*<, SAS_DSNAME=*name*>);

Required Argument

LSAF_PATH=*path*
The path of the job.

Optional Argument

SAS_DSNAME=*name*

The name of the output data set that contains the metadata for the setup. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETWORKSPACEJOBSETUP.

The data set contains a row for each job setup item and columns with the following names. The data set retains the order of the setup items.

Column Name	Description
jobPath	The path of the job.
jobVersion	Versions are not relevant in the workspace. This variable is included for consistency with the data set that is returned by the %LSAF_GETJOBSETUP macro. The value from this macro call is WORKSPACE.
setupPath	The path of the setup file or container.
setupType	The item type for the setup item. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=ItemType`.
setupVersion	The version of the setup file. A blank value specifies the latest version of the setup file is used by the job.
setupIncludeSubFolders	Indicates whether the subfolders of a container setup are included during job execution.
setupIncludeFiles	Indicates whether the files of a container setup are included during job execution.

%LSAF_SETJOBRUNASOWNER Macro

Sets the job property Run as owner. Only the owner can change the property.

Category: JobService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_SETJOBRUNASOWNER(LSAF_PATH=*path*, LSAF_RUNASOWNER=0 | 1);

Required Arguments

LSAF_PATH=*path*
The case-sensitive path of the job.

LSAF_RUNASOWNER=0 | 1
The value for property.

Details

Sets the job property Run as owner. Only the owner can change the property. To get the current value of the property, call %LSAF_GETJOBRUNASOWNER.

%LSAF_UPDATEWORKSPACEJOB Macro

Updates a job in workspace of the current user.

Category: JobService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_UPDATEWORKSPACEJOB(LSAF_PATH=*path*<, SAS_DSNAME_INFO=*name*><, SAS_DSNAME_PROGRAMS=*name*><, SAS_DSNAME_PARAMETERS=*name*><, SAS_DSNAME_SETUP=*name*><, SAS_DSNAME_CHECKIN=*name*>);

Required Argument

LSAF_PATH=*path*
The path for the job to update in the workspace of the current user.

Optional Arguments

SAS_DSNAME_INFO=*name*
The name of the output data set that contains the job information. See “Data Set Macros” on page 5 for information about specifying the name of the data set.

The data set must contain at least the following columns. Additional variables are ignored.

Column Name	Description
name	The name of the job property. The information that can be specified is DOCUMENTATION, LOGSLOCATION, RESULTSLOCATION, MANIFESTLOCATION, NEWOUTPUTS_CUSTOMVERSION, NEWOUTPUTS_ENABLEVERSION, and NEWOUTPUTS_VERSIONTYPE. All other values listed in the data set are ignored. All data are case-insensitive. If a name is listed more than once, the last value processed is used. If a name is omitted, the value in the job is not updated.
value	The value of the property.

SAS_DSNAME_PROGRAMS=*name*

The name of the output data set that contains the list of the paths to the SAS programs to run. See "Data Set Macros" on page 5 for information about specifying the name of the data set. If the data set is empty, all programs are removed from the job.

The data set must contain at least the following columns. Additional variables are ignored.

Column Name	Description
programPath	The path of the SAS program file.
programVersion	The version of the SAS program file. Blank values specify the latest file version.

SAS_DSNAME_PARAMETERS=*name*

The name of the output data set that contains the job parameter information. See "Data Set Macros" on page 5 for information about specifying the name of the data set. If the data set is empty, all job parameters are removed from the job.

The data set must contain at least the following columns. Additional variables are ignored.

Column Name	Description
name	The name of the parameter.
label	The label for the parameter.
type	The parameter type. For valid values, call

Column Name	Description
	%LSAF_GETSYSTEMCONSTANTS, where `name=ParameterType`.
defaultValue	The default value for the parameter. See “Format Date Values” on page 4 for information about date formats.
fileVersion	If the parameter is type FILE, the file version. Blank values specify the latest file version.
includeSubFolders (numeric)	Indicates whether the container includes subfolders.

Data set rows with duplicate values for parameter name result in a data set validation error.

SAS_DSNAME_SETUP=*name*

The name of the output data set that contains the setup list of the job. See “Data Set Macros” on page 5 for information about specifying the name of the data set. If the data set is empty, all job setup items are removed from the job.

The data set must contain at least the following columns. Additional variables are ignored.

Column Name	Description
setupPath (character)	The path of the file or container to use as setup.
setupType (character)	The type of the setup item. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=ItemType`.
setupVersion (character)	The version of the FILE item. Blank values specify the latest file version.The value is ignored for setup items of type CONTAINER.
setupIncludeSubFolders (numeric)	Indicates whether the setup item folder includes subfolders. The value is ignored for setup items of type FILE.
setupIncludeFiles (numeric)	Indicates whether the setup includes files.

Data set rows with duplicate values for setupPath result in a data set validation error.

SAS_DSNAME_CHECKIN=*name*

The name of the output data set that contains the list of the job checkin locations. See "Data Set Macros" on page 5 for information about specifying the name of the data set. If the data set is empty, all job checkin locations are removed from the job.

The data set must contain at least the following columns. Additional variables are ignored.

Column Name	Description
checkinPath (character)	The checkin location path of the container.
checkinIncludeSubFolders (numeric)	Indicates whether the checkin location container includes subfolders.

Details

Updates a job in workspace of the current user.

Data sets that represent the job data to update must be specified in the call to %LSAF_UPDATEWORKSPACEJOB. The information in the data sets replaces the data of the existing job. The output data sets from the following macros contain all of the data that is relevant to the job properties that can be updated:

- "%LSAF_GETJOBINFO Macro"
- "%LSAF_GETWORKSPACEJOBINFO Macro"
- "%LSAF_GETJOBPROGRAMS Macro"
- "%LSAF_GETWORKSPACEJOBPROGRAMS Macro"
- "%LSAF_GETJOBPARAMETERS Macro"
- "%LSAF_GETWORKSPACEJOBPARAMETERS Macro"
- "%LSAF_GETJOBSETUP Macro"
- "%LSAF_GETWORKSPACEJOBSETUP Macro"
- "%LSAF_GETJOBCHECKINLOCATIONS Macro"
- "%LSAF_GETWORKSPACEJOBCHECKINLOCS Macro"

The output data sets from the following job manifest file macros contain all of the data that is relevant to the job properties that can be specified when the job is updated:

- "%LSAF_GETJOBMANIFESTINFO Macro"
- "%LSAF_GETJOBMANIFESTPROGRAMS Macro"
- "%LSAF_GETJOBMANIFESTPARAMETERS Macro"
- "%LSAF_GETJOBMANIFESTOUTPUTS Macro"

22

JobSubmissionService Module

Overview

Macros to execute jobs and to report the status of a submitted job.

Macro Summary Table

Category	Language Elements	Description
JobSubmissionService Module	%LSAF_GETJOBPUBLISHHISTORY Macro (p. 274)	Gets the publish history for the specified job in the repository. The metadata is stored in a SAS data set.
	%LSAF_GETJOBSUBMISSIONBYSESSIONID Macro (p. 276)	Gets the properties of a job submission by the session identifier. The properties are stored in a SAS data set.
	%LSAF_GETSUBMISSIONSTATUS Macro (p. 277)	Gets the status of a repository or workspace job submission.

Category	Language Elements	Description
	%LSAF_SUBMITANDPOPULATEWSJOB Macro (p. 277)	Submits a workspace job for immediate execution, and updates the input and output specifications in the job based on inputs and outputs that are accessed during the job run.
	%LSAF_SUBMITJOB Macro (p. 278)	Submits a repository job for immediate execution.
	%LSAF_SUBMITWORKSPACEJOB Macro (p. 279)	Submits a workspace job for immediate execution.

Dictionary

%LSAF_GETJOBPUBLISHHISTORY Macro

Gets the publish history for the specified job in the repository. The metadata is stored in a SAS data set.

Category: JobSubmissionService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETJOBPUBLISHHISTORY(LSAF_PATH=*path*<, SAS_DSNAME=*name*>);

Required Argument

LSAF_PATH=*path*
The case-sensitive path of the job file.

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the publish history. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETJOBPUBLISHHISTORY.

The data set contains a row for each job publish history and columns with the following names:

Column Name	Description
path	The path of the job file.
id	The submission identifier of the job file.
jobId	The identifier of the job file.
version	The version of the job file that is associated with the submission.
sessionId	The session identifier of the job submission.
submittedBy	The identifier of the user who submitted the job.
submittedDate	The date on which the job was submitted.
submittedDateSAS	The SAS formatted date on which the job was submitted, in DATETIME format.
completedDate	The date on which the submitted job was completed.
completedDateSAS	The SAS formatted date on which the submitted job was completed, in DATETIME format.
summaryStatus	The summary status of the job submission.
detailStatus	The detail status of the job submission.
owner	The identifier of the owner of the job.
runAsUser	The identifier of the run-as user account that is associated with running the job.

Note: See "Format Date Values" on page 4 for information about date formats.

%LSAF_GETJOBSUBMISSIONBYSESSIONID Macro

Gets the properties of a job submission by the session identifier. The properties are stored in a SAS data set.

Category: JobSubmissionService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETJOBSUBMISSIONBYSESSIONID(LSAF_SESSIONID=*session-id*<, SAS_DSNAME=*name*>);

Required Argument

LSAF_SESSIONID=*session-id*

The session identifier of the job submission. To get the session identifier, call %LSAF_GETSESSIONSBYQUERY.

Optional Argument

SAS_DSNAME=*name*

The name of the output data set that contains the metadata for the job submission. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFJOBSUBMISSIONBYSESSIONID.

The data set contains the following variables that represent each element of metadata for the submission, sorted by the NAME variable:

Column Name	Description
sessionId	The job submission session identifier.
name	The name of the property.
value	The value of the property.

%LSAF_GETSUBMISSIONSTATUS Macro

Gets the status of a repository or workspace job submission.

Category: JobSubmissionService Module

Note: This macro sets the standard macro variables and _lsafJobSubmissionStatus_, which is the job submission status.

Syntax

%LSAF_GETSUBMISSIONSTATUS(LSAF_JOBSUBMISSION_ID=*submission-id*);

Required Argument

LSAF_JOBSUBMISSION_ID=*submission-id*
The submission identifier returned by the %LSAF_SUBMITJOB macro.

Details

Sets the value of the macro variable _lsafJobSubmissionStatus_ to the job submission status. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=DetailStatus`. See "Value Macros" on page 5 for information about value macros.

If the macro runs without error, _lsafRC_ is set to 0, the _lsafJobSubmissionStatus_ macro variable is set to the appropriate value, and a job submission status message is printed to the SAS log file. If the job completed with warnings or errors, the job submission message shows in the SAS log file as a warning or an error, as appropriate.

%LSAF_SUBMITANDPOPULATEWSJOB Macro

Submits a workspace job for immediate execution, and updates the input and output specifications in the job based on inputs and outputs that are accessed during the job run.

Category: JobSubmissionService Module

Note: This macro sets the standard macro variables and _lsafPopulateWsJobSummissionId_, which is the job submission identifier.

Syntax

%LSAF_SUBMITANDPOPULATEWSJOB(LSAF_PATH=*path*<, LSAF_USERELATIVEPATHS=0 | 1><, SAS_DSNAME=*name*>);

Required Argument

LSAF_PATH=*path*
The path of the job in the workspace of the current user.

Optional Arguments

LSAF_USERELATIVEPATHS=0 (Default) | 1
Indicates whether to generate inputs and outputs as absolute or relative paths.

SAS_DSNAME=*name*
The name of the data set that contains the job parameter override values. See "Data Set Macros" on page 5 for information about specifying the name of the data set. See "Job Parameter Override Values" on page 7 for information about job parameter override variables.

Details

Submits a workspace job for immediate execution, and updates the input and output specifications in the job based on inputs and outputs that are accessed during the job run. Sets the value of the macro variable _lsafPopulateWsJobSummissionId_ to the identifier for the job submission. See "Value Macros" on page 5 for information about value macros.

If the job runs with errors, the job is not updated. If the job runs successfully or with warnings, it is updated, and it replaces existing input and output definitions in the job.

%LSAF_SUBMITJOB Macro

Submits a repository job for immediate execution.

Category: JobSubmissionService Module

Note: This macro sets the standard macro variables and _lsafJobSubmissionId_, which is the job submission identifier.

Syntax

%LSAF_SUBMITJOB(LSAF_PATH=*path*<, LSAF_VERSION=*version*><, SAS_DSNAME=*name*>);

Required Argument

LSAF_PATH=*path*
The path of the job in the repository.

Optional Arguments

LSAF_VERSION=*version*
The specific version number of the job. See “LSAF_VERSION=version” on page 6 for more information about this argument.

SAS_DSNAME=*name*
The name of the data set that contains the job parameter override values. See “Data Set Macros” on page 5 for information about specifying the name of the data set. See “Job Parameter Override Values” on page 7 for information about job parameter override variables.

Details

Submits a repository job for immediate execution and sets the value of the macro variable _IsafJobSubmissionId_ to the identifier for the job submission. See “Value Macros” on page 5 for information about value macros.

%LSAF_SUBMITWORKSPACEJOB Macro

Submits a workspace job for immediate execution.

Category: JobSubmissionService Module

Note: This macro sets the standard macro variables and _IsafWorkspaceJobSubmissionId_, which is the job submission identifier.

Syntax

%LSAF_SUBMITWORKSPACEJOB(LSAF_PATH=*path*<, LSAF_RUNASTEST=0 (Default) | 1><, SAS_DSNAME=*name*>);

Required Argument

LSAF_PATH=*path*
The path of the job in the workspace of the current user.

Optional Arguments

LSAF_RUNASTEST=0 (Default) | 1

If 1, the job is submitted like a published job, which accesses items in the repository, except output is not checked in. Any output from the job is written to the current user's workspace. The test submission checks that the job is set up correctly in the repository and that the items that are required to run the job are accessible. A manifest file is generated to indicate the inputs that were referenced and the outputs that were generated.

SAS_DSNAME=*name*

The name of the data set that contains the job parameter override values. See “Data Set Macros” on page 5 for information about specifying the name of the data set. See “Job Parameter Override Values” on page 7 for information about job parameter override variables.

Details

Submits a workspace job for immediate execution, and sets the value of the macro variable _lsafWorkspaceJobSubmissionId_ to the identifier for the job submission. See “Value Macros” on page 5 for information about value macros.

23

ScheduleService Module

Overview

Macros to manage and create schedules.

Macro Summary Table

Category	Language Elements	Description
ScheduleService Module	%LSAF_CREATEINTERVALSCHEDULE Macro (p. 283)	Creates a schedule that runs at the specified interval.
	%LSAF_CREATEONETIMESCHEDULE Macro (p. 284)	Creates a schedule that runs one time.
	%LSAF_CREATEWEEKLYSCHEDULE Macro (p. 285)	Creates a schedule that runs on the specified weekdays.
	%LSAF_DELETESCHEDULE Macro (p. 287)	Deletes a schedule.
	%LSAF_GETALLSCHEDULES Macro (p. 287)	Gets the metadata for all of the schedules. The metadata is stored in a SAS data set.
	%LSAF_GETMYSCHEDULES Macro (p. 288)	Gets the metadata for the schedules that are owned by the current user. The metadata is stored in a SAS data set.
	%LSAF_GETSCHEDULEJOBPARAMETERS Macro (p. 289)	Gets the override the job parameters for the schedule. The parameters are stored in a SAS data set.
	%LSAF_GETSCHEDULEPROPERTIES Macro (p. 291)	Gets the properties for the specified schedule. The properties are stored in a SAS data set.
	%LSAF_GETSCHEDULESTATUS Macro (p. 292)	Gets the status of a schedule.
	%LSAF_ISJOBSCHEDULED Macro (p. 292)	Indicates whether a job is scheduled.
	%LSAF_PAUSESCHEDULE Macro (p. 293)	Pauses an active schedule.
	%LSAF_RESUMESCHEDULE Macro (p. 293)	Resumes a paused schedule.
	%LSAF_SCHEDULEEXISTSBYID Macro (p. 294)	Indicates whether a schedule with the specified identifier exists.
	%LSAF_SCHEDULEEXISTSBYNAME Macro (p. 294)	Indicates whether a schedule with the specified name exists.

Dictionary

%LSAF_CREATEINTERVALSCHEDULE Macro

Creates a schedule that runs at the specified interval.

Category: ScheduleService Module

Note: This macro sets the standard macro variables and _lsafIntervalScheduleId_, which is the identifier for the created schedule.

Syntax

%LSAF_CREATEINTERVALSCHEDULE(LSAF_INTERVALTYPE=*interval-type*, LSAF_INTERVAL=*interval*, LSAF_NAME=*name*, LSAF_LOCATION=WORKSPACE | REPOSITORY, LSAF_PATH=*path*, LSAF_STARTDATE=*date*, LSAF_ENDTYPE=*end-type*<, LSAF_DESCRIPTION=*description*><, LSAF_VERSION=*version*><, SAS_DSNAME=*name*><, LSAF_ENDVALUE=*value*>);

Required Arguments

LSAF_INTERVALTYPE=*interval-type*
The case-insensitive type of interval. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=ScheduleIntervalType`.

LSAF_INTERVAL=*interval*
The integer value for the interval to pass before each schedule run.

LSAF_NAME=*name*
The name of the schedule to create.

LSAF_LOCATION=WORKSPACE | REPOSITORY
The case-insensitive location of the job to schedule.

LSAF_PATH=*path*
The path of the job to schedule.

LSAF_STARTDATE=*date*
The date and time on which the schedule runs begin. See “Format Date Values” on page 4 for information about date formats.

LSAF_ENDTYPE=*end-type*
The case-insensitive value that specifies the method to end the schedule. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=ScheduleEndType`.

Optional Arguments

LSAF_DESCRIPTION=*description*
The description of the schedule to create.

LSAF_VERSION=*version*
The version of the repository job to schedule. See “LSAF_VERSION=version” on page 6 for more information about this argument.

SAS_DSNAME=*name*
The name of the data set that contains the job parameter override values to use when the schedule runs. See “Data Set Macros” on page 5 for information about specifying the name of the data set. See “Job Parameter Override Values” on page 7 for information about job parameter override variables.

LSAF_ENDVALUE=*value*
If LSAF_ENDTYPE is DATE, this value must be a valid date for the last schedule run. See “Format Date Values” on page 4 for information about date formats. If LSAF_ENDTYPE is OCCURRENCES, this value must be a valid integer value for the number of times to run the schedule. If LSAF_ENDTYPE is NONE, this value is ignored.

Details

Creates a schedule that runs at the specified interval and sets the value of the macro variable _lsafIntervalScheduleId_ to the identifier for the created schedule. See “Value Macros” on page 5 for information about value macros.

%LSAF_CREATEONETIMESCHEDULE Macro

Creates a schedule that runs one time.

Category: ScheduleService Module

Note: This macro sets the standard macro variables and _lsafOneTimeScheduleId_, which is the identifier for the created schedule.

Syntax

%LSAF_CREATEONETIMESCHEDULE(LSAF_NAME=*name*, LSAF_LOCATION=WORKSPACE | REPOSITORY, LSAF_PATH=*path*, LSAF_STARTDATE=*date*<, LSAF_DESCRIPTION=*description*><, LSAF_VERSION=*version*><, SAS_DSNAME=*name*>);

Required Arguments

LSAF_NAME=*name*
The name of the schedule to create.

LSAF_LOCATION=WORKSPACE | REPOSITORY
The case-insensitive location of the job to schedule.

LSAF_PATH=*path*
The path of the job to schedule.

LSAF_STARTDATE=*date*
The date and time on which the schedule runs. See “Format Date Values” on page 4 for information about date formats.

Optional Arguments

LSAF_DESCRIPTION=*description*
The description of the schedule to create.

LSAF_VERSION=*version*
The version of the repository job to schedule. See “LSAF_VERSION=version” on page 6 for more information about this argument.

SAS_DSNAME=*name*
The name of the data set that contains the job parameter override values to use when the schedule runs. See “Data Set Macros” on page 5 for information about specifying the name of the data set. See “Job Parameter Override Values” on page 7 for information about job parameter override variables.

Details

Creates a schedule that runs one time and sets the value of the macro variable _lsafOneTimeScheduleId_ to the identifier for the created schedule. See “Value Macros” on page 5 for information about value macros.

%LSAF_CREATEWEEKLYSCHEDULE Macro

Creates a schedule that runs on the specified weekdays.

Category: ScheduleService Module

Note: This macro sets the standard macro variables and _lsafWeeklyScheduleId_, which is the identifier for the created schedule.

Syntax

%LSAF_CREATEWEEKLYSCHEDULE(LSAF_NAME=*name*, LSAF_LOCATION=WORKSPACE | REPOSITORY, LSAF_PATH=*path*, LSAF_STARTDATE=*date*, LSAF_ENDTYPE=*end-type*, LSAF_DAYSOFWEEK=*days-of-week* <, LSAF_DESCRIPTION=*description*><, LSAF_VERSION=*version*><, SAS_DSNAME=*name*><, LSAF_ENDVALUE=*value*>);

Required Arguments

LSAF_NAME=*name*
The name of the schedule to create.

LSAF_LOCATION=WORKSPACE | REPOSITORY
The case-insensitive location of the job to schedule.

LSAF_PATH=*path*
The path of the job to schedule.

LSAF_STARTDATE=*date*
The date and time on which the schedule runs begin. See “Format Date Values” on page 4 for information about date formats.

LSAF_ENDTYPE=*end-type*
The case-insensitive value that specifies the method to end the schedule. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=ScheduleEndType`.

LSAF_DAYSOFWEEK=*days-of-week*
A case-insensitive, comma-delimited list of the days of the week on which to run the schedule. The days can be listed in any order. At least one day must be specified. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=DayOfWeek`.

Note: If specifying all weekdays, consider calling %LSAF_CREATEINTERVALSCHEDULE and specifying DAILY as the LSAF_INTERVALTYPE.

Optional Arguments

LSAF_DESCRIPTION=*description*
The description of the schedule to create.

LSAF_VERSION=*version*
The version of the repository job to schedule. See “LSAF_VERSION=version” on page 6 for more information about this argument.

SAS_DSNAME=*name*
The name of the data set that contains the job parameter override values to use when the schedule runs. See “Data Set Macros” on page 5 for information about specifying the name of the data set. See “Job Parameter Override Values” on page 7 for information about job parameter override variables.

LSAF_ENDVALUE=*value*
If LSAF_ENDTYPE is DATE, this value must be a valid date for the last schedule run. See “Format Date Values” on page 4 for information about date formats. If LSAF_ENDTYPE is OCCURERENCES, this value must be a valid integer value for the number of times to run the schedule. This value is ignored when LSAF_ENDTYPE is NONE.

Details

Creates a schedule that runs on the specified weekdays and sets the value of the macro variable _lsafWeeklyScheduleId_ to the identifier for the created schedule. See “Value Macros” on page 5 for information about value macros.

%LSAF_DELETESCHEDULE Macro

Deletes a schedule.

Category: ScheduleService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_DELETESCHEDULE(LSAF_SCHEDULEID=*schedule-id*);

Required Argument

LSAF_SCHEDULEID=*schedule-id*
The case-sensitive identifier of the schedule. To get the identifier, call %LSAF_GETALLSCHEDULES or %LSAF_GETMYSCHEDULES.

%LSAF_GETALLSCHEDULES Macro

Gets the metadata for all of the schedules. The metadata is stored in a SAS data set.

Category: ScheduleService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETALLSCHEDULES(<SAS_DSNAME=*name*>);

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the schedules. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETALLSCHEDULES.

The data set contains a row for each schedule deployed, sorted by schedule identifier, and the following columns:

Column Name	Description
scheduleID	The identifier of the schedule.
name	The name of the schedule.
description	The description of the schedule.
owner	The owner of the schedule.
status	The status of the schedule.
lastModified	The date on which the schedule was last modified, represented as a String.
dateLastModified	The date on which the schedule was last modified, represented as a SAS Datetime format.

Note: See “Format Date Values” on page 4 for information about date formats.

Details

Gets the metadata for all of the schedules. The metadata is stored in a SAS data set.

The global privilege PRIVILEGE_MANAGE_ALL_SCHEDULES is required to run this macro.

%LSAF_GETMYSCHEDULES Macro

Gets the metadata for the schedules that are owned by the current user. The metadata is stored in a SAS data set.

Category: ScheduleService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETMYSCHEDULES(<SAS_DSNAME=*name*>);

Optional Argument

SAS_DSNAME=*name*

The name of the output data set that contains the metadata for the schedules. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETMYSCHEDULES.

The data set contains a row for each schedule deployed, sorted by schedule identifier, and the following columns:

Column Name	Description
scheduleID	The identifier of the schedule.
name	The name of the schedule.
description	The description of the schedule.
owner	The owner of the schedule.
status	The status of the schedule.
lastModified	The date on which the schedule was last modified, represented as a String.
dateLastModified	The date on which the schedule was last modified, represented as a SAS Datetime format.

Note: See "Format Date Values" on page 4 for information about date formats.

%LSAF_GETSCHEDULEJOBPARAMETERS Macro

Gets the override the job parameters for the schedule. The parameters are stored in a SAS data set.

Category: ScheduleService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETSCHEDULEJOBPARAMETERS(LSAF_SCHEDULEID=*schedule-id*<, SAS_DSNAME=*name*>);

Required Argument

LSAF_SCHEDULEID=*schedule-id*
The case-sensitive identifier of the schedule. To get the identifier, call %LSAF_GETALLSCHEDULES or %LSAF_GETMYSCHEDULES.

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the override parameters for schedule. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETSCHEDULEJOBPARAMETERS.

The data set contains a row for each override parameter and columns with the following names:

Column Name	Description
scheduleID	The identifier of the schedule.
scheduleName	The name of the schedule.
jobLocation	The location of the job. Valid values: ■ REPOSITORY ■ WORKSPACE
jobPath	The path for the job.
jobVersion	The version of the job.
name	The name of the parameter.
type	The type of the parameter. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=ParameterType`.
value	The value of the parameter.
fileVersion	If the parameter is type FILE, the file version. A blank value specifies that either the file is not versioned or the file is versioned but no version was specified.
includeSubFolders	Indicates whether the subfolders of a container are included during job execution.

Details

Gets the override parameters for the schedule. The parameters are stored in a SAS data set.

Job schedules maintain a list of the parameters that are used to override the default parameters when the job executes. The output data set from the %LSAF_GETJOBPARAMETERS macro can be used to get the full list and metadata for the job parameters.

%LSAF_GETSCHEDULEPROPERTIES Macro

Gets the properties for the specified schedule. The properties are stored in a SAS data set.

Category: ScheduleService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETSCHEDULEPROPERTIES(LSAF_SCHEDULEID=*schedule-id*<, SAS_DSNAME=*name*>);

Required Argument

LSAF_SCHEDULEID=*schedule-id*

The case-sensitive identifier of the schedule. To get the identifier, call %LSAF_GETALLSCHEDULES or %LSAF_GETMYSCHEDULES.

Optional Argument

SAS_DSNAME=*name*

The name of the output data set that contains the metadata for the schedule. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETSCHEDULEPROPERTIES.

The data set contains the following columns that represent each element of the metadata for the schedule, sorted by the NAME variable:

Column Name	Description
scheduleId	The identifier of the schedule.
scheduleName	The name of the schedule.
name	The name of the property.
value	The value of the property.

%LSAF_GETSCHEDULESTATUS Macro

Gets the status of a schedule.

Category: ScheduleService Module

Note: This macro sets the standard macro variables and _lsafScheduleStatus_, which is the status of the schedule.

Syntax

%LSAF_GETSCHEDULESTATUS(LSAF_SCHEDULEID=*schedule-id*);

Required Argument

LSAF_SCHEDULEID=*schedule-id*
The case-sensitive identifier of the schedule. To get the identifier, call %LSAF_GETALLSCHEDULES or %LSAF_GETMYSCHEDULES.

Details

Sets the value of the macro variable _lsafScheduleStatus_ to the status of the schedule. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=ScheduleStatus`. See "Value Macros" on page 5 for information about value macros.

%LSAF_ISJOBSCHEDULED Macro

Indicates whether a job is scheduled.

Category: ScheduleService Module

Note: This macro sets the standard macro variables and _lsafJobIsScheduled_, which indicates whether the job is scheduled.

Syntax

%LSAF_ISJOBSCHEDULED(LSAF_LOCATION=WORKSPACE | REPOSITORY, LSAF_PATH=*path*);

Required Arguments

LSAF_LOCATION=WORKSPACE | REPOSITORY
The case-insensitive location of the job.

LSAF_PATH=*path*
The path of the job.

Details

Sets the Boolean macro variable _lsafJobIsScheduled_, which indicates whether a job is scheduled. See “Boolean Macros” on page 4 for information about return values.

If you do not have access permission to the schedule, the macro returns False. Access permission to the schedules is controlled by the global privileges PRIVILEGE_MANAGE_SCHEDULES and PRIVILEGE_MANAGE_ALL_SCHEDULES.

%LSAF_PAUSESCHEDULE Macro

Pauses an active schedule.

Category: ScheduleService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_PAUSESCHEDULE(LSAF_SCHEDULEID=*schedule-id*);

Required Argument

LSAF_SCHEDULEID=*schedule-id*
The case-sensitive identifier of the schedule. To get the identifier, call %LSAF_GETALLSCHEDULES or %LSAF_GETMYSCHEDULES.

%LSAF_RESUMESCHEDULE Macro

Resumes a paused schedule.

Category: ScheduleService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_RESUMESCHEDULE(LSAF_SCHEDULEID=*schedule-id*);

Required Argument

LSAF_SCHEDULEID=*schedule-id*
The case-sensitive identifier of the schedule. To get the identifier, call %LSAF_GETALLSCHEDULES or %LSAF_GETMYSCHEDULES.

%LSAF_SCHEDULEEXISTSBYID Macro

Indicates whether a schedule with the specified identifier exists.

Category: ScheduleService Module

Note: This macro sets the standard macro variables and _lsafScheduleExistsById_, which indicates whether the schedule exists.

Syntax

LSAF_SCHEDULEEXISTSBYID(LSAF_SCHEDULEID=*schedule-id*);

Required Argument

LSAF_SCHEDULEID=*schedule-id*
The case-sensitive identifier of the schedule. To get the identifier, call %LSAF_GETALLSCHEDULES or %LSAF_GETMYSCHEDULES.

Details

Sets the Boolean macro variable _lsafScheduleExistsById_, which indicates whether a schedule with the specified identifier exists. See “Boolean Macros” on page 4 for information about return values.

If you do not have access permission to the schedule, the macro returns False. Access permission to the schedules is controlled by the global privileges PRIVILEGE_MANAGE_SCHEDULES and PRIVILEGE_MANAGE_ALL_SCHEDULES.

%LSAF_SCHEDULEEXISTSBYNAME Macro

Indicates whether a schedule with the specified name exists.

Category: ScheduleService Module

Note: This macro sets the standard macro variables and _lsafScheduleExistsByName_, which indicates whether the schedule exists.

Syntax

%LSAF_SCHEDULEEXISTSBYNAME(LSAF_NAME=*name*);

Required Argument

LSAF_NAME=*name*
The name of the schedule.

Details

Sets the Boolean macro variable _lsafScheduleExistsByName_, which indicates whether a schedule with the specified name exists. See “Boolean Macros” on page 4 for information about return values.

Note: Schedule names are not unique. If a schedule with the specified name exists, this macro returns True.

If you do not have access permission to the schedule, the macro returns False. Access permission to schedules is controlled by the global privileges PRIVILEGE_MANAGE_SCHEDULES and PRIVILEGE_MANAGE_ALL_SCHEDULES.

24

SessionService Module

Overview

Macros to manage and create interactive sessions.

Macro Summary Table

Category	Language Elements	Description
SessionService Module	%LSAF_DELETESESSION Macro (p. 298)	Deletes a session.
	%LSAF_ENDSESSION Macro (p. 299)	Ends a session.

Category	Language Elements	Description
	%LSAF_GETSESSIONLISTSIZE Macro (p. 299)	Returns the list size of a session. The macro variable _lsafSessionListSize_ contains the value.
	%LSAF_GETSESSIONLOGSIZE Macro (p. 300)	Returns the log size of a session.
	%LSAF_GETSESSIONSBYQUERY Macro (p. 301)	Gets the metadata for all of the sessions that are based on a user-specified query. The metadata is stored in a SAS data set.
	%LSAF_GETSESSIONSTATE Macro (p. 303)	Returns the state of a session.
	%LSAF_SAVESESSIONLOG Macro (p. 303)	Saves the log of the specified session to the user's workspace.
	%LSAF_SAVESESSIONRESULTS Macro (p. 304)	Saves the results of the specified session to the user's workspace.
	%LSAF_SESSIONEXISTS Macro (p. 305)	Indicates whether a session exists.

Dictionary

%LSAF_DELETESESSION Macro

Deletes a session.

Category: SessionService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_DELETESESSION(LSAF_SESSIONID=*session-id*);

Required Argument

LSAF_SESSIONID=*session-id*
The case-sensitive identifier of the session. To get the identifier, call %LSAF_GETSESSIONSBYQUERY.

%LSAF_ENDSESSION Macro

Ends a session.

Category: SessionService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_ENDSESSION(LSAF_SESSIONID=*session-id*);

Required Argument

LSAF_SESSIONID=*session-id*
The case-sensitive identifier of the session. To get the identifier, call %LSAF_GETSESSIONSBYQUERY.

%LSAF_GETSESSIONLISTSIZE Macro

Returns the list size of a session. The macro variable _lsafSessionListSize_ contains the value.

Category: SessionService Module

Note: This macro sets the standard macro variables and _lsafSessionListSize_, which is the size of the session list.

Syntax

%LSAF_GETSESSIONLISTSIZE(LSAF_SESSIONID=*session-id*<, LSAF_FORMATVALUE=0 | 1>);

Required Argument

LSAF_SESSIONID=*session-id*
The case-sensitive identifier of the session. To get the identifier, call %LSAF_GETSESSIONSBYQUERY.

Optional Argument

LSAF_FORMATVALUE=0 (Default) | 1
Indicates whether to format the value. Non-formatted returns the raw numeric value in bytes. Formatted output sets the value to the character representation with the appropriate label.

For example, the non-formatted value 1024 would be 1 KB when formatted.

Details

Sets the value of the macro variable _lsafSessionListSize_ to the size of the session list. See “Value Macros” on page 5 for information about value macros.

%LSAF_GETSESSIONLOGSIZE Macro

Returns the log size of a session.

Category: SessionService Module

Note: This macro sets the standard macro variables and _lsafSessionLogSize_, which is the size of the session log.

Syntax

%LSAF_GETSESSIONLOGSIZE(LSAF_SESSIONID=*session-id*<, LSAF_FORMATVALUE=0 | 1>);

Required Argument

LSAF_SESSIONID=*session-id*
The case-sensitive identifier of the session. To get the identifier, call %LSAF_GETSESSIONSBYQUERY.

Optional Argument

LSAF_FORMATVALUE=0 (Default) | 1
Indicates whether to format the value. Non-formatted returns the raw numeric value in bytes. Formatted output sets the value to the character representation with the appropriate label.

For example, the non-formatted value 1024 would be 1 KB when formatted.

Details

Sets the value of the macro variable _lsafSessionLogSize_ to the size of the session log. See “Value Macros” on page 5 for information about value macros.

%LSAF_GETSESSIONSBYQUERY Macro

Gets the metadata for all of the sessions that are based on a user-specified query. The metadata is stored in a SAS data set.

Category: SessionService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETSESSIONSBYQUERY(<LSAF_USERID=*user-id*><, LSAF_ENGINE=*session-engine*><, LSAF_TYPE=*session-type*><, LSAF_STATE=*session-state*><, LSAF_SUBMISSIONLABEL=*label*><, LSAF_STARTEDFROMDATE=*date*><, LSAF_STARTEDTODATE=*date*><, LSAF_ENDEDFROMDATE=*date*><, LSAF_ENDEDTODATE=*date*><, LSAF_SESSIONLIMIT=*limit*><, SAS_DSNAME=*name*>);

Optional Arguments

LSAF_USERID=*user-id*
The identifier of the user who owns the session. If a value is not specified, the sessions of all users are returned.

LSAF_ENGINE=*session-engine*
The session engine to return from the search. If a value is not specified, all engines are returned. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=SessionEngine`.

LSAF_TYPE=*session-type*
The session type to return from the search. If a value is not specified, all types are returned. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=SessionType`.

LSAF_STATE=*session-state*
The session state to return from the search. If a value is not specified, all session states are returned. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=SessionState`.

LSAF_SUBMISSIONLABEL=*label*
The name that is associated with a submission. If a value is not specified, all sessions are returned.

LSAF_STARTEDFROMDATE=*date*
The started from date range for sessions within a specific time period. See “Format Date Values” on page 4 for information about date formats.

LSAF_STARTEDTODATE=*date*
The started to date range for sessions within a specific time period. See “Format Date Values” on page 4 for information about date formats.

LSAF_ENDEDFROMDATE=*date*
The ended from date range for sessions within a specific time period. See “Format Date Values” on page 4 for information about date formats.

LSAF_ENDEDTODATE=*date*
The ended to date range for sessions within a specific time period. See “Format Date Values” on page 4 for information about date formats.

LSAF_SESSIONLIMIT=*limit*
The maximum number of sessions to return at one time. The default is 5000. A message is printed to the SAS log file to indicate whether the limit is reached. If True, there might be more sessions that meet the search criteria. A more refined search is needed to retrieve those records.

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for all the sessions. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETSESSIONSBYQUERY.

The data set contains a row for each session and columns with the following names:

Column Name	Description
sessionId	The unique identifier for the session.
dateStarted	The date on which the session started.
dateStartedSASFormat	The date on which the session started, in SAS DATETIME format.
dateEnded	The date on which the session ended.
dateEndededSASFormat	The date on which the session started, in SAS DATETIME format.
sessionType	The session type.
sessionOwner	The session owner.
sessionState	The session state.
submissionLabel	The label of the session.
clientId	The incremental number that is assigned to a session.

Note: See “Format Date Values” on page 4 for information about date formats.

Details

Gets the metadata for all of the sessions that are based on a user-specified query.The metadata is stored in a SAS data set.

The privilege PRIVILEGE_MANAGE_ALL_SESSIONS is required to retrieve sessions that are not owned by you.

%LSAF_GETSESSIONSTATE Macro

Returns the state of a session.

Category: SessionService Module

Note: This macro sets the standard macro variables and _lsafSessionState_, which is the state.

Syntax

%LSAF_GETSESSIONSTATE(LSAF_SESSIONID=*session-id*);

Required Argument

LSAF_SESSIONID=*session-id*
The case-sensitive identifier of the session. To get the identifier, call %LSAF_GETSESSIONSBYQUERY.

Details

Sets the value of the macro variable _lsafSessionState_ to the state. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=SessionState`. See “Value Macros” on page 5 for information about value macros.

%LSAF_SAVESESSIONLOG Macro

Saves the log of the specified session to the user’s workspace.

Category: SessionService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_SAVESESSIONLOG(LSAF_PATH=*lsaf-path*, LSAF_SESSIONID=*session-id*<, LSAF_ENGINE=*engine-type*>);

Required Arguments

LSAF_PATH=*lsaf-path*
The path where the log is saved.

LSAF_SESSIONID=*session-id*
The case-sensitive identifier for the session. To get the identifier, call %LSAF_GETSESSIONSBYQUERY.

Optional Argument

LSAF_ENGINE=*engine-type*
The engine type. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=EngineType`. The default value is SAS.

%LSAF_SAVESESSIONRESULTS Macro

Saves the results of the specified session to the user's workspace.

Category: SessionService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_SAVESESSIONRESULTS(LSAF_PATH=*lsaf-path*, LSAF_SESSIONID=*session-id*<, LSAF_ENGINE=*engine-type*>);

Required Arguments

LSAF_PATH=*lsaf-path*
The path where the results are saved.

LSAF_SESSIONID=*session-id*
The case-sensitive identifier for the session. To get the identifier, call %LSAF_GETSESSIONSBYQUERY.

Optional Argument

LSAF_ENGINE=*engine-type*
The engine type. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=EngineType`. The default value is SAS.

%LSAF_SESSIONEXISTS Macro

Indicates whether a session exists.

Category: SessionService Module

Note: This macro sets the standard macro variables and _lsafSessionExists_, which indicates whether the session exists.

Syntax

%LSAF_SESSIONEXISTS(LSAF_SESSIONID=*session-id*);

Required Argument

LSAF_SESSIONID=*session-id*
The case-sensitive identifier of the session. To get the identifier, call %LSAF_GETSESSIONSBYQUERY.

Details

Sets the Boolean macro variable _lsafSessionExists_, which indicates whether a session exists. See “Boolean Macros” on page 4 for information about return values.

If you do not have access permission to the session, the macro returns False. Access permission to sessions is controlled by the global privileges PRIVILEGE_CREATE_SASSESSION and PRIVILEGE_MANAGE_ALL_SASSESSIONS.

PART 6

Messaging Macros

25

SubscriptionService Module

Overview

Macros to manage subscriptions.

Macro Summary Table

Category	Language Elements	Description
SubscriptionService Module	%LSAF_GETSUBSCRIPTIONS Macro (p. 310)	Gets the metadata for all subscriptions for the current user. The metadata is stored in a SAS data set.
	%LSAF_SETITEMSUBSCRIPTIONS Macro (p. 311)	Sets the event subscriptions for a repository item. Previous subscriptions are overwritten.

Dictionary

%LSAF_GETSUBSCRIPTIONS Macro

Gets the metadata for all subscriptions for the current user. The metadata is stored in a SAS data set.

Category: SubscriptionService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETSUBSCRIPTIONS(<LSAF_PATH=*path*><, SAS_DSNAME=*name*>);

Optional Arguments

LSAF_PATH=*path*
The path to the item in the repository. If omitted, all subscriptions for the current user are returned. If specified, subscriptions for the current user for only that path are returned.

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the subscriptions. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETSUBSCRIPTIONS.

The data set contains a row for each subscription, sorted by event and path, and the following columns:

Column Name	Description
event	The name of the event subscription. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=SubscriptionEvent`.
path	The path of the item to which a subscription applies.

%LSAF_SETITEMSUBSCRIPTIONS Macro

Sets the event subscriptions for a repository item. Previous subscriptions are overwritten.

Category: SubscriptionService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_SETITEMSUBSCRIPTIONS(LSAF_PATH=*path*<, LSAF_EVENTS=*event*><, SAS_DSNAME=*name*>);

Required Argument

LSAF_PATH=*path*
The name of the repository item.

Optional Arguments

LSAF_EVENTS=*event*
Job events apply to both containers and job files. File events are applicable to all item types. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=SubscriptionEvent`.

To unsubscribe to an item, omit or specify a blank value for LSAF_EVENTS. An invalid or non- applicable event results in a failure.

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the subscriptions that were set by the operation. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFSETITEMSUBSCRIPTIONS.

The data set contains a row for each subscription, sorted by event and path, and the following columns:

Column Name	Description
event	The name of the event subscription. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=SubscriptionEvent`.
path	The path of the item to which a subscription applies.

PART 7

Query Macros

26

QueryService Module

Overview

Macros to help construct and query data within SAS Life Science Analytics Framework. Query macros extract data based on a query built from an input data set. The extracted data is stored in a CSV file in the workspace or repository.

The input data set is validated prior to executing the query. The results of that validation are returned in an output data set. If the input data set validation succeeds with no errors, the resulting query is printed to the SAS log file.

To get information that concerns the necessary columnClass and columnName data, and their data types, call %LSAF_GETQUERYCOLUMNS.

You must have the proper privilege to execute the query, the required privileges for each query macro are listed in the macro details below. If the query results in more than one million records, the macro fails.

Global macro variables specific to each query contain the full path of the query results that are specified in a comma-separated file, which includes the file extension. If the query is not executed or if the macro processing results in a failure, the macro variable is blank.

For more information about setting up and running queries, including examples, see "Queries" in *SAS Life Science Analytics Framework: SAS Macro API User's Guide*.

Common Arguments

General Syntax of the Query Macros

Most of the query macros share the following common syntax:

%LSAF_QUERY*QUERY-NAME*(
LSAF_QUERYDATASET=*query-data-set-name*,
LSAF_VALIDATEDDATASET=*validated-data-set-name*,
LSAF_EXPORTLOCATION=REPOSITORY | WORKSPACE,
LSAF_EXPORTPATH=*export-path*
<, LSAF_OVERWRITE=0 | 1>
<, LSAF_ENABLEVERSIONING=0 | 1>
<, LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM>
<, LSAF_CUSTOMVERSION=*custom-version*>
<, LSAF_COMMENT=*comment*>);

Required Parameters

LSAF_QUERYDATASET=*query-data-set-name*

The name of the input data set that contains the query metadata. See "Data Set Macros" on page 5 for information about specifying the name of the data set.

The data set must contain at least the following columns. All other columns are ignored.

Column Name	Description
recordType	The type of query element the record represents. See "Input Data Set" in *SAS*

Column Name	Description
	Life Science Analytics Framework: SAS Macro API User's Guide for descriptions of values.
columnClass	The class of the column to use with the query operation associated with the record type. ColumnClass is not required when recordType is LOGICAL_OPERATOR and, if specified, is ignored. To determine the column names that apply to a query, call %LSAF_GETQUERYCOLUMNS.
columnName	The name of the column to use with the query operation associated with the record type. ColumnName is not required when recordType is LOGICAL_OPERATOR and, if specified, is ignored. The valid values depend on the query. To determine the column names that apply to a query, call %LSAF_GETQUERYCOLUMNS.
value	The value to use with the query operation associated with the record type. See "Input Data Set" in *SAS Life Science Analytics Framework: SAS Macro API User's Guide* for descriptions of values.
comparator	Indicates the method that the query uses to handle the constraint. Comparator is required when recordType is CONSTRAINT. If specified for any other record type, it is ignored. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=Operator` and `source=Comparison`.
isCaseSensitive	Indicates whether a column value is treated as case-sensitive. Case sensitivity is applicable only when recordType is ORDER_ASCENDING, ORDER_DESCENDING, or CONSTRAINT and the associated column is type STRING. Otherwise, the value, if specified, is ignored. If a value is not specified for applicable record or

Column Name	Description
	column types, the default value is applied. The default value is 1 (true).

LSAF_VALIDATEDDATASET=*validated-data-set-name*
The name of the output data set that contains the results of the validation of the records from the input data set LSAF_QUERYDATASET. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default data set name varies with each macro.

The data set contains all the columns and unaltered row data from the LSAF_QUERYDATASET plus the following additional columns:

Column Name	Description
columnType	Indicates the data type of the associated column. This value is blank when recordType is LOGICAL_OPERATOR.
validationNote	A note that explains why the validation failed or, if no error was found, blank.

LSAF_EXPORTLOCATION=REPOSITORY | WORKSPACE
The case-insensitive output location for the exported CSV file.

LSAF_EXPORTPATH=*export-path*
The case-sensitive output path for the exported CSV file.

Optional Parameters

LSAF_OVERWRITE=0 | 1
Indicates whether an existing unversioned repository or existing workspace file is overwritten by a new file with a same path. See "LSAF_OVERWRITE=0 (Default) | 1" on page 6 for more information about this argument.

LSAF_ENABLEVERSIONING=0 | 1
Indicates whether versioning is enabled for a new file in the repository. See "LSAF_ENABLEVERSIONING=0 (Default) | 1" on page 6 for more information about this argument.

LSAF_VERSIONTYPE=MAJOR (Default) | MINOR | CUSTOM
Indicates the version type to apply to a new file or a new version of an existing versioned repository file. See "LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM" on page 6 for more information about this argument.

LSAF_CUSTOMVERSION=*custom-version*
The version number to use to create a customized version of a new or existing versioned file. See "LSAF_CUSTOMVERSION=version" on page 6 for more information about this argument.

LSAF_COMMENT=*comment*
The check-in comment to associate with the action of adding a file or file version to the repository. When LSAF_EXPORTLOCATION=WORKSPACE, this value is ignored.

Macro Summary Table

Category	Language Elements	Description
QueryService Module	%LSAF_GETQUERYCOLUMNS Macro (p. 320)	Gets the metadata for the columns that are applicable for the specified query type. The metadata is stored in a SAS data set.
	%LSAF_GETQUERYTEMPLATEDATASET Macro (p. 321)	Gets the structure that is required as input to the macros that query the system. The metadata is stored in a SAS data set with zero observations.
	%LSAF_QUERYAUDITDETAILS Macro (p. 322)	Extracts detailed data from entries that are in the audit history records. The detailed data to extract is specified in a query that is contained in a data set.
	%LSAF_QUERYAUDITENTRIES Macro (p. 324)	Extracts audit entries from the audit history records. The entries to extract are specified in a query that is contained in a data set.
	%LSAF_QUERYCONTAINER Macro (p. 326)	Extracts repository container information to a file based on the specified query. The query is contained in a SAS data set.
	%LSAF_QUERYCONTEXTMEMBERSHIP Macro (p. 327)	Extracts the assigned membership of a context. You can also include the defined user groups. The entries to extract are specified in a query that is contained in a SAS data set.
	%LSAF_QUERYCURRENTPERMISSIONS Macro (p. 329)	Extracts the current effective permissions for items in the repository.
	%LSAF_QUERYDEFAULTPERMISSIONS Macro (p. 331)	Extracts the default effective permissions for containers in the repository.
	%LSAF_QUERYDISTCONTEXTMEMBERSHIP Macro (p. 332)	Extracts the distinct members of a context. The entries to extract are specified in a query that is contained in a SAS data set.
	%LSAF_QUERYFILE Macro (p. 334)	Extracts repository file information for the latest version of a file based on the specified query. The query is contained in a SAS data set.

Category	Language Elements	Description
	%LSAF_QUERYFILEVERSION Macro (p. 336)	Extracts repository file information for each version of a file based on the specified query. The query is contained in a SAS data set.
	%LSAF_QUERYITEM Macro (p. 337)	Extracts information about a file or container that is in the repository to a file based on the specified query. The query is contained in a SAS data set.
	%LSAF_QUERYRECYCLEBINCONTAINER Macro (p. 339)	Extracts individual containers that have been deleted and that reside in a recycle bin, based on the specified query. The query is contained in a SAS data set.
	%LSAF_QUERYRECYCLEBINFILE Macro (p. 341)	Extracts individual files that have been deleted and that reside in a recycle bin, based on the specified query. The query is contained in a SAS data set.
	%LSAF_QUERYRECYCLEBINFILEVERSION Macro (p. 342)	Extracts individual files that have been deleted and that reside in a recycle bin, based on the specified query. The query is contained in a SAS data set.
	%LSAF_QUERYRECYCLEBINITEM Macro (p. 344)	Extracts individual items that have been deleted and that reside in a recycle bin, based on the specified query. The query is contained in a SAS data set.

Dictionary

%LSAF_GETQUERYCOLUMNS Macro

Gets the metadata for the columns that are applicable for the specified query type. The metadata is stored in a SAS data set.

Category: QueryService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETQUERYCOLUMNS(LSAF_QUERYTYPE=*query-type*<, SAS_DSNAME=*name*>);

Required Argument

LSAF_QUERYTYPE=*query-type*
The case-insensitive query type of the column. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=QueryType`.

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for all of the query columns. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETQUERYCOLUMNS.

The data set contains a row for each query column and columns with the following names:

Column Name	Description
queryType	The query type of the column.
columnQueryClass	The query class of the column.
columnName	The name of the column.
columnType	The type of the column.

%LSAF_GETQUERYTEMPLATEDATASET Macro

Gets the structure that is required as input to the macros that query the system. The metadata is stored in a SAS data set with zero observations.

Category: QueryService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETQUERYTEMPLATEDATASET(<SAS_DSNAME=*name*>);

Optional Argument

SAS_DSNAME=*name*
The name of the output data set. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETQUERYTEMPLATEDATASET.

The data set contains no rows, only the following columns that are required to submit a query:

Column Name	Description
recordType	The type of query element that the record represents.
columnClass	The class of the column to use with the query operation that is associated with the record type.
columnName	The name of the column to use with the query operation that is associated with the record type.
value	The value to use with the query operation that is associated with the record type.
comparator	The operator that the query uses to handle the constraint. This value is required when recordType is CONSTRAINT.
isCaseSensitive	Indicates whether a column value is treated as case-sensitive.

For more information, see “LSAF_QUERYDATASET=*query-data-set-name*” on page 316.

%LSAF_QUERYAUDITDETAILS Macro

Extracts detailed data from entries that are in the audit history records. The detailed data to extract is specified in a query that is contained in a data set.

Category: QueryService Module

Note: This macro sets the standard macro variables and _lsafAuditDetailsExportPath_, which is the full path to the query results CSV file.

Syntax

%LSAF_QUERYAUDITDETAILS(LSAF_QUERYDATASET=*name*, LSAF_VALIDATEDDATASET=*name*, LSAF_EXPORTLOCATION=WORKSPACE | REPOSITORY, LSAF_EXPORTPATH=*path*<, LSAF_OVERWRITE=0 | 1><, LSAF_ENABLEVERSIONING=0 | 1><, LSAF_VERSIONTYPE=MINOR | MAJOR |

CUSTOM><, LSAF_CUSTOMVERSION=*version*><,
LSAF_COMMENT=*comment*>);

Required Arguments

LSAF_QUERYDATASET=*name*
The name of the input data set that contains the query metadata. See "Data Set Macros" on page 5 for information about specifying the name of the data set. When querying against the columnName MODE with the columnClass of AuditEntry, the valid values are USER, SYSTEM, and ADMIN.

For more information, see "LSAF_QUERYDATASET=*query-data-set-name*" on page 316.

LSAF_VALIDATEDDATASET=*name*
The name of the output data set that contains the results of the validation of the records from the input data set LSAF_QUERYDATASET. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFQUERYAUDITENTRIES.

For more information, see "LSAF_VALIDATEDDATASET=*validated-data-set-name*" on page 318.

LSAF_EXPORTLOCATION=WORKSPACE | REPOSITORY
The case-insensitive output location for the exported CSV file.

LSAF_EXPORTPATH=*path*
The case-sensitive output path for the exported CSV file.

Optional Arguments

LSAF_OVERWRITE=0 (Default) | 1
Indicates whether an existing unversioned repository or existing workspace file is overwritten by a new file with a same path. See "LSAF_OVERWRITE=0 (Default) | 1" on page 6 for more information about this argument.

LSAF_ENABLEVERSIONING=0 (Default) | 1
Indicates whether versioning is enabled for a new file in the repository. See "LSAF_ENABLEVERSIONING=0 (Default) | 1" on page 6 for more information about this argument.

LSAF_VERSIONTYPE=MINOR | MAJOR (Default) | CUSTOM
Indicates the version type to apply to a new file or a new version of an existing versioned repository file. See "LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM" on page 6 for more information about this argument.

LSAF_CUSTOMVERSION=*version*
The version number to use to create a customized version of a new or existing versioned file. See "LSAF_CUSTOMVERSION=version" on page 6 for more information about this argument.

LSAF_COMMENT=*comment*
The check-in comment to associate with the action of adding a file or file version to the repository. When LSAF_EXPORTLOCATION=WORKSPACE, this value is ignored.

Details

Extracts detailed data from entries that are in the audit history records. The detailed data to extract is specified in a query that is contained in a data set. The extracted data is stored in a comma-separated values file that is located in your workspace or in the repository. Each detail is in a separate row that includes the parent audit entry information.

To get information that concerns the necessary columnClass and columnName data, and their data types, call %LSAF_GETQUERYCOLUMNS with the type AuditEntryDetailQuery.

Sets the value of the macro variable _lsafAuditDetailsExportPath_ to the full path to the query results CSV file, which includes the file extension See "Value Macros" on page 5 for information about value macros.

You must have the View Audit History privilege to execute the query.

See "Query Audit Records" in *SAS Life Science Analytics Framework: SAS Macro API User's Guide* for more information about querying audit records, including examples.

%LSAF_QUERYAUDITENTRIES Macro

Extracts audit entries from the audit history records. The entries to extract are specified in a query that is contained in a data set.

Category: QueryService Module

Note: This macro sets the standard macro variables and _lsafAuditDetailsExportPath_, which is the full path to the query results CSV file.

Syntax

%LSAF_QUERYAUDITENTRIES(LSAF_QUERYDATASET=*name*, LSAF_VALIDATEDDATASET=*name*, LSAF_EXPORTLOCATION=WORKSPACE | REPOSITORY, LSAF_EXPORTPATH=*path*<, LSAF_OVERWRITE=0 | 1><, LSAF_ENABLEVERSIONING=0 | 1><, LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM><, LSAF_CUSTOMVERSION=*version*><, LSAF_COMMENT=*comment*>);

Required Arguments

LSAF_QUERYDATASET=*name*

The name of the input data set that contains the query metadata. See "Data Set Macros" on page 5 for information about specifying the name of the data set. When querying against the columnName MODE with the columnClass of AuditEntry, the valid values are USER, SYSTEM, and ADMIN.

For more information, see "LSAF_QUERYDATASET=*query-data-set-name*" on page 316.

LSAF_VALIDATEDDATASET=*name*
The name of the output data set that contains the results of the validation of the records from the input data set LSAF_QUERYDATASET. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFQUERYAUDITENTRIES.

For more information, see "LSAF_VALIDATEDDATASET=*validated-data-set-name*" on page 318.

LSAF_EXPORTLOCATION=WORKSPACE | REPOSITORY
The case-insensitive output location for the exported CSV file.

LSAF_EXPORTPATH=*path*
The case-sensitive output path for the exported CSV file.

Optional Arguments

LSAF_OVERWRITE=0 (Default) | 1
Indicates whether an existing unversioned repository or existing workspace file is overwritten by a new file with a same path. See "LSAF_OVERWRITE=0 (Default) | 1" on page 6 for more information about this argument.

LSAF_ENABLEVERSIONING=0 (Default) | 1
Indicates whether versioning is enabled for a new file in the repository. See "LSAF_ENABLEVERSIONING=0 (Default) | 1" on page 6 for more information about this argument.

LSAF_VERSIONTYPE=MAJOR (Default) | MINOR | CUSTOM
Indicates the version type to apply to a new file or a new version of an existing versioned repository file. See "LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM" on page 6 for more information about this argument.

LSAF_CUSTOMVERSION=*version*
The version number to use to create a customized version of a new or existing versioned file. See "LSAF_CUSTOMVERSION=version" on page 6 for more information about this argument.

LSAF_COMMENT=*comment*
The check-in comment to associate with the action of adding a file or file version to the repository. When LSAF_EXPORTLOCATION=WORKSPACE, this value is ignored.

Details

Extracts audit entries from the audit history records. The entries to extract are specified in a query that is contained in a data set. The extracted data is stored in a comma-separated values file that is located in your workspace or in the repository.

To get information that concerns the necessary columnClass and columnName data, and their data types, call %LSAF_GETQUERYCOLUMNS with the type AuditEntryQuery.

Sets the value of the macro variable _lsafAuditDetailsExportPath_ to the full path to the query results CSV file, which includes the file extension. See "Value Macros" on page 5 for information about value macros.

You must have the View Audit History privilege to execute the query.

See "Query Audit Records" in *SAS Life Science Analytics Framework: SAS Macro API User's Guide* for more information about querying audit records, including examples.

%LSAF_QUERYCONTAINER Macro

Extracts repository container information to a file based on the specified query. The query is contained in a SAS data set.

Category: QueryService Module

Note: This macro sets the standard macro variables and _lsafContainerExportPath_, which is the full path to the query results CSV file.

Syntax

%LSAF_QUERYCONTAINER(LSAF_QUERYDATASET=*name*, LSAF_VALIDATEDDATASET=*name*, LSAF_EXPORTLOCATION=WORKSPACE | REPOSITORY, LSAF_EXPORTPATH=*path*<, LSAF_OVERWRITE=0 | 1><, LSAF_ENABLEVERSIONING=0 | 1><, LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM><, LSAF_CUSTOMVERSION=*version*><, LSAF_COMMENT=*comment*>);

Required Arguments

LSAF_QUERYDATASET=*name*

The name of the input data set that contains the query metadata. See "Data Set Macros" on page 5 for information about specifying the name of the data set. When querying against the columnName STATE with the columnQueryClass of RepositoryItem, the valid values are ACTIVE and CLOSED.

For more information, see "LSAF_QUERYDATASET=*query-data-set-name*" on page 316.

LSAF_VALIDATEDDATASET=*name*

The name of the output data set that contains the results of the validation of the records from the input data set LSAF_QUERYDATASET. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFQUERYCONTAINER.

For more information, see "LSAF_VALIDATEDDATASET=*validated-data-set-name*" on page 318.

LSAF_EXPORTLOCATION=WORKSPACE | REPOSITORY

The case-insensitive output location for the exported CSV file.

LSAF_EXPORTPATH=*path*

The case-sensitive output path for the exported CSV file.

Optional Arguments

LSAF_OVERWRITE=0 (Default) | 1
Indicates whether an existing unversioned repository or existing workspace file is overwritten by a new file with a same path. See “LSAF_OVERWRITE=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_ENABLEVERSIONING=0 (Default) | 1
Indicates whether versioning is enabled for a new file in the repository. See “LSAF_ENABLEVERSIONING=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM
Indicates the version type to apply to a new file or a new version of an existing versioned repository file. See “LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM” on page 6 for more information about this argument.

LSAF_CUSTOMVERSION=*version*
The version number to use to create a customized version of a new or existing versioned file. See “LSAF_CUSTOMVERSION=version” on page 6 for more information about this argument.

LSAF_COMMENT=*comment*
The check-in comment to associate with the action of adding a file or file version to the repository. When LSAF_EXPORTLOCATION=WORKSPACE, this value is ignored.

Details

Extracts repository container information to a file based on the specified query. The query is contained in a SAS data set. The extracted data is stored in a comma-separated values file that is located in your workspace or in the repository.

To get information that is needed to specify the required columnClass and columnName data, and their data types, call %LSAF_GETQUERYCOLUMNS with the type CONTAINER. The size column represents the total size of all of the files that are in the container and below the container, which, if it is versioned, includes all of the versions of a file.

Sets the value of the macro variable _lsafContainerExportPath_ to the full path to the query results CSV file, which includes the file extension. See “Value Macros” on page 5 for information about value macros.

You must have the Administrative mode privilege to execute the query.

%LSAF_QUERYCONTEXTMEMBERSHIP Macro

Extracts the assigned membership of a context. You can also include the defined user groups. The entries to extract are specified in a query that is contained in a SAS data set.

Category: QueryService Module

Note: This macro sets the standard macro variables and _lsafContxtMembershpExpPath_, which is the full path to the query results CSV file.

Syntax

%LSAF_QUERYCONTEXTMEMBERSHIP(LSAF_PATH=*path*, LSAF_VALIDATEDDATASET=*name*, LSAF_EXPORTLOCATION=WORKSPACE | REPOSITORY, LSAF_EXPORTPATH=*path*<, LSAF_SCOPE=*membership-scope*><, LSAF_QUERYDATASET=*name*><, LSAF_OVERWRITE=0 | 1><, LSAF_ENABLEVERSIONING=0 | 1><, LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM><, LSAF_CUSTOMVERSION=*version*><, LSAF_COMMENT=*comment*>);

Required Arguments

LSAF_PATH=*path*
The context path.

LSAF_VALIDATEDDATASET=*name*
The name of the output data set that contains the results of the validation of the records from the input data set LSAF_QUERYDATASET. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFQUERYCONTEXTMEMBERSHIP.

For more information, see “LSAF_VALIDATEDDATASET=*validated-data-set-name*” on page 318.

LSAF_EXPORTLOCATION=WORKSPACE | REPOSITORY
The case-insensitive output location for the exported CSV file.

LSAF_EXPORTPATH=*path*
The case-sensitive output path for the exported CSV file.

Optional Arguments

LSAF_SCOPE=*membership-scope*
The membership scope. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=MembershipScope`.

LSAF_QUERYDATASET=*name*
The name of the input data set that contains the query metadata. See “Data Set Macros” on page 5 for information about specifying the name of the data set. If not specified, the query returns the membership information for the context.

For more information, see “LSAF_QUERYDATASET=*query-data-set-name*” on page 316.

LSAF_OVERWRITE=0 (Default) | 1
Indicates whether an existing unversioned repository or existing workspace file is overwritten by a new file with a same path. See “LSAF_OVERWRITE=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_ENABLEVERSIONING=0 (Default) | 1
Indicates whether versioning is enabled for a new file in the repository. See “LSAF_ENABLEVERSIONING=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_VERSIONTYPE=MAJOR (Default) | MINOR | CUSTOM
Indicates the version type to apply to a new file or a new version of an existing versioned repository file. See “LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM” on page 6 for more information about this argument.

LSAF_CUSTOMVERSION=*version*
The version number to use to create a customized version of a new or existing versioned file. See “LSAF_CUSTOMVERSION=version” on page 6 for more information about this argument.

LSAF_COMMENT=*comment*
The check-in comment to associate with the action of adding a file or file version to the repository. When LSAF_EXPORTLOCATION=WORKSPACE, this value is ignored.

Details

Extracts the assigned membership of a context. You can also include the defined user groups. The entries to extract are specified in a query that is contained in a SAS data set. The extracted data is stored in a comma-separated values file that is located in your workspace or in the repository

To get information that is needed to specify the required columnClass and columnName data, and their data types, call %LSAF_GETQUERYCOLUMNS with the type ContextMembership.

Sets the value of the macro variable _lsafContxtMembershpExpPath_ to the full path to the query results CSV file, which includes the file extension. See “Value Macros” on page 5 for information about value macros.

You must be an administrative mode user to execute the query.

%LSAF_QUERYCURRENTPERMISSIONS Macro

Extracts the current effective permissions for items in the repository.

Category: QueryService Module

Note: This macro sets the standard macro variables and _lsafcurrpermissionsexportpath_, which is the full path to the query results CSV file.

Syntax

%LSAF_QUERYCURRENTPERMISSIONS(LSAF_QUERYDATASET=*name*, LSAF_VALIDATEDDATASET=*name*, LSAF_EXPORTLOCATION=WORKSPACE | REPOSITORY, LSAF_EXPORTPATH=*path*<, LSAF_OVERWRITE=0 | 1><, LSAF_ENABLEVERSIONING=0 | 1><, LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM><, LSAF_CUSTOMVERSION=*version*><, LSAF_COMMENT=*comment*>);

Required Arguments

LSAF_QUERYDATASET=*name*
The name of the input data set that contains the query metadata. See “Data Set Macros” on page 5 for information about specifying the name of the data set.

For more information, see “LSAF_QUERYDATASET=*query-data-set-name*” on page 316.

LSAF_VALIDATEDDATASET=*name*
The name of the output data set that contains the results of the validation of the records from the input data set LSAF_QUERYDATASET. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFQUERYCURRENTPERMISSIONS.

For more information, see “LSAF_VALIDATEDDATASET=*validated-data-set-name*” on page 318.

LSAF_EXPORTLOCATION=WORKSPACE | REPOSITORY
The case-insensitive output location for the exported CSV file.

LSAF_EXPORTPATH=*path*
The case-sensitive output path for the exported CSV file.

Optional Arguments

LSAF_OVERWRITE=0 (Default) | 1
Indicates whether an existing unversioned repository or existing workspace file is overwritten by a new file with a same path. See “LSAF_OVERWRITE=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_ENABLEVERSIONING=0 (Default) | 1
Indicates whether versioning is enabled for a new file in the repository. See “LSAF_ENABLEVERSIONING=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_VERSIONTYPE=MAJOR (Default) | MINOR | CUSTOM
Indicates the version type to apply to a new file or a new version of an existing versioned repository file. See “LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM” on page 6 for more information about this argument.

LSAF_CUSTOMVERSION=*version*
The version number to use to create a customized version of a new or existing versioned file. See “LSAF_CUSTOMVERSION=version” on page 6 for more information about this argument.

LSAF_COMMENT=*comment*
The check-in comment to associate with the action of adding a file or file version to the repository. When LSAF_EXPORTLOCATION=WORKSPACE, this value is ignored.

Details

Extracts the current effective permissions for items in the repository. The query is contained in a SAS data set. The extracted data is stored in a comma-separated values file that is located in your workspace or in the repository

To get information that is needed to specify the required columnClass and columnName data, and their data types, call %LSAF_GETQUERYCOLUMNS with the type PERMISSIONS.

Sets the value of the macro variable _lsafcurrpermissionsexportpath_ to the full path to the query results CSV file, which includes the file extension. See “Value Macros” on page 5 for information about value macros.

You must be an administrative mode user to execute the query.

%LSAF_QUERYDEFAULTPERMISSIONS Macro

Extracts the default effective permissions for containers in the repository.

Category: QueryService Module

Note: This macro sets the standard macro variables and _lsafdefpermissionsexportpath_, which is the full path to the query results CSV file.

Syntax

%LSAF_QUERYDEFAULTPERMISSIONS(LSAF_QUERYDATASET=*name*, LSAF_VALIDATEDDATASET=*name*, LSAF_EXPORTLOCATION=WORKSPACE | REPOSITORY, LSAF_EXPORTPATH=*path*<, LSAF_OVERWRITE=0 | 1><, LSAF_ENABLEVERSIONING=0 | 1><, LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM><, LSAF_CUSTOMVERSION=*version*><, LSAF_COMMENT=*comment*>);

Required Arguments

LSAF_QUERYDATASET=*name*
The name of the input data set that contains the query metadata. See “Data Set Macros” on page 5 for information about specifying the name of the data set.

For more information, see “LSAF_QUERYDATASET=*query-data-set-name*” on page 316.

LSAF_VALIDATEDDATASET=*name*
The name of the output data set that contains the results of the validation of the records from the input data set LSAF_QUERYDATASET. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFQUERYDEFAULTPERMISSIONS.

For more information, see “LSAF_VALIDATEDDATASET=*validated-data-set-name*” on page 318.

LSAF_EXPORTLOCATION=WORKSPACE | REPOSITORY
The case-insensitive output location for the exported CSV file.

LSAF_EXPORTPATH=*path*
The case-sensitive output path for the exported CSV file.

Optional Arguments

LSAF_OVERWRITE=0 (Default) | 1
Indicates whether an existing unversioned repository or existing workspace file is overwritten by a new file with a same path. See “LSAF_OVERWRITE=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_ENABLEVERSIONING=0 (Default) | 1
Indicates whether versioning is enabled for a new file in the repository. See “LSAF_ENABLEVERSIONING=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_VERSIONTYPE=MAJOR (Default) | MINOR | CUSTOM
Indicates the version type to apply to a new file or a new version of an existing versioned repository file. See “LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM” on page 6 for more information about this argument.

LSAF_CUSTOMVERSION=*version*
The version number to use to create a customized version of a new or existing versioned file. See “LSAF_CUSTOMVERSION=version” on page 6 for more information about this argument.

LSAF_COMMENT=*comment*
The check-in comment to associate with the action of adding a file or file version to the repository. When LSAF_EXPORTLOCATION=WORKSPACE, this value is ignored.

Details

Extracts the default effective permissions for containers in the repository. The query is contained in a SAS data set. The extracted data is stored in a comma-separated values file that is located in your workspace or in the repository

To get information that is needed to specify the required columnClass and columnName data, and their data types, call %LSAF_GETQUERYCOLUMNS with the type PERMISSIONS.

Sets the value of the macro variable _lsafdefpermissionsexportpath_ to the full path to the query results CSV file, which includes the file extension. See “Value Macros” on page 5 for information about value macros.

You must be an administrative mode user to execute the query.

%LSAF_QUERYDISTCONTEXTMEMBERSHIP Macro

Extracts the distinct members of a context. The entries to extract are specified in a query that is contained in a SAS data set.

Category: QueryService Module

Note: This macro sets the standard macro variables and _lsafDistContxtMembershpExpPath_, which is the full path to the query results CSV file.

Syntax

%LSAF_QUERYDISTCONTEXTMEMBERSHIP(LSAF_PATH=*path*, LSAF_EXPORTLOCATION=WORKSPACE | REPOSITORY,

LSAF_EXPORTPATH=*path*<, LSAF_SCOPE=*membership-scope*><, LSAF_QUERYDATASET=*name*><, LSAF_VALIDATEDDATASET=*name*><, LSAF_OVERWRITE=0 | 1><, LSAF_ENABLEVERSIONING=0 | 1><, LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM><, LSAF_CUSTOMVERSION=*version*><, LSAF_COMMENT=*comment*>);

Required Arguments

LSAF_PATH=*path*
The path of the context to query.

LSAF_EXPORTLOCATION=WORKSPACE | REPOSITORY
The case-insensitive output location for the exported CSV file.

LSAF_EXPORTPATH=*path*
The case-sensitive output path for the exported CSV file.

Optional Arguments

LSAF_SCOPE=*membership-scope*
The membership scope. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=MembershipScope`.

LSAF_QUERYDATASET=*name*
The name of the input data set that contains the query metadata. See "Data Set Macros" on page 5 for information about specifying the name of the data set. If not specified, the query returns the membership information for the context.

For more information, see "LSAF_QUERYDATASET=*query-data-set-name*" on page 316.

LSAF_VALIDATEDDATASET=*name*
The name of the output data set that contains the results of the validation of the records from the input data set LSAF_QUERYDATASET. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFQUERYDISTCONTEXTMEMBERSHIP.

For more information, see "LSAF_VALIDATEDDATASET=*validated-data-set-name*" on page 318.

LSAF_OVERWRITE=0 (Default) | 1
Indicates whether an existing unversioned repository or existing workspace file is overwritten by a new file with a same path. See "LSAF_OVERWRITE=0 (Default) | 1" on page 6 for more information about this argument.

LSAF_ENABLEVERSIONING=0 (Default) | 1
Indicates whether versioning is enabled for a new file in the repository. See "LSAF_ENABLEVERSIONING=0 (Default) | 1" on page 6 for more information about this argument.

LSAF_VERSIONTYPE=MAJOR (Default) | MINOR | CUSTOM
Indicates the version type to apply to a new file or a new version of an existing versioned repository file. See "LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM" on page 6 for more information about this argument.

LSAF_CUSTOMVERSION=*version*
The version number to use to create a customized version of a new or existing versioned file. See "LSAF_CUSTOMVERSION=version" on page 6 for more information about this argument.

LSAF_COMMENT=*comment*
The check-in comment to associate with the action of adding a file or file version to the repository. When LSAF_EXPORTLOCATION=WORKSPACE, this value is ignored.

Details

Extracts the distinct members of a context. The entries to extract are specified in a query that is contained in a SAS data set. The extracted data is stored in a comma-separated values file that is located in your workspace or in the repository.

To get information that is needed to specify the required columnClass and columnName data, and their data types, call %LSAF_GETQUERYCOLUMNS with the type DistinctContextMembership.

Sets the value of the macro variable _lsafDistContxtMembershpExpPath_ to the full path to the query results CSV file, which includes the file extension. See “Value Macros” on page 5 for information about value macros.

You must be an administrative mode user to execute the query.

%LSAF_QUERYFILE Macro

Extracts repository file information for the latest version of a file based on the specified query. The query is contained in a SAS data set.

Category: QueryService Module

Note: This macro sets the standard macro variables and _lsafFileExportPath_, which is the full path to the query results CSV file.

Syntax

%LSAF_QUERYFILE(LSAF_QUERYDATASET=*name*, LSAF_VALIDATEDDATASET=*name*, LSAF_EXPORTLOCATION=WORKSPACE | REPOSITORY, LSAF_EXPORTPATH=*path*<, LSAF_OVERWRITE=0 | 1><, LSAF_ENABLEVERSIONING=0 | 1><, LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM><, LSAF_CUSTOMVERSION=*version*><, LSAF_COMMENT=*comment*>);

Required Arguments

LSAF_QUERYDATASET=*name*
The name of the input data set that contains the query metadata. See “Data Set Macros” on page 5 for information about specifying the name of the data set. When querying against the columnName STATE with the columnQueryClass of RepositoryItem, the valid values are ACTIVE and CLOSED. For columnName signingStatus and columnQueryClass of RepositoryFile, the valid values are NONE, PREVIOUS, CURRENT, and CURRENT_AND_PREVIOUS.

For more information, see “LSAF_QUERYDATASET=*query-data-set-name*” on page 316.

LSAF_VALIDATEDDATASET=*name*
The name of the output data set that contains the results of the validation of the records from the input data set LSAF_QUERYDATASET. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFQUERYFILE.

For more information, see “LSAF_VALIDATEDDATASET=*validated-data-set-name*” on page 318.

LSAF_EXPORTLOCATION=WORKSPACE | REPOSITORY
The case-insensitive output location for the exported CSV file.

LSAF_EXPORTPATH=*path*
The case-sensitive output path for the exported CSV file.

Optional Arguments

LSAF_OVERWRITE=0 (Default) | 1
Indicates whether an existing unversioned repository or existing workspace file is overwritten by a new file with a same path. See “LSAF_OVERWRITE=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_ENABLEVERSIONING=0 (Default) | 1
Indicates whether versioning is enabled for a new file in the repository. See “LSAF_ENABLEVERSIONING=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_VERSIONTYPE=MAJOR (Default) | MINOR | CUSTOM
Indicates the version type to apply to a new file or a new version of an existing versioned repository file. See “LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM” on page 6 for more information about this argument.

LSAF_CUSTOMVERSION=*version*
The version number to use to create a customized version of a new or existing versioned file. See “LSAF_CUSTOMVERSION=version” on page 6 for more information about this argument.

LSAF_COMMENT=*comment*
The check-in comment to associate with the action of adding a file or file version to the repository. When LSAF_EXPORTLOCATION=WORKSPACE, this value is ignored.

Details

Extracts repository file information for the latest version of a file based on the specified query. The query is contained in a SAS data set. The extracted data is stored in a comma-separated values file that is located in your workspace or in the repository.

To get information that is needed to specify the required columnClass and columnName data, and their data types, call %LSAF_GETQUERYCOLUMNS with the type FILE.

Sets the value of the macro variable _lsafFileExportPath_ to the full path to the query results CSV file, which includes the file extension. See “Value Macros” on page 5 for information about value macros.

You must have the Administrative mode privilege to execute the query.

%LSAF_QUERYFILEVERSION Macro

Extracts repository file information for each version of a file based on the specified query. The query is contained in a SAS data set.

Category: QueryService Module

Note: This macro sets the standard macro variables and _lsafFileVersionExportPath_, which is the full path to the query results CSV file.

Syntax

%LSAF_QUERYFILEVERSION(LSAF_QUERYDATASET=*name*, LSAF_VALIDATEDDATASET=*name*, LSAF_EXPORTLOCATION=WORKSPACE | REPOSITORY, LSAF_EXPORTPATH=*path*<, LSAF_OVERWRITE=0 | 1><, LSAF_ENABLEVERSIONING=0 | 1><, LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM><, LSAF_CUSTOMVERSION=*version*><, LSAF_COMMENT=*comment*>);

Required Arguments

LSAF_QUERYDATASET=*name*

The name of the input data set that contains the query metadata. See “Data Set Macros” on page 5 for information about specifying the name of the data set. When querying against the columnName STATE with the columnQueryClass of RepositoryItem, the valid values are ACTIVE and CLOSED. For columnName signingStatus and columnQueryClass of RepositoryFile, the valid values are NONE, PREVIOUS, CURRENT, and CURRENT_AND_PREVIOUS.

For more information, see “LSAF_QUERYDATASET=*query-data-set-name*” on page 316.

LSAF_VALIDATEDDATASET=*name*

The name of the output data set that contains the results of the validation of the records from the input data set LSAF_QUERYDATASET. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFQUERYFILEVERSION.

For more information, see “LSAF_VALIDATEDDATASET=*validated-data-set-name*” on page 318.

LSAF_EXPORTLOCATION=WORKSPACE | REPOSITORY

The case-insensitive output location for the exported CSV file.

LSAF_EXPORTPATH=*path*

The case-sensitive output path for the exported CSV file.

Optional Arguments

LSAF_OVERWRITE=0 (Default) | 1
Indicates whether an existing unversioned repository or existing workspace file is overwritten by a new file with a same path. See “LSAF_OVERWRITE=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_ENABLEVERSIONING=0 (Default) | 1
Indicates whether versioning is enabled for a new file in the repository. See “LSAF_ENABLEVERSIONING=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_VERSIONTYPE=MAJOR (Default) | MINOR | CUSTOM
Indicates the version type to apply to a new file or a new version of an existing versioned repository file. See “LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM” on page 6 for more information about this argument.

LSAF_CUSTOMVERSION=*version*
The version number to use to create a customized version of a new or existing versioned file. See “LSAF_CUSTOMVERSION=version” on page 6 for more information about this argument.

LSAF_COMMENT=*comment*
The check-in comment to associate with the action of adding a file or file version to the repository. When LSAF_EXPORTLOCATION=WORKSPACE, this value is ignored.

Details

Extracts repository file information for each version of a file based on the specified query. The query is contained in a SAS data set. The extracted data is stored in a comma-separated values file that is located in your workspace or in the repository. Each version of the file is listed in a separate row.

To get information that is needed to specify the required columnClass and columnName data, and their data types, can be retrieved by call %LSAF_GETQUERYCOLUMNS with the type FILEVERSION.

Sets the value of the macro variable _lsafFileVersionExportPath_ to the full path to the query results CSV file, which includes the file extension. See “Value Macros” on page 5 for information about value macros.

You must have the Administrative mode privilege to execute the query.

%LSAF_QUERYITEM Macro

Extracts information about a file or container that is in the repository to a file based on the specified query. The query is contained in a SAS data set.

Category: QueryService Module

Note: This macro sets the standard macro variables and _lsafItemExportPath_, which is the full path to the query results CSV file.

Syntax

%LSAF_QUERYITEM(LSAF_QUERYDATASET=*name*, LSAF_VALIDATEDDATASET=*name*, LSAF_EXPORTLOCATION=WORKSPACE | REPOSITORY, LSAF_EXPORTPATH=*path*<, LSAF_OVERWRITE=0 | 1><, LSAF_ENABLEVERSIONING=0 | 1><, LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM><, LSAF_CUSTOMVERSION=*version*><, LSAF_COMMENT=*comment*>);

Required Arguments

LSAF_QUERYDATASET=*name*
The name of the input data set that contains the query metadata. See “Data Set Macros” on page 5 for information about specifying the name of the data set. When querying against the columnName STATE with the columnQueryClass of RepositoryItem, the valid values are ACTIVE and CLOSED.

For more information, see “LSAF_QUERYDATASET=*query-data-set-name*” on page 316.

LSAF_VALIDATEDDATASET=*name*
The name of the output data set that contains the results of the validation of the records from the input data set LSAF_QUERYDATASET. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFQUERYITEM.

For more information, see “LSAF_VALIDATEDDATASET=*validated-data-set-name*” on page 318.

LSAF_EXPORTLOCATION=WORKSPACE | REPOSITORY
The case-insensitive output location for the exported CSV file.

LSAF_EXPORTPATH=*path*
The case-sensitive output path for the exported CSV file.

Optional Arguments

LSAF_OVERWRITE=0 (Default) | 1
Indicates whether an existing unversioned repository or existing workspace file is overwritten by a new file with a same path. See “LSAF_OVERWRITE=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_ENABLEVERSIONING=0 (Default) | 1
Indicates whether versioning is enabled for a new file in the repository. See “LSAF_ENABLEVERSIONING=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_VERSIONTYPE=MAJOR (Default) | MINOR | CUSTOM
Indicates the version type to apply to a new file or a new version of an existing versioned repository file. See “LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM” on page 6 for more information about this argument.

LSAF_CUSTOMVERSION=*version*
The version number to use to create a customized version of a new or existing versioned file. See “LSAF_CUSTOMVERSION=version” on page 6 for more information about this argument.

LSAF_COMMENT=*comment*
The check-in comment to associate with the action of adding a file or file version to the repository. When LSAF_EXPORTLOCATION=WORKSPACE, this value is ignored.

Details

Extracts information about a file or container that is in the repository to a file based on the specified query. The query is contained in a SAS data set. The extracted data is stored in a file that is located in your workspace or in the repository. Each item is in a separate row. The results do not include content that is in the /Users folder.

To get information that is needed to specify the required columnClass and columnName data, and their data types, can be retrieved by call %LSAF_GETQUERYCOLUMNS with the type ITEM.

Sets the value of the macro variable _lsafItemExportPath_ to the full path to the query results CSV file, which includes the file extension. See “Value Macros” on page 5 for information about value macros.

You must have the Administrative mode privilege to execute the query.

%LSAF_QUERYRECYCLEBINCONTAINER Macro

Extracts individual containers that have been deleted and that reside in a recycle bin, based on the specified query. The query is contained in a SAS data set.

Category: QueryService Module

Note: This macro sets the standard macro variables and _lsafRBContainerExportPath_, which is the full path to the query results CSV file.

Syntax

%LSAF_QUERYRECYCLEBINCONTAINER(LSAF_QUERYDATASET=*name*, LSAF_VALIDATEDDATASET=*name*, LSAF_EXPORTLOCATION=WORKSPACE | REPOSITORY, LSAF_EXPORTPATH=*path*<, LSAF_OVERWRITE=0 | 1><, LSAF_ENABLEVERSIONING=0 | 1><, LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM><, LSAF_CUSTOMVERSION=*version*><, LSAF_COMMENT=*comment*>);

Required Arguments

LSAF_QUERYDATASET=*name*
The name of the input data set that contains the query metadata. See “Data Set Macros” on page 5 for information about specifying the name of the data set.

For more information, see “LSAF_QUERYDATASET=*query-data-set-name*” on page 316.

LSAF_VALIDATEDDATASET=*name*
The name of the output data set that contains the results of the validation of the records from the input data set LSAF_QUERYDATASET. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFQUERYRECYCLEBINCONTAINER.

For more information, see "LSAF_VALIDATEDDATASET=*validated-data-set-name*" on page 318.

LSAF_EXPORTLOCATION=WORKSPACE | REPOSITORY
The case-insensitive output location for the exported CSV file.

LSAF_EXPORTPATH=*path*
The case-sensitive output path for the exported CSV file.

Optional Arguments

LSAF_OVERWRITE=0 (Default) | 1
Indicates whether an existing unversioned repository or existing workspace file is overwritten by a new file with a same path. See "LSAF_OVERWRITE=0 (Default) | 1" on page 6 for more information about this argument.

LSAF_ENABLEVERSIONING=0 (Default) | 1
Indicates whether versioning is enabled for a new file in the repository. See "LSAF_ENABLEVERSIONING=0 (Default) | 1" on page 6 for more information about this argument.

LSAF_VERSIONTYPE=MAJOR (Default) | MINOR | CUSTOM
Indicates the version type to apply to a new file or a new version of an existing versioned repository file. See "LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM" on page 6 for more information about this argument.

LSAF_CUSTOMVERSION=*version*
The version number to use to create a customized version of a new or existing versioned file. See "LSAF_CUSTOMVERSION=version" on page 6 for more information about this argument.

LSAF_COMMENT=*comment*
The check-in comment to associate with the action of adding a file or file version to the repository. When LSAF_EXPORTLOCATION=WORKSPACE, this value is ignored.

Details

Extracts individual containers that have been deleted and that reside in a recycle bin, based on the specified query. The query is contained in a SAS data set. The extracted data is stored in a file that is located in your workspace or in the repository. Each item is in a separate row.

Information about the necessary columnClass and columnName data, and their data types, can be retrieved calling %LSAF_GETQUERYCOLUMNS with the type RBCONTAINER.

Sets the value of the macro variable _lsafRBContainerExportPath_ to the full path to the query results CSV file, which includes the file extension. See "Value Macros" on page 5 for information about value macros.

You must have the Administrative mode privilege to execute the query.

See "Query the Recycle Bin" in *SAS Life Science Analytics Framework: SAS Macro API User's Guide* for more information about querying the recycle bin, including examples.

%LSAF_QUERYRECYCLEBINFILE Macro

Extracts individual files that have been deleted and that reside in a recycle bin, based on the specified query. The query is contained in a SAS data set.

Category: QueryService Module

Note: This macro sets the standard macro variables and _lsafRBFileExportPath_, which is the full path to the query results CSV file.

Syntax

%LSAF_QUERYRECYCLEBINFILE(LSAF_QUERYDATASET=*name*, LSAF_VALIDATEDDATASET=*name*, LSAF_EXPORTLOCATION=WORKSPACE | REPOSITORY, LSAF_EXPORTPATH=*path*<, LSAF_OVERWRITE=0 | 1><, LSAF_ENABLEVERSIONING=0 | 1><, LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM><, LSAF_CUSTOMVERSION=*version*><, LSAF_COMMENT=*comment*>);

Required Arguments

LSAF_QUERYDATASET=*name*
The name of the input data set that contains the query metadata. See "Data Set Macros" on page 5 for information about specifying the name of the data set.

For more information, see "LSAF_QUERYDATASET=*query-data-set-name*" on page 316.

LSAF_VALIDATEDDATASET=*name*
The name of the output data set that contains the results of the validation of the records from the input data set LSAF_QUERYDATASET. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFQUERYRECYCLEBINFILE.

For more information, see "LSAF_VALIDATEDDATASET=*validated-data-set-name*" on page 318.

LSAF_EXPORTLOCATION=WORKSPACE | REPOSITORY
The case-insensitive output location for the exported CSV file.

LSAF_EXPORTPATH=*path*
The case-sensitive output path for the exported CSV file.

Optional Arguments

LSAF_OVERWRITE=0 (Default) | 1
Indicates whether an existing unversioned repository or existing workspace file is overwritten by a new file with a same path. See “LSAF_OVERWRITE=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_ENABLEVERSIONING=0 (Default) | 1
Indicates whether versioning is enabled for a new file in the repository. See “LSAF_ENABLEVERSIONING=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM
Indicates the version type to apply to a new file or a new version of an existing versioned repository file. See “LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM” on page 6 for more information about this argument.

LSAF_CUSTOMVERSION=*version*
The version number to use to create a customized version of a new or existing versioned file. See “LSAF_CUSTOMVERSION=version” on page 6 for more information about this argument.

LSAF_COMMENT=*comment*
The check-in comment to associate with the action of adding a file or file version to the repository. When LSAF_EXPORTLOCATION=WORKSPACE, this value is ignored.

Details

Extracts individual files that have been deleted and that reside in a recycle bin, based on the specified query. The query is contained in a SAS data set. The extracted data is stored in a file that is located in your workspace or in the repository. Each item is in a separate row.

To get information about the necessary columnClass and columnName data, and their data types, can be retrieved by call %LSAF_GETQUERYCOLUMNS with the type RBFILE.

Sets the value of the macro variable _lsafRBFileExportPath_ to the name of the full path to the query results CSV file, which includes the file extension. See “Value Macros” on page 5 for information about value macros.

You must have the Administrative mode privilege to execute the query.

See “Query the Recycle Bin” in *SAS Life Science Analytics Framework: SAS Macro API User’s Guide* for more information about querying the recycle bin, including examples.

%LSAF_QUERYRECYCLEBINFILEVERSION Macro

Extracts individual files that have been deleted and that reside in a recycle bin, based on the specified query. The query is contained in a SAS data set.

Category: QueryService Module

Note: This macro sets the standard macro variables and _lsafRBFileVersionExportPath_, which is the full path to the query results CSV file.

Syntax

%LSAF_QUERYRECYCLEBINFILEVERSION(LSAF_QUERYDATASET=*name*, LSAF_VALIDATEDDATASET=*name*, LSAF_EXPORTLOCATION=WORKSPACE | REPOSITORY, LSAF_EXPORTPATH=*path*<, LSAF_OVERWRITE=0 | 1><, LSAF_ENABLEVERSIONING=0 | 1><, LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM><, LSAF_CUSTOMVERSION=*version*><, LSAF_COMMENT=*comment*>);

Required Arguments

LSAF_QUERYDATASET=*name*
The name of the input data set that contains the query metadata. See "Data Set Macros" on page 5 for information about specifying the name of the data set.

For more information, see "LSAF_QUERYDATASET=*query-data-set-name*" on page 316.

LSAF_VALIDATEDDATASET=*name*
The name of the output data set that contains the results of the validation of the records from the input data set LSAF_QUERYDATASET. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFQUERYRECYCLEBINFILEVERSION.

For more information, see "LSAF_VALIDATEDDATASET=*validated-data-set-name*" on page 318.

LSAF_EXPORTLOCATION=WORKSPACE | REPOSITORY
The case-insensitive output location for the exported CSV file.

LSAF_EXPORTPATH=*path*
The case-sensitive output path for the exported CSV file.

Optional Arguments

LSAF_OVERWRITE=0 (Default) | 1
Indicates whether an existing unversioned repository or existing workspace file is overwritten by a new file with a same path. See "LSAF_OVERWRITE=0 (Default) | 1" on page 6 for more information about this argument.

LSAF_ENABLEVERSIONING=0 (Default) | 1
Indicates whether versioning is enabled for a new file in the repository. See "LSAF_ENABLEVERSIONING=0 (Default) | 1" on page 6 for more information about this argument.

LSAF_VERSIONTYPE=MAJOR (Default) | MINOR | CUSTOM
Indicates the version type to apply to a new file or a new version of an existing versioned repository file. See "LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM" on page 6 for more information about this argument.

LSAF_CUSTOMVERSION=*version*
The version number to use to create a customized version of a new or existing versioned file. See “LSAF_CUSTOMVERSION=version” on page 6 for more information about this argument.

LSAF_COMMENT=*comment*
The check-in comment to associate with the action of adding a file or file version to the repository. When LSAF_EXPORTLOCATION=WORKSPACE, this value is ignored.

Details

Extracts individual files that have been deleted and that reside in a recycle bin, based on the specified query. The query is contained in a SAS data set. The extracted data is stored in a file that is located in your workspace or in the repository. Each item is in a separate row.

To get information about the necessary columnClass and columnName data, and their data types, can be retrieved by call %LSAF_GETQUERYCOLUMNS with the type RBFILEVERSION.

Sets the value of the macro variable _lsafRBFileVersionExportPath_ the full path to the query results CSV file, which includes the file extension. See “Value Macros” on page 5 for information about value macros.

You must have the Administrative mode privilege to execute the query.

See “Query the Recycle Bin” in *SAS Life Science Analytics Framework: SAS Macro API User’s Guide* for more information about querying the recycle bin, including examples.

%LSAF_QUERYRECYCLEBINITEM Macro

Extracts individual items that have been deleted and that reside in a recycle bin, based on the specified query. The query is contained in a SAS data set.

Category: QueryService Module

Note: This macro sets the standard macro variables and _lsafRBItemExportPath_, which is the full path to the query results CSV file.

Syntax

%LSAF_QUERYRECYCLEBINITEM(LSAF_QUERYDATASET=*name*, LSAF_VALIDATEDDATASET=*name*, LSAF_EXPORTLOCATION=WORKSPACE | REPOSITORY, LSAF_EXPORTPATH=*path*<, LSAF_OVERWRITE=0 | 1><, LSAF_ENABLEVERSIONING=0 | 1><, LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM><, LSAF_CUSTOMVERSION=*version*><, LSAF_COMMENT=*comment*>);

Required Arguments

LSAF_QUERYDATASET=*name*
The name of the input data set that contains the query metadata. See “Data Set Macros” on page 5 for information about specifying the name of the data set.

For more information, see “LSAF_QUERYDATASET=*query-data-set-name*” on page 316.

LSAF_VALIDATEDDATASET=*name*
The name of the output data set that contains the results of the validation of the records from the input data set LSAF_QUERYDATASET. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFQUERYRECYCLEBINITEM.

For more information, see “LSAF_VALIDATEDDATASET=*validated-data-set-name*” on page 318.

LSAF_EXPORTLOCATION=WORKSPACE | REPOSITORY
The case-insensitive output location for the exported CSV file.

LSAF_EXPORTPATH=*path*
The case-sensitive output path for the exported CSV file.

Optional Arguments

LSAF_OVERWRITE=0 (Default) | 1
Indicates whether an existing unversioned repository or existing workspace file is overwritten by a new file with a same path. See “LSAF_OVERWRITE=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_ENABLEVERSIONING=0 (Default) | 1
Indicates whether versioning is enabled for a new file in the repository. See “LSAF_ENABLEVERSIONING=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM
Indicates the version type to apply to a new file or a new version of an existing versioned repository file. See “LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM” on page 6 for more information about this argument.

LSAF_CUSTOMVERSION=*version*
The version number to use to create a customized version of a new or existing versioned file. See “LSAF_CUSTOMVERSION=version” on page 6 for more information about this argument.

LSAF_COMMENT=*comment*
The check-in comment to associate with the action of adding a file or file version to the repository. When LSAF_EXPORTLOCATION=WORKSPACE, this value is ignored.

Details

Extracts individual items that have been deleted and that reside in a recycle bin, based on the specified query. The query is contained in a SAS data set. The extracted data is stored in a file that is located in your workspace or in the repository. Each item is in a separate row.

To get information about the necessary columnClass and columnName data, and their data types, can be retrieved by call %LSAF_GETQUERYCOLUMNS with the type RBITEM.

Sets the value of the macro variable _lsafRBItemExportPath_ to the full path to the query results CSV file, which includes the file extension. See “Value Macros” on page 5 for information about value macros.

You must have the Administrative mode privilege to execute the query.

See “Query the Recycle Bin” in *SAS Life Science Analytics Framework: SAS Macro API User’s Guide* for more information about querying the recycle bin, including examples.

PART 8

Security Macros

27

AclService Module

Overview

Macros to manage owner and access permission information of a repository item.

Valid Permission Values

A permission argument can have these valid values:

- 1: Allowed.
- -1: Denied.
- 0: Not specified (inherited).

Note: For valid values, you can also call %LSAF_GETSYSTEMCONSTANTS, where `name=PermissionValue`.

Macro Summary Table

Category	Language Elements	Description
AclService Module	%LSAF_GETACLS Macro (p. 350)	Gets the access control list for items in the repository.
	%LSAF_GETOWNER Macro (p. 352)	Gets the current owner of an item in the repository.
	%LSAF_ISOWNER Macro (p. 353)	Indicates whether the specified user is the owner of an item in the repository.
	%LSAF_UPDATEACLS Macro (p. 354)	Sets the access permissions on items, using a SAS data set as input.
	%LSAF_UPDATEOWNER Macro (p. 358)	Updates the owner of an item in the repository.

Dictionary

%LSAF_GETACLS Macro

Gets the access control list for items in the repository.

Category: AclService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETACLS(LSAF_PATH=*path*<, SAS_DSNAME=*name*><, LSAF_RECURSIVE=*recursion-level*>);

Required Argument

LSAF_PATH=*path*
The path of the content item in the repository. This can be either a file or a container.

Optional Arguments

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the access control lists. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETACLS.

The data set contains a row for each access control entry, sorted by path and ACL type, and the following columns:

Column Name	Description
name	The name of the item.
path	The path to the item.
itemType	The type of the item (such as folder or data set).
isContainer	Indicates whether the item is a container.
owner	The identifier of the user who owns the item.
aclType	The type of ACL. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=AclType`.
aclPrincipalType	The type of principal. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=AceType`.
aclPrincipalName	■ If the aclPrincipalType is ACLOWNER, this is “Owner”. ■ If the aclPrincipalType is ACLMEMBERS, this is “Members”. ■ If the aclPrincipalType is USER, this is the identifier of the user. ■ If the aclPrincipalType is GROUP, this is the name of the user group.
grpSrcCtxt	For user group principals, specifies the context path in which the user group is defined.

Column Name	Description
adminPermission	The access permission to administer the item.
readPermission	The access permission to read the item. Valid values:
writePropPerm	The access permission to update the item properties.
writeContentPerm	The access permission to update the item content.
deletePermission	The access permission to delete the item.

Note: See "Valid Permission Values" on page 349 for valid permission values.

LSAF_RECURSIVE=*recursion-level*
The level of recursion. Required when the path is a container. Valid values:

- 0 (Default): Returns ACLs for the input container only.
- 1 - 99: Returns the container and traverses the tree with the number of levels indicated.

Details

Gets the access control list for items in the repository. The access control list is stored in a SAS data set.

For containers, both the default and current access control lists are returned. For files, only the current access control list is returned.

%LSAF_GETOWNER Macro

Gets the current owner of an item in the repository.

Category: AclService Module

Note: This macro sets the standard macro variables and _lsafOwner_, which is the current owner of the item.

Syntax

%*%LSAF_GETOWNER(LSAF_PATH=*path*<, LSAF_ACLTYPE =*acl-type*>);

Required Argument

LSAF_PATH=*path*
The path of the item in the repository.

Optional Argument

LSAF_ACLTYPE =*acl-type*
The type of access permissions. If the item is a container, this value is required. If the item is a file, this value defaults to CURRENT. The value is case-insensitive. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=AclType`.

Details

Sets the value of the macro variable _lsafOwner_ to the user identifier of the current owner of an item in the repository. In the event that no user is specified as the owner of the default access permissions of a container item, _lsafOwner_ is set to <*creator*>. See "Value Macros" on page 5 for information about value macros.

%LSAF_ISOWNER Macro

Indicates whether the specified user is the owner of an item in the repository.

Category: AclService Module

Note: This macro sets the standard macro variables and _lsafIsOwner_, which indicates whether the user is the owner of the item.

Syntax

%LSAF_ISOWNER(LSAF_PATH=*path*, LSAF_USERID=*user-id*<, LSAF_ACLTYPE=*acl-type*>);

Required Arguments

LSAF_PATH=*path*
The path of the item in the repository

LSAF_USERID=*user-id* | <creator>
The user account identifier or <creator>.

Optional Argument

LSAF_ACLTYPE=*acl-type*
The type of access permissions. If the item is a container, this value is required. If the item is a file, this value defaults to CURRENT. The value is case-insensitive. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=AclType`.

Details

Sets the Boolean macro variable _lsafIsOwner_, which indicates whether the specified user value is the valid owner of the item. See "Boolean Macros" on page 4 for information about return values.

%LSAF_UPDATEACLS Macro

Sets the access permissions on items, using a SAS data set as input.

Category: AclService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_UPDATEACLS(SAS_DSNAME=*name*);

Required Argument

SAS_DSNAME=*name*
The name of the output data set that contains the access permission information for all of the items to modify. See "Data Set Macros" on page 5 for information about specifying the name of the data set.

The data set must contain at least the following columns. Additional columns are ignored.

Column Name	Description
path	The case-sensitive path of the item, which includes the item name.
aclType	The ACL to modify for the item and is case-insensitive. For files, the value must be CURRENT or blank. For containers, the value must be specified. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=AclType`.

Column Name	Description
aclPrincipalType	The type of ACE to update. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=AceType`.
aclPrincipalName	The name of the principal ACE to update or add. ■ If the aclPrincipalType is USER, this is the identifier of the user. ■ If the aclPrincipalType is GROUP, this is the name of the user group. ■ Otherwise, the variable must be present, but the value is ignored.
grpSrcCtxt	■ If aclPrincipalType is GROUP, the case-sensitive path of the context in which the user group is defined. This value is required. ■ For other values of aclPrincipalType, the variable must be present but the value is ignored.
adminPermission	The access permission to administer the item.
readPermission	The access permission to read the item.
writePropPerm	The access permission to update the item properties.
writeContentPerm	The access permission to update the item content.
deletePermission	The access permission to delete the item.

Note: See "Valid Permission Values" on page 349 for valid permission values.

Details

Sets the permissions on items, using a SAS data set as input.

The access permission lists (known as Access Control Lists or ACLs) for multiple items can be modified by specifying a single data set. Access permissions are applied only to items in the repository.

There are two types of access permission lists: current and default.

The current access permissions are applied to an individual item. The default access permissions apply only to container items (folders and contexts) and define the permissions that are inherited by new items that are added to the container.

For this macro, the access permission type is referred to as aclType. All records in the input data set that pertain to the same repository item and aclType are processed together as a single ACL.

A set of permissions for a specific repository item for an individual principal within an access permission type is referred to as an Access Control Entry (ACE). An ACE is represented as a single row in the input data set.

An ACE is defined by these characteristics:

- path: The full path of the repository item (file or container).
- ACL type: The access permission type (DEFAULT or CURRENT).
- ACL principal type: The type of the principal. Valid values:
  - ACLOWNER
  - ACLMEMBERS
  - USER
  - GROUP
- ACL principal name: The name of the user or user group.
- A set of permissions: The specific permissions to assign (such as Administration, Read, Properties Write, Content Write, or Delete).

The output data set from the %LSAF_GETACLS macro contains all of the necessary variables and data needed as the input data set for the %LSAF_UPDATEACLS macro. Modify the data set with changes before calling %LSAF_UPDATEACLS.

To add a principal with specific access permissions to a repository item, add a new row to the data set for that principal.

To modify the access permissions for a principal who is associated with a repository item, edit the appropriate row in the data set.

To remove a principal with specific access permissions from a repository item, delete the appropriate row from the data set. Note that the ACLOWNER and ACLMEMBERS access control entries cannot be removed from the item. If they are removed from the data set, a failure occurs.

To maintain the current permissions for a principal who is associated with a repository item, leave the record as it is in the data set. The record must remain, otherwise the access permissions for this principal are removed.

Failure and Message Handling

The macro performs a series of validation procedures to determine that the input data set has the necessary structure for processing. If any of these validation procedures fail, an appropriate failure message is printed to the SAS log file and the macro stops additional processing.

In addition to other modifications, the macro sorts the input data set and stores the modified data in a temporary data set called WORK.__SASMACRO_ACLTEMP__. In the event that a failure occurs during processing, this temporary data set is retained as a reference. If all updates are performed successfully, this data set is deleted from the work library. A message is printed to the SAS log file that indicates

whether the data set was retained. Regardless of the success or failure of previous runs, this data set is deleted at the beginning of each new run of the macro. The data values and sort order of the original input data set are maintained.

All records that pertain to the same repository item and aclType are processed as a single access control list by the macro. Messages are printed to the SAS log file as each ACL is processed. If the update of an ACL fails, the macro stops processing.

If a failure occurs while processing a specific access control entry (such as a principal), the macro stops processing and a failure message is written to the SAS log file. Successful updates to ACLs made prior to the observation that contains the failure are maintained in SAS Life Science Analytics Framework. Changes made to access permissions for entries that are part of the same ACL that contained the observation with the failure are lost.

Example: Get the Access Permissions for a Folder

```
* start code example;
%lsaf_getacls(lsaf_path=%str(/SAS/Study1/Files/folder1),
lsaf_recursive=0);

data UpdateACLS;
set lsafGetACLS end=eof;

output;

if eof then do;
* Adding user permission under DEFAULT;
aclType = "DEFAULT";
aclPrincipalType = "USER";
aclPrincipalName = "user1";
grpSrcCtxt = "";
adminPermission = 0;
readPermission = 1;
writePropPerm = 1;
writeContentPerm = 1;
deletePermission = 1;
output;

* Adding group permission under CURRENT;
aclType = "CURRENT";
aclPrincipalType = "GROUP";
aclPrincipalName = "group1";
grpSrcCtxt = "/SAS";
adminPermission = 0;
readPermission = 1;
writePropPerm = 1;
writeContentPerm = 1;
deletePermission = 1;
output;
end;

run;
```

```
%lsaf_updateacls(sas_dsname=UpdateACLS);
* end code example;
```

Note: User or user group Access Control Entries (ACEs) with values of 0 for all access permission variables result in a validation error that causes the processing to stop.

Note: Users and user groups to add as ACEs must be members of the context that contains the item that is being updated.

%LSAF_UPDATEOWNER Macro

Updates the owner of an item in the repository.

Category: AclService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_UPDATEOWNER(LSAF_PATH=*path*, LSAF_USERID=*user-id*<, LSAF_ACLTYPE=*acl-type*>);

Required Arguments

LSAF_PATH=*path*
The path to the item in the repository.

LSAF_USERID=*user-id*
The identifier for the user account to assign as the owner. To set the owner for the DEFAULT permissions to the creator of the item, specify the value <creator>.

Optional Argument

LSAF_ACLTYPE=*acl-type*
The type of access permissions. If the item is a container, this value is required. If the item is a file, this value defaults to CURRENT. The value is case-insensitive. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=AclType`.

28

AuditService Module

Overview

Macros to access the audit history metadata.

Macro Summary Table

Category	Language Elements	Description
AuditService Module	%LSAF_GETAUDITACTIONS Macro (p. 360)	Gets the metadata for the auditable actions that are applicable to the specified object type. If no object type is specified, gets all of the auditable actions.

Dictionary

%LSAF_GETAUDITACTIONS Macro

Gets the metadata for the auditable actions that are applicable to the specified object type. If no object type is specified, gets all of the auditable actions.

Category: AuditService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETAUDITACTIONS(<LSAF_TYPEID=*type-id*><, SAS_DSNAME=data-set-name>);

Optional Arguments

LSAF_TYPEID=*type-id*
The case-insensitive object type identifier.

SAS_DSNAME=data-set-name
The name of the output data set that contains the metadata for all the audit actions. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETAUDITACTIONS.

The data set contains a row for each audit action and columns with the following names:

Column Name	Description
typeID	The object type identifier for which the audit action applies.
action	The audit action.
displayName	The display name of the audit action.

Details

Gets the metadata for the auditable actions that are applicable to the specified object type. If no object type is specified, gets all of the auditable actions. The metadata is stored in a SAS data set and is sorted by typeID and action.

To get a list of auditable object type identifiers, call one of the following macros:

- %LSAF_GETALLTYPES
- %LSAF_GETCONTEXTTYPES
- %LSAF_GETTYPESBYCAPABILITY
- %LSAF_GETSEARCHABLETYPES

29

AuthenticationService Module

Overview

Macros to authenticate user access to SAS Life Science Analytics Framework.

Macro Summary Table

Category	Language Elements	Description
AuthenticationService Module	%LSAF_LOGIN Macro (p. 364)	Creates a connection to SAS Life Science Analytics Framework.
	%LSAF_LOGOUT Macro (p. 365)	Closes an open connection to SAS Life Science Analytics Framework.

Dictionary

%LSAF_LOGIN Macro

Creates a connection to SAS Life Science Analytics Framework.

Category: AuthenticationService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_LOGIN(LSAF_URL=*url*, LSAF_USERID=*user-id*, LSAF_PASSWORD=*password*<, PROXY_HOST=*host*><, PROXY_PORT=*port*><, PROXY_USERID=*proxy-user-id*><, PROXY_PASSWORD=*proxy-password*>);

Required Arguments

LSAF_URL=*url*
The URL of the SAS Life Science Analytics Framework instance to access, specified in the form https://lsafname.sas.com or http://lsafname.sas.com.

LSAF_USERID=*user-id*
The SAS Life Science Analytics Framework identifier of the user account.

LSAF_PASSWORD=*password*
The SAS Life Science Analytics Framework password for the user account.

Optional Arguments

PROXY_HOST=*host*
The URL to the host computer, when SAS Life Science Analytics Framework is accessed through a proxy server.

PROXY_PORT=*port*
The port number that is required for the proxy host computer. Required when a proxy_host is specified.

PROXY_USERID=*proxy-user-id*
The identifier when the proxy computer requires authentication.

PROXY_PASSWORD=*proxy-password*
The password when the proxy computer requires authentication.

Details

Creates a connection to the SAS Life Science Analytics Framework instance. This macro is required only when running the macros from a PC SAS session. It must be called before any other SAS Life Science Analytics Framework macro.

Note: This macro is supported only from a PC SAS session. Calling this macro within an active SAS Life Science Analytics Framework session results in a failure.

Examples

Example 1: Special Characters Might Need to Be Wrapped in a SAS Function Call

The parameters that are passed to the login macro are often strings that contain special characters that might cause problems when resolved by SAS. It is recommended to use a SAS function call for each parameter to prevent this issue, such as:

```
%lsaf_login(lsaf_url=%str(https://lsafinstance), lsaf_userid=%str(lsaf-
user-id), lsaf_password=%str (lsafpassword ));
```

Example 2: Special Characters Might Need Quotation Marks

Certain special characters that are common in passwords (such as ampersands and percent signs) might require macro quotation marks, such as:

```
%lsaf_login(lsaf_url=%str(https://lsafinstance), lsaf_userid=%str(lsaf-
user-id), lsaf_password=%nrbquote(lsafpassword ));
```

%LSAF_LOGOUT Macro

Closes an open connection to SAS Life Science Analytics Framework.

Category: AuthenticationService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_LOGOUT;

Details

Closes an open connection to SAS Life Science Analytics Framework. This macro is used only when running the macros from a PC SAS session. This should be the last macro you call in a program. If the logout operation is successful, all variables (such as _lsafRC_) that are related to SAS Life Science Analytics Framework macros are deleted.

Note: This macro is supported only from a PC SAS session. Calling this macro within an active SAS Life Science Analytics Framework session results in a failure.

30

GlobalPrivilegeService Module

Overview

Macros to manage the privileges that are available to user accounts at the global level.

Macro Summary Table

Category	Language Elements	Description
GlobalPrivilegeService Module	%LSAF_ADDUSERPRIVILEGE Macro (p. 368)	Assigns a global privilege to a user account.
	%LSAF_GETGLOBALPRIVILEGES Macro (p. 369)	Gets the metadata for all of the privileges that are available at the global level. The metadata is stored in a SAS data set.

Category	Language Elements	Description
	%LSAF_GETUSERPRIVILEGES Macro (p. 369)	Gets the metadata for all of the global privileges that are assigned to a user account. The metadata is stored in a SAS data set.
	%LSAF_GETUSERSWITHGLOBALPRIVILEGE Macro (p. 370)	Gets all of the user accounts that have the specified privilege identifier. The properties are stored in a SAS data set.
	%LSAF_HASGLOBALPRIVILEGE Macro (p. 371)	Indicates whether a global privilege is assigned to a user account.
	%LSAF_ISGLOBALPRIVILEGE Macro (p. 372)	Indicates whether the specified value is a valid global privilege.
	%LSAF_REMOVEUSERPRIVILEGE Macro (p. 373)	Removes a global privilege from a user account.

Dictionary

%LSAF_ADDUSERPRIVILEGE Macro

Assigns a global privilege to a user account.

Category: GlobalPrivilegeService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_ADDUSERPRIVILEGE(LSAF_USERID=*user-id*, LSAF_PRIVILEGE=*privilege*);

Required Arguments

LSAF_USERID=*user-id*
DESCRIPTION

LSAF_PRIVILEGE=*privilege*
The identifier of the global privilege. To get the list of available global privileges, call %LSAF_GETGLOBALPRIVILEGES.

%LSAF_GETGLOBALPRIVILEGES Macro

Gets the metadata for all of the privileges that are available at the global level. The metadata is stored in a SAS data set.

Category: GlobalPrivilegeService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETGLOBALPRIVILEGES(<SAS_DSNAME=*name*>);

Optional Argument

SAS_DSNAME=*name*

The name of the output data set that contains the metadata for all of the global privileges. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETGLOBALPRIVILEGES.

The data set contains a row for each global privilege, sorted by privilegeId, and the following columns:

Column Name	Description
privilegeType	The privilege type: GLOBAL.
privilegeId	The identifier of the privilege.
privilegeName	The name of the privilege.

%LSAF_GETUSERPRIVILEGES Macro

Gets the metadata for all of the global privileges that are assigned to a user account. The metadata is stored in a SAS data set.

Category: GlobalPrivilegeService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETUSERPRIVILEGES(LSAF_USERID=*user-id*<, SAS_DSNAME=*name*>);

Required Argument

LSAF_USERID=*user-id*
The identifier of the user account.

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the global privileges. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETUSERPRIVILEGES.

The data set contains a row for each global privilege that is assigned to the user account, sorted by privilegeId, and the following columns:

Column Name	Description
userId	The identifier of the user.
privilegeId	The identifier of the privilege.
privilegeName	The name of the privilege.

%LSAF_GETUSERSWITHGLOBALPRIVILEGE Macro

Gets all of the user accounts that have the specified privilege identifier. The properties are stored in a SAS data set.

Category: GlobalPrivilegeService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETUSERSWITHGLOBALPRIVILEGE(LSAF_PRIVILEGEID=*privilege-id*<, SAS_DSNAME=*name*>);

Required Argument

LSAF_PRIVILEGEID=*privilege-id*
The identifier of the global privilege. To get the list of the privileges, call %LSAF_GETGLOBALPRIVILEGES.

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the user accounts with the corresponding privilege identifier. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETUSERSWITHGLOBALPRIVILEGE.

The data set contains a row for each user account, sorted by user account name, and the following columns:

Column Name	Description
privilegeType	The privilege type. Valid value: GLOBAL.
privilegeId	The identifier of the privilege.
privilegeName	The name of the privilege.
userId	The identifier of the user account.
displayName	The user account name.

%LSAF_HASGLOBALPRIVILEGE Macro

Indicates whether a global privilege is assigned to a user account.

Category: GlobalPrivilegeService Module

Note: This macro sets the standard macro variables and _lsafHasGlobalPrivilege_, which indicates whether the privilege is assigned to the user account.

Syntax

%LSAF_HASGLOBALPRIVILEGE(LSAF_USERID=*user-id*, LSAF_PRIVILEGE=*privilege-id*);

Required Arguments

LSAF_USERID=*user-id*
The identifier of the user account.

LSAF_PRIVILEGE=*privilege-id*
The identifier of the global privilege. To get the list of the privileges, call %LSAF_GETGLOBALPRIVILEGES.

Details

Sets the Boolean macro variable _lsafHasGlobalPrivilege_, which indicates whether a global privilege is assigned to a user account. See “Boolean Macros” on page 4 for information about return values.

%LSAF_ISGLOBALPRIVILEGE Macro

Indicates whether the specified value is a valid global privilege.

Category: GlobalPrivilegeService Module

Note: This macro sets the standard macro variables and _lsafIsGlobalPrivilege_, which indicates whether the item is a valid global privilege.

Syntax

%LSAF_ISGLOBALPRIVILEGE(LSAF_PRIVILEGE=*privilege-id*);

Required Argument

LSAF_PRIVILEGE=*privilege-id*
The global privilege identifier. To get the list of the privileges, call %LSAF_GETGLOBALPRIVILEGES.

Details

Sets the Boolean macro variable _lsafIsGlobalPrivilege_, which indicates whether the specified value is a valid global privilege. See “Boolean Macros” on page 4 for information about return values.

%LSAF_REMOVEUSERPRIVILEGE Macro

Removes a global privilege from a user account.

Category: GlobalPrivilegeService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_REMOVEUSERPRIVILEGE(LSAF_USERID=*user-id*, LSAF_PRIVILEGE=*privilege-id*);

Required Arguments

LSAF_USERID=*user-id*
The identifier of the user account.

LSAF_PRIVILEGE=*privilege-id*
The identifier of the global privilege. To get the list of the privileges, call %LSAF_GETGLOBALPRIVILEGES.

31

GroupService Module

Overview

Macros to manage and create user groups, their availability in contexts, and their membership lists.

Macro Summary Table

Category	Language Elements	Description
GroupService Module	%LSAF_ADDGROUPMEMBER Macro (p. 376)	Adds a user or user group as a member of a user group in the specified context.

Category	Language Elements	Description
	%LSAF_CREATEGROUP Macro (p. 377)	Creates a user group in the specified context.
	%LSAF_DELETEGROUP Macro (p. 378)	Deletes a user group from the specified context.
	%LSAF_GETGROUPMEMBERS Macro (p. 378)	Gets the metadata for all of the members that are assigned to a user group within the specified context. The metadata is stored in a SAS data set.
	%LSAF_GETGROUPS Macro (p. 379)	Gets the metadata for all of the user groups that are defined within the specified context. The metadata is stored in a SAS data set.
	%LSAF_GETUSERGROUPS Macro (p. 380)	Gets the metadata for each of the user groups in which the specified user is a member. The metadata is stored in a SAS data set.
	%LSAF_GROUPEXISTS Macro (p. 381)	Indicates whether a user group is defined for the specified context.
	%LSAF_ISGROUPMEMBER Macro (p. 382)	Indicates whether a user or user group is a member of the user group in the specified context.
	%LSAF_REMOVEGROUPMEMBER Macro (p. 383)	Removes a user or a user group as a member of a user group in the specified context.

Dictionary

%LSAF_ADDGROUPMEMBER Macro

Adds a user or user group as a member of a user group in the specified context.

Category: GroupService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_ADDGROUPMEMBER(LSAF_PATH=*path*, LSAF_GROUP=*user-group-name*, LSAF_MEMBER=*member-id*<, LSAF_TYPE=USER | GROUP><, LSAF_GROUP_CONTEXT=*context*>);

Required Arguments

LSAF_PATH=*path*
The path of the context.

LSAF_GROUP=*user-group-name*
The name of the user group in which to add the member.

LSAF_MEMBER=*member-id*
The identifier of the user or the name of the user group to add.

Optional Arguments

LSAF_TYPE=USER (Default) | GROUP
The type of member to add.

LSAF_GROUP_CONTEXT=*context*
The context in which the user group is defined, specified as a path (such as / SAS). If LSAF_TYPE is GROUP, this argument is required. Otherwise, it is ignored.

%LSAF_CREATEGROUP Macro

Creates a user group in the specified context.

Category: GroupService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_CREATEGROUP(LSAF_PATH=*path*, LSAF_GROUP=*user-group-name*<, LSAF_DESCRIPTION=*description*>);

Required Arguments

LSAF_PATH=*path*
The path of the context.

LSAF_GROUP=*user-group-name*
The name of the user group.

Optional Argument

LSAF_DESCRIPTION=*description*
The description for the user group.

%LSAF_DELETEGROUP Macro

Deletes a user group from the specified context.

Category: GroupService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_DELETEGROUP(LSAF_PATH=*path*, LSAF_GROUP=*user-group-name*);

Required Arguments

LSAF_PATH=*path*
The path of the context in which the user group is defined.

LSAF_GROUP=*user-group-name*
The name of the user group.

%LSAF_GETGROUPMEMBERS Macro

Gets the metadata for all of the members that are assigned to a user group within the specified context. The metadata is stored in a SAS data set.

Category: GroupService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETGROUPMEMBERS(LSAF_PATH=*path*, LSAF_GROUP=*user-group-name*<, SAS_DSNAME=*name*>);

Required Arguments

LSAF_PATH=*path*
The path of the context in which the user group is defined.

LSAF_GROUP=*user-group-name*
The name of the user group.

Optional Argument

SAS_DSNAME=*name*

The name of the output data set that contains the metadata for the members. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETGROUPMEMBERS.

The data set contains a row for each member, sorted by member type and principalId, and the following columns:

Column Name	Description
contextPath	The full path of the context item in the repository.
group	The name of the user group that the members are associated with.
type	The member type. Valid values are USER and GROUP.
principalId	The name of the member. Valid values are the user group name or user identifier.
description	The description of the member. Valid values are user group name or user display name.
grpSrcCtxt	The context in which the user group member is defined.

%LSAF_GETGROUPS Macro

Gets the metadata for all of the user groups that are defined within the specified context. The metadata is stored in a SAS data set.

Category: GroupService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETGROUPS(LSAF_PATH=*path*<, SAS_DSNAME=*name*>);

Required Argument

LSAF_PATH=*path*
The path of the context in which the user groups are defined.

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for all of the user groups. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETGROUPS.

The data set contains a row for each user group, sorted by user group name, and the following columns:

Column Name	Description
contextPath	The path of the context item in the repository.
name	The name of the user group.
description	The user group description.

%LSAF_GETUSERGROUPS Macro

Gets the metadata for each of the user groups in which the specified user is a member. The metadata is stored in a SAS data set.

Category: GroupService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETUSERGROUPS(LSAF_USERID=*user-id*<, LSAF_PATH=*path*><, SAS_DSNAME=*name*>);

Required Argument

LSAF_USERID=*user-id*
The identifier of the user for whom to retrieve the user groups.

Optional Arguments

LSAF_PATH=*path*
The context path of the groups to be returned. If not specified, all the user's groups are returned.

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the user groups. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETUSERGROUPS.

The data set contains a row for each user group, sorted by the user group name and user group context path, and the following columns:

Column Name	Description
userId	The identifier of the user.
name	The name of the user group.
description	The description of the user group.
grpSrcCtxt	The context path in which the user group is defined.

%LSAF_GROUPEXISTS Macro

Indicates whether a user group is defined for the specified context.

Category: GroupService Module

Note: This macro sets the standard macro variables and _lsafGroupExists_, which indicates whether the user group exists in the context.

Syntax

%LSAF_GROUPEXISTS(LSAF_PATH=*path*, LSAF_GROUP=*user-group-name*);

Required Arguments

LSAF_PATH=*path*
The path of the context in which the user groups are defined.

LSAF_GROUP=*user-group-name*
The name of the user group.

Details

Sets the Boolean macro variable _lsafGroupExists_, which indicates whether a user group is defined for the specified context. See “Boolean Macros” on page 4 for information about return values.

%LSAF_ISGROUPMEMBER Macro

Indicates whether a user or user group is a member of the user group in the specified context.

Category: GroupService Module

Note: This macro sets the standard macro variables and _lsafIsGroupMember_, which indicates whether the user or user group is a member of the specified user group.

Syntax

%LSAF_ISGROUPMEMBER(LSAF_PATH=*path*, LSAF_GROUP=*user-group-name*, LSAF_MEMBER=*member*<, LSAF_TYPE=USER | GROUP><, LSAF_GROUP_CONTEXT=*context*><, LSAF_INCLUDEIMPLICIT=0 | 1>);

Required Arguments

LSAF_PATH=*path*
The path of the context in which the user group is assigned.

LSAF_GROUP=*user-group-name*
The name of the user group.

LSAF_MEMBER=*member*
The identifier or user group name of the member.

Optional Arguments

LSAF_TYPE=USER (Default) | GROUP
The type of member. The values are case-insensitive. An invalid value results in the member not being found.

LSAF_GROUP_CONTEXT=*context*
The context in which the member user group is defined, specified as a path (such as /SAS). If LSAF_TYPE is GROUP, this argument is required. Otherwise, it is ignored.

LSAF_INCLUDEIMPLICIT=0 (Default) | 1
Indicates whether to include the implied members, such as the members of user groups within user groups. The value 0 excludes the implied members, the value 1 includes them.

If this value is not specified or an incorrect value is specified, it defaults to 0.

Details

Sets the Boolean macro variable _lsafIsGroupMember_, which indicates whether a user or user group is a member of the user group in the specified context. A flag can be set to indicate whether the search for membership includes implicit membership. See “Boolean Macros” on page 4 for information about return values.

%LSAF_REMOVEGROUPMEMBER Macro

Removes a user or a user group as a member of a user group in the specified context.

Category: GroupService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_REMOVEGROUPMEMBER(LSAF_PATH=*path*, LSAF_GROUP=*user-group-name*, LSAF_MEMBER=*member*<, LSAF_TYPE=USER | GROUP><, LSAF_GROUP_CONTEXT=*context*>);

Required Arguments

LSAF_PATH=*path*
The path of the context in which the user group is defined.

LSAF_GROUP=*user-group-name*
The name of the user group from which to remove the member.

LSAF_MEMBER=*member*
The identifier of the user or the name of the user group to remove.

Optional Arguments

LSAF_TYPE=USER (Default) | GROUP
The type of the member. The values are case-insensitive.

LSAF_GROUP_CONTEXT=*context*
The context in which the user group to remove is defined, specified as a path (such as /SAS). If LSAF_TYPE=GROUP, this argument is required. Otherwise, it is ignored.

32

MembershipService Module

Overview

Macros to manage and retrieve information about the members of a context.

Macro Summary Table

Category	Language Elements	Description
MembershipService Module	%LSAF_ADDMEMBER Macro (p. 386)	Adds a user or a user group as a member of the specified context.
	%LSAF_GETASSIGNEDMEMBERS Macro (p. 387)	Gets the metadata for all of the members that are assigned to the specified context. The metadata is stored in a SAS data set.
	%LSAF_GETPOTENTIALMEMBERS Macro (p. 388)	Gets the metadata for all of the users or user groups that are available to add as members to the specified context. The metadata is stored in a SAS data set.

Category	Language Elements	Description
	%LSAF_ISMEMBER Macro (p. 389)	Indicates whether a user or user group is a member in the specified context.
	%LSAF_REMOVEMEMBER Macro (p. 390)	Removes a user or a user group as a member of the specified context.

Dictionary

%LSAF_ADDMEMBER Macro

Adds a user or a user group as a member of the specified context.

Category: MembershipService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_ADDMEMBER(LSAF_PATH=*path*, LSAF_MEMBER=*member*<, LSAF_TYPE=USER | GROUP><, LSAF_GROUP_CONTEXT=*context*>);

Required Arguments

LSAF_PATH=*path*
The path of the context to which to add the member.

LSAF_MEMBER=*member*
The identifier of the user or the name of the user group.

Optional Arguments

LSAF_TYPE=USER (Default) | GROUP
The type of member. The values are case-insensitive.

LSAF_GROUP_CONTEXT=*context*
The context in which the user group is defined, specified as a path (such as / SAS). If LSAF_TYPE=GROUP, this argument is required. Otherwise, it is ignored.

%LSAF_GETASSIGNEDMEMBERS Macro

Gets the metadata for all of the members that are assigned to the specified context. The metadata is stored in a SAS data set.

Category: MembershipService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETASSIGNEDMEMBERS(LSAF_PATH=*path*<, SAS_DSNAME=*name*>);

Required Argument

LSAF_PATH=*path*
The path of the context that contains the members.

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the members. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETASSIGNEDMEMBERS.

The data set contains a row for each member, sorted by member type and principalId, and the following columns:

Column Name	Description
contextPath	The path of the context in which the members are assigned.
type	The member type. Valid values: ■ USER ■ GROUP
principalId	The name of the member. The returned value is either a user group name or an identifier.
description	The description of the member. The returned value is either a user group name or a user display name.

Column Name	Description
grpSrcCtxt	If the member is a user group, the context path in which the user group member is defined.

%LSAF_GETPOTENTIALMEMBERS Macro

Gets the metadata for all of the users or user groups that are available to add as members to the specified context. The metadata is stored in a SAS data set.

Category: MembershipService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETPOTENTIALMEMBERS(LSAF_PATH=*path*<, SAS_DSNAME=*name*>);

Required Argument

LSAF_PATH=*path*
The path of the context to determine potential members.

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the potential members. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETPOTENTIALMEMBERS.

The data set contains a row for each potential member, sorted by member type and principalId, and the following columns:

Column Name	Description
contextPath	The path of the context in which the members are assigned.
type	The member type. Valid values: ■ USER ■ GROUP

Column Name	Description
principalId	The name of the member. The returned value is either a user group name or an identifier.
description	The description of the member. The returned value is either a user group name or a user display name.
grpSrcCtxt	If the member is a user group, the context path in which the user group member is defined.

%LSAF_ISMEMBER Macro

Indicates whether a user or user group is a member in the specified context.

Category: MembershipService Module

Note: This macro sets the standard macro variables and _lsafIsMember_, which indicates whether the user or user group is a member of the specified context.

Syntax

%LSAF_ISMEMBER(LSAF_PATH=*path*, LSAF_MEMBER=*member*<, LSAF_TYPE=USER | GROUP><, LSAF_GROUP_CONTEXT=*context*><, LSAF_INCLUDEIMPLICIT=0 | 1>);

Required Arguments

LSAF_PATH=*path*
The absolute path of the context.

LSAF_MEMBER=*member*
The identifier of the user or user group name of the member.

Optional Arguments

LSAF_TYPE=USER (Default) | GROUP
The type of member. The values are case-insensitive.

LSAF_GROUP_CONTEXT=*context*
The context in which the member user group is defined, specified as a path (such as /SAS). If LSAF_TYPE=GROUP, this argument is required. Otherwise, it is ignored.

LSAF_INCLUDEIMPLICIT=0 (Default) | 1
Indicates whether to report that the user is an implicit member of the context through user group membership. The default value is 0, which indicates whether the user is an explicit member of the context only.

Details

Sets the Boolean macro variable _lsafIsMember_, which indicates whether a user or user group is a member in the specified context. Groups defined at the context level are not considered as members of that context. See “Boolean Macros” on page 4 for information about return values.

%LSAF_REMOVEMEMBER Macro

Removes a user or a user group as a member of the specified context.

Category: MembershipService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_REMOVEMEMBER(LSAF_PATH=*path*, LSAF_MEMBER=*member*<, LSAF_TYPE=USER | GROUP><, LSAF_GROUP_CONTEXT=*context*>);

Required Arguments

LSAF_PATH=*path*
The path of the context from which to remove the member.

LSAF_MEMBER=*member*
The identifier of the user or the name of the user group.

Optional Arguments

LSAF_TYPE=USER (Default) | GROUP
The type of the member. The values are case-insensitive.

LSAF_GROUP_CONTEXT=*context*
The context in which the user group to remove is defined, specified as a path (such as /SAS). If LSAF_TYPE=GROUP, this argument is required. Otherwise, it is ignored.

33

RoleService Module

Overview

Macros to manage and create roles, their availability in contexts, and their membership lists.

Macro Summary Table

Category	Language Elements	Description
RoleService Module	%LSAF_ADDINHERITEDROLE Macro (p. 393)	Assigns a role to the specified context as an inherited role.
	%LSAF_ADDROLEMEMBER Macro (p. 394)	Adds a user or user group as a member of a role in the specified context.
	%LSAF_ADDROLEPRIVILEGE Macro (p. 394)	Adds a privilege to a role in the specified context.
	%LSAF_CREATEROLE Macro (p. 395)	Creates a role in the specified context.
	%LSAF_DELETEROLE Macro (p. 396)	Deletes a role that is defined in the specified context.
	%LSAF_GETINHERITEDROLES Macro (p. 396)	Gets all inherited roles across contexts that are associated with the specified role.
	%LSAF_GETROLEMEMBERS Macro (p. 397)	Gets the metadata for all of the members that are assigned to a role within the specified context. The metadata is stored in a SAS data set.
	%LSAF_GETROLEPRIVILEGES Macro (p. 398)	Gets the metadata for all of the privileges that are assigned to a role. The metadata is stored in a SAS data set.
	%LSAF_GETROLES Macro (p. 399)	Gets the metadata for all of the roles that are defined and inherited in a context. The metadata is stored in a SAS data set.
	%LSAF_GETROLESBYPRIVILEGE Macro (p. 400)	Gets the metadata for all of the roles in a given context that include the specified privilege.
	%LSAF_HASSCOPEDPRIVILEGE Macro (p. 401)	Indicates whether a user or user group has a specific scoped privilege in a context.
	%LSAF_ISROLEMEMBER Macro (p. 402)	Indicates whether a user or user group is a member of a role in the specified context.
	%LSAF_ISROLEPRIVILEGE Macro (p. 403)	Indicates whether a privilege is assigned to a role in the specified context.
	%LSAF_REMOVEINHERITEDROLE Macro (p. 404)	Removes a role from the specified context.
	%LSAF_REMOVEROLEMEMBER Macro (p. 405)	Removes a user or a user group as a member of a role in the specified context.

Category	Language Elements	Description
	%LSAF_REMOVEROLEPRIVILEGE Macro (p. 405)	Removes a privilege from a role in the specified context.
	%LSAF_RENAMEROLE Macro (p. 406)	Renames a defined role.
	%LSAF_ROLEEXISTS Macro (p. 406)	Indicates whether a role exists as defined or inherited in a specified context.
	%LSAF_UPDATEROLEDESCRIPTION Macro (p. 407)	Updates the description of a defined role.

Dictionary

%LSAF_ADDINHERITEDROLE Macro

Assigns a role to the specified context as an inherited role.

Category: RoleService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_ADDINHERITEDROLE(LSAF_PATH=*path*, LSAF_ROLE=*name*, LSAF_ROLE_CONTEXT=*context*);

Required Arguments

LSAF_PATH=*path*
The path of the context to inherit the role.

LSAF_ROLE=*name*
The name of the role.

LSAF_ROLE_CONTEXT=*context*
The context in which the role is defined, specified as a path (such as /SAS).

%LSAF_ADDROLEMEMBER Macro

Adds a user or user group as a member of a role in the specified context.

Category: RoleService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_ADDROLEMEMBER(LSAF_PATH=*path*, LSAF_ROLE=*name*, LSAF_MEMBER=*member*<, LSAF_ROLE_CONTEXT=*context*><, LSAF_TYPE=USER | GROUP><, LSAF_GROUP_CONTEXT=*context*>);

Required Arguments

LSAF_PATH=*path*
The path of the context item in which the role is assigned (either defined or inherited).

LSAF_ROLE=*name*
The name of the role.

LSAF_MEMBER=*member*
The identifier of the user or the name of the user group to add.

Optional Arguments

LSAF_ROLE_CONTEXT=*context*
The context in which the role is defined, specified as a path (such as /SAS). If not specified, the member is added to the role that is defined in the context that is specified by LSAF_PATH.

LSAF_TYPE=USER (Default) | GROUP
The type of member to add.

LSAF_GROUP_CONTEXT=*context*
The context in which the user group is defined, specified as a path (such as /SAS). If LSAF_TYPE=GROUP, this argument is required. Otherwise, it is ignored.

%LSAF_ADDROLEPRIVILEGE Macro

Adds a privilege to a role in the specified context.

Category: RoleService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_ADDROLEPRIVILEGE(LSAF_PATH=*path*, LSAF_ROLE=*name*, LSAF_PRIVILEGE=*privilege*);

Required Arguments

LSAF_PATH=*path*
The path of the context in which the role is defined.

LSAF_ROLE=*name*
The name of the role.

LSAF_PRIVILEGE=*privilege*
The identifier of the privilege. To get the privilege identifier, call %LSAF_GETSCOPEDPRIVILEGES.

%LSAF_CREATEROLE Macro

Creates a role in the specified context.

Category: RoleService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_CREATEROLE(LSAF_PATH=*path*, LSAF_ROLE=*name*<, LSAF_DESCRIPTION=*description*>);

Required Arguments

LSAF_PATH=*path*
The path of the context in which to create the role.

LSAF_ROLE=*name*
The name of the role.

Optional Argument

LSAF_DESCRIPTION=*description*
The description of the role.

%LSAF_DELETEROLE Macro

Deletes a role that is defined in the specified context.

Category: RoleService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_DELETEROLE(LSAF_PATH=*path*, LSAF_ROLE=*name*);

Required Arguments

LSAF_PATH=*path*
The path of the context in which the role is defined.

LSAF_ROLE=*name*
The name of the role.

%LSAF_GETINHERITEDROLES Macro

Gets all inherited roles across contexts that are associated with the specified role.

Category: RoleService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETINHERITEDROLES(LSAF_PATH=*path*, LSAF_ROLE=*name*<, SAS_DSNAME=*name*>);

Required Arguments

LSAF_PATH=*path*
The path of the context in which the role is defined.

LSAF_ROLE=*name*
The name of the role.

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the inherited roles. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETINHERITEDROLES.

The data set contains a row for each role, sorted by roleSrcCtxt.

Column Name	Description
path	The path of the context in which the role is defined.
name	The name of the role.
roleSrcCtxt	The context where the role is inherited.

%LSAF_GETROLEMEMBERS Macro

Gets the metadata for all of the members that are assigned to a role within the specified context. The metadata is stored in a SAS data set.

Category: RoleService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETROLEMEMBERS(LSAF_PATH=*path*, LSAF_ROLE=*name*<, LSAF_ROLE_CONTEXT=*context*><, SAS_DSNAME=*name*>);

Required Arguments

LSAF_PATH=*path*
The path of the context in which the role is assigned (either defined or inherited).

LSAF_ROLE=*name*
The name of the role.

Optional Arguments

LSAF_ROLE_CONTEXT=*context*
The context in which the role is defined, specified as a path (such as /SAS). If not specified, the members returned are from the role that is specified by LSAF_PATH.

SAS_DSNAME=*name*

The name of the output data set that contains the metadata for the members. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETROLEMEMBERS.

The data set contains a row for each member in the role and columns with the following names, sorted by member type and principalId:

Column Name	Description
contextPath	The path of the context.
name	The name of the role that the members are associated with.
roleSrcCtxt	The context where the role is defined. This is blank if it is not provided as lsaf_role_context.
type	The member type. Valid values: ■ USER ■ GROUP
principalId	The name of the member. Valid values: ■ user group name ■ identifier
descriptor	The description of the member. Valid values: ■ user group name ■ user display name
grpSrcCtxt	If the member is a user group, the context in which the user group member is defined.

%LSAF_GETROLEPRIVILEGES Macro

Gets the metadata for all of the privileges that are assigned to a role. The metadata is stored in a SAS data set.

Category: RoleService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETROLEPRIVILEGES(LSAF_PATH=*path*, LSAF_ROLE=*name*<, SAS_DSNAME=*name*>);

Required Arguments

LSAF_PATH=*path*
The path of the context in which the role is defined.

LSAF_ROLE=*name*
The name of the role.

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the privileges. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETROLEPRIVILEGES.

The data set contains a row for each privilege in the role, sorted by privilegeId, and the following columns:

Column Name	Description
contextPath	The path of the context in which the role is defined.
roleName	The name of the role.
privilegeId	The identifier of the privilege.
privilegeName	The name of the privilege.

%LSAF_GETROLES Macro

Gets the metadata for all of the roles that are defined and inherited in a context. The metadata is stored in a SAS data set.

Category: RoleService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETROLES(LSAF_PATH=*path*<, LSAF_MEMBER=*member*><, LSAF_TYPE=USER | GROUP><, LSAF_GROUP_CONTEXT=*context*><, SAS_DSNAME=*name*>);

Required Argument

LSAF_PATH=*path*
The path of the context in which the roles are located.

Optional Arguments

LSAF_MEMBER=*member*
The identifier of the user or the name of the user group whose roles you want to retrieve. If omitted, all roles in the specified context are returned.

LSAF_TYPE=USER (Default) | GROUP
The type of member, which is ignored if LSAF_MEMBER is not provided and is required if LSAF_MEMBER is a group.

LSAF_GROUP_CONTEXT=*context*
The context in which the member user group is defined, specified as a path (such as /SAS). If LSAF_TYPE=GROUP, this argument is required. Otherwise, it is ignored.

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the roles. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETROLES.

The data set contains a row for each role in the context, sorted by role name, and the following columns:

Column Name	Description
path	The path of the context.
name	The name of the role.
description	The role description.
roleSrcCtxt	The context in which the role is defined.
isInherited	Indicates whether the role is inherited.

%LSAF_GETROLESBYPRIVILEGE Macro

Gets the metadata for all of the roles in a given context that include the specified privilege.

Category: RoleService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETROLESBYPRIVILEGE(LSAF_PATH=*path*, LSAF_PRIVILEGE=*privilege*<, SAS_DSNAME=*name*>);

Required Arguments

LSAF_PATH=*path*
The path of the context in which the roles are located.

LSAF_PRIVILEGE=*privilege*
The identifier of the privilege. To get the identifier, call %LSAF_GETSCOPEDPRIVILEGES.

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the roles. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETROLESBYPRIVILEGE.

The data set contains a row for each role, sorted by roleSrcCtxt.

Column Name	Description
privilegeId	The identifier for the privilege.
contextPath	The path of the context.
name	The name of the role.
description	The role description.
roleSrcCtxt	The context in which the role is inherited.
isInherited	Indicates whether the role is inherited.

%LSAF_HASSCOPEDPRIVILEGE Macro

Indicates whether a user or user group has a specific scoped privilege in a context.

Category: RoleService Module

Note: This macro sets the standard macro variables and _lsafHasScopedPrivilege_, which indicates whether the user or user group has the privilege in the specified context.

Syntax

%LSAF_HASSCOPEDPRIVILEGE(LSAF_PATH=*path*, LSAF_PRIVILEGE=*privilege*, LSAF_MEMBER=*member*<, LSAF_TYPE=USER | GROUP><, LSAF_GROUP_CONTEXT=*context*>);

Required Arguments

LSAF_PATH=*path*
The path of the context.

LSAF_PRIVILEGE=*name*
The identifier of the privilege.

LSAF_MEMBER=*member*
The user identifier or user group name of the member.

Optional Arguments

LSAF_TYPE=USER (Default) | GROUP
The type of member.

LSAF_GROUP_CONTEXT=*context*
The context in which the member user group is defined, specified as a path (such as /SAS). If LSAF_TYPE=GROUP, this argument is required. Otherwise, it is ignored.

Details

Sets the Boolean macro variable _lsafHasScopedPrivilege_, which indicates whether a user or user group has a specific scoped privilege in the specified context. See “Boolean Macros” on page 4 for information about return values.

%LSAF_ISROLEMEMBER Macro

Indicates whether a user or user group is a member of a role in the specified context.

Category: RoleService Module

Note: This macro sets the standard macro variables and _lsafIsRoleMember_, which indicates whether the user or user group is a member of the role in the specified context.

Syntax

%LSAF_ISROLEMEMBER(LSAF_PATH=*path*, LSAF_ROLE=*name*, LSAF_MEMBER=*member*<, LSAF_ROLE_CONTEXT=*context*><, LSAF_TYPE=USER | GROUP><, LSAF_GROUP_CONTEXT=*context*>);

Required Arguments

LSAF_PATH=*path*
The path of the context.

LSAF_ROLE=*name*
The name of the role.

LSAF_MEMBER=*member*
The user identifier or user group name of the member.

Optional Arguments

LSAF_ROLE_CONTEXT=*context*
The context in which the role is defined, specified as a path (such as /SAS). If not specified, the membership is tested in the role that is defined at the path that is specified by LSAF_PATH.

LSAF_TYPE=USER (Default) | GROUP
The type of the member.

LSAF_GROUP_CONTEXT=*context*
The context in which the member user group is defined, specified as a path (such as /SAS). If LSAF_TYPE=GROUP, this argument is required. Otherwise, it is ignored.

Details

Sets the Boolean macro variable _lsafIsRoleMember_, which indicates whether a user or user group is a member of a role in the specified context. See “Boolean Macros” on page 4 for information about return values.

%LSAF_ISROLEPRIVILEGE Macro

Indicates whether a privilege is assigned to a role in the specified context.

Category: RoleService Module

Note: This macro sets the standard macro variables and _lsafIsRolePrivilege_, which indicates whether the privilege is assigned to the role in the specified context.

Syntax

%LSAF_ISROLEPRIVILEGE(LSAF_PATH=*path*, LSAF_ROLE=*name*, LSAF_PRIVILEGE=*privilege*);

Required Arguments

LSAF_PATH=*path*
The path of the context in which the role is assigned.

LSAF_ROLE=*name*
The name of the role.

LSAF_PRIVILEGE=*privilege*
The identifier of the privilege. To get the identifier, call %LSAF_GETSCOPEDPRIVILEGES.

Details

Sets the Boolean macro variable _lsafIsRolePrivilege_, which indicates whether a privilege is assigned to a role in the specified context. Applies only for a defined role, not an inherited role, in the specified context. See “Boolean Macros” on page 4 for information about return values.

%LSAF_REMOVEINHERITEDROLE Macro

Removes a role from the specified context.

Category: RoleService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_REMOVEINHERITEDROLE(LSAF_PATH=*path*, LSAF_ROLE=*name*, LSAF_ROLE_CONTEXT=*context*);

Required Arguments

LSAF_PATH=*path*
The path of the context from which to remove the role.

LSAF_ROLE=*name*
The name of the role.

LSAF_ROLE_CONTEXT=*context*
The context in which the role is defined, specified as a path (such as /SAS).

%LSAF_REMOVEROLEMEMBER Macro

Removes a user or a user group as a member of a role in the specified context.

Category: RoleService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_REMOVEROLEMEMBER(LSAF_PATH=*path*, LSAF_ROLE=*name*, LSAF_MEMBER=*member*<, LSAF_ROLE_CONTEXT=*context*><, LSAF_TYPE=USER | GROUP><, LSAF_GROUP_CONTEXT=*context*>);

Required Arguments

LSAF_PATH=*path*
The path of the context that contains the role.

LSAF_ROLE=*name*
The name of the role.

LSAF_MEMBER=*member*
The identifier of the user or the name of the user group.

Optional Arguments

LSAF_ROLE_CONTEXT=*context*
The context in which the role is defined, specified as a path (such as /SAS). If not specified, the member is removed from the role that is defined at the context that is specified by LSAF_PATH.

LSAF_TYPE=USER (Default) | GROUP
The type of member to remove.

LSAF_GROUP_CONTEXT=*context*
The context in which the user group to remove is defined, specified as a path (such as /SAS). If LSAF_TYPE=GROUP, this argument is required. Otherwise, it is ignored.

%LSAF_REMOVEROLEPRIVILEGE Macro

Removes a privilege from a role in the specified context.

Category: RoleService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_REMOVEROLEPRIVILEGE(LSAF_PATH=*path*, LSAF_ROLE=*name*, LSAF_PRIVILEGE=*privilege*);

Required Arguments

LSAF_PATH=*path*
The path of the context in which the role is defined.

LSAF_ROLE=*name*
The name of the role.

LSAF_PRIVILEGE=*privilege*
The name of the privilege. To get the name, call %LSAF_GETSCOPEDPRIVILEGES.

%LSAF_RENAMEROLE Macro

Renames a defined role.

Category: RoleService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_RENAMEROLE(LSAF_PATH=*path*, LSAF_ROLE=*name*, LSAF_NEWNAME=*name*);

Required Arguments

LSAF_PATH=*path*
The path in which the role is defined.

LSAF_ROLE=*name*
The name of the role.

LSAF_NEWNAME=*name*
The new name of the role. Naming conventions for roles are enforced, as is case sensitivity.

%LSAF_ROLEEXISTS Macro

Indicates whether a role exists as defined or inherited in a specified context.

Category: RoleService Module

Note: This macro sets the standard macro variables and _lsafRoleExists_, which indicates the existence of the role.

Syntax

%LSAF_ROLEEXISTS(LSAF_PATH=*path*, LSAF_ROLE=*name*<, LSAF_ROLE_CONTEXT=*context*>);

Required Arguments

LSAF_PATH=*path*
The path of the context.

LSAF_ROLE=*name*
The name of the role.

Optional Argument

LSAF_ROLE_CONTEXT=*context*
The context in which the role is defined, specified as a path (such as /SAS). If not specified or if the value is equal to the path parameter, the macro looks for the role that is defined at the path context that is specified by LSAF_PATH. Otherwise, the macro searches for the role inherited at the path context.

Details

Sets the Boolean macro variable _lsafRoleExists_, which indicates whether a role exists as defined or inherited in a specified context. See “Boolean Macros” on page 4 for information about return values.

%LSAF_UPDATEROLEDESCRIPTION Macro

Updates the description of a defined role.

Category: RoleService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_UPDATEROLEDESCRIPTION(LSAF_PATH=*path*, LSAF_ROLE=*name*, LSAF_DESCRIPTION=*description*);

Required Arguments

LSAF_PATH=*path*
The path in which the role is defined.

LSAF_ROLE=*name*
The name of the role.

LSAF_DESCRIPTION=*description*
The new description for the role.

34

ScopedPrivilegeService Module

Overview

Macros to manage the privileges that are available at the context membership level.

Macro Summary Table

Category	Language Elements	Description
ScopedPrivilege Service Module	%LSAF_GETSCOPEDPRIVILEGES Macro (p. 410)	Gets the metadata for all of the scoped privileges. The metadata is stored in a SAS data set.
	%LSAF_ISSCOPEDPRIVILEGE Macro (p. 410)	Indicates whether the specified value is a valid scoped privilege.

Dictionary

%LSAF_GETSCOPEDPRIVILEGES Macro

Gets the metadata for all of the scoped privileges. The metadata is stored in a SAS data set.

Category: ScopedPrivilegeService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETSCOPEDPRIVILEGES(< SAS_DSNAME=*name*>);

Optional Argument

SAS_DSNAME=*name*

The name of the output data set that contains the metadata for the scoped privileges. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETSCOPEDPRIVILEGES.

The data set contains a row for each privilege, sorted by privilegeId, and the following columns:

Column Name	Description
privilegeType	The privilege type: SCOPED.
privilegeId	The identifier of the privilege.
privilegeName	The name of the privilege.

%LSAF_ISSCOPEDPRIVILEGE Macro

Indicates whether the specified value is a valid scoped privilege.

Category: ScopedPrivilegeService Module

Note: This macro sets the standard macro variables and _lsafIsScopedPrivilege_, which indicates whether the item is a valid scoped privilege.

Syntax

%LSAF_ISSCOPEDPRIVILEGE(LSAF_PRIVILEGE=*privilege-id*);

Required Argument

LSAF_PRIVILEGE=*privilege-id*
The scoped privilege identifier. To get the identifier, call %LSAF_GETSCOPEDPRIVILEGES.

Details

Sets the Boolean macro variable _lsafIsScopedPrivilege_, which indicates whether the specified value is a valid scoped privilege. See “Boolean Macros” on page 4 for information about return values.

35

UserService Module

Overview

Macros to manage and create user accounts.

Macro Summary Table

Category	Language Elements	Description
UserService Module	%LSAF_CREATEUSER Macro (p. 415)	Creates a user account.
	%LSAF_DELETEUSER Macro (p. 416)	Deletes a user account.
	%LSAF_ENCRYPTPASSWORD Macro (p. 416)	Encrypts the password for a user account.
	%LSAF_GETALLUSERS Macro (p. 417)	Gets the metadata for all of the user accounts. The metadata is stored in a SAS data set.
	%LSAF_GETUSERAUTHPROVIDERS Macro (p. 420)	Gets all of the user authentication providers available to the system. The providers are stored in a SAS data set.
	%LSAF_GETUSERLICENSETYPES Macro (p. 420)	Gets all of the user license types. The types are stored in a SAS data set.
	%LSAF_GETUSERPROPERTIES Macro (p. 421)	Gets all of the properties for the specified user account. The properties are stored in a SAS data set.
	%LSAF_ISUSERLOCKED Macro (p. 422)	Indicates whether a user account is locked.
	%LSAF_LOCKUSER Macro (p. 423)	Locks a user account.
	%LSAF_RESETPASSWORD Macro (p. 423)	Resets the password for a user account.
	%LSAF_SETUSERDEFAULTAUTHENTICATOR Macro (p. 424)	Sets the default authenticator for a user account.
	%LSAF_SETUSEREXTAUTHENTICATOR Macro (p. 424)	Sets the external authenticator for the user account.
	%LSAF_UNLOCKUSER Macro (p. 425)	Unlocks a user account.
	%LSAF_UPDATEUSERPROPERTIES Macro (p. 426)	Updates the editable properties of a user account using a SAS data set as input.
	%LSAF_USEREXISTS Macro (p. 427)	Indicates whether a user account exists.

Dictionary

%LSAF_CREATEUSER Macro

Creates a user account.

Category: UserService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_CREATEUSER(LSAF_USERID=*user-id*, LSAF_DISPLAYNAME=*name*, LSAF_EMAIL=*email*, LSAF_ACCOUNTTYPE=*account-type*, LSAF_PASSWORD=*password*<, LSAF_FIRSTNAME=*name*><, LSAF_LASTNAME=*name*><, LSAF_DEPARTMENT=*department*><, LSAF_PHONE=*phone*>);

Required Arguments

LSAF_USERID=*user-id*
The user identifier of the account.

LSAF_DISPLAYNAME=*name*
The display name of the account.

LSAF_EMAIL=*email*
The email address of the account.

LSAF_ACCOUNTTYPE=*account-type*
The account type. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=AccountType`.

LSAF_PASSWORD=*password*
The password for the account.

Optional Arguments

LSAF_FIRSTNAME=*name*
The first name of the user for the account.

LSAF_LASTNAME=*name*
The last name of the user for the account.

LSAF_DEPARTMENT=*department*
The department name for the account.

LSAF_PHONE=*phone*
The phone number of the user for the account.

%LSAF_DELETEUSER Macro

Deletes a user account.

Category: UserService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_DELETEUSER(LSAF_USERID=*user-id*);

Required Argument

LSAF_USERID=*user-id*
The user identifier of the account.

Details

Deletes a user account.

Users who have logged in are not eligible for deletion and can only be deactivated.

%LSAF_ENCRYPTPASSWORD Macro

Encrypts the password for a user account.

Category: UserService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_ENCRYPTPASSWORD(LSAF_PASSWORD=*password*);

Required Argument

LSAF_PASSWORD=*password*
The password.

Details

Encrypts the password for a user account.

The returned encrypted password can be used as a valid password to log in to SAS Life Science Analytics Framework. The encrypted password is invalid for the user interface log in to SAS Life Science Analytics Framework. The value of the encrypted password is written to the SAS log file.

%LSAF_GETALLUSERS Macro

Gets the metadata for all of the user accounts. The metadata is stored in a SAS data set.

Category: UserService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETALLUSERS(<SAS_DSNAME=*name*>);

Optional Argument

SAS_DSNAME=*name*

The name of the output data set that contains the metadata for the users. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETALLUSERS.

The data set contains a row for each user, sorted by the identifier, and the following columns:

Column Name	Description
userId	The unique identifier for the account.
displayName	The display name for the account.
firstName	The first name of the user.
lastName	The last name of the user.
email	The email address for the user.
department	The department for the user.
phone	The phone number for the user.
isAccountActive	Indicates whether the account is active.

Column Name	Description
accountLicenseType	The account license type. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=LicenseType`.
isAccountLocked	Indicates whether the account is locked.
accountActivationDate	The date on which the account becomes active, represented in Character format.
accountDeactivationDate	The date on which the account becomes inactive, represented in Character format.
accountLockedBy	The identifier of the user who locked the account.
accountLockedDate	The date on which the account was locked, represented in Character format.
accountType	The type of account. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=AccountType`.
lastLogin	The date on which the user last logged in, represented in Character format.
loginAttempts	The current number of failed login attempts that are associated with the account. If the number exceeds the configured threshold, this account is locked. The counter is reset after a successful login.
passwordCreated	The date on which the current password was created, represented in Character format.
passwordExpiration	The date on which the current password expires, represented in Character format.
isPasswordExpired	Indicates whether the password has expired.
dateAccountActivation	The date on which the account becomes active, represented in SAS DateTime format.

Column Name	Description
dateAccountDeactivation	The date on which the account becomes inactive, represented in SAS DateTime format.
dateAccountLocked	The date on which the account was locked, represented in SAS DateTime format.
dateLastLogin	The date on which the user last logged in, represented in SAS DateTime format.
datePasswordCreated	The date on which the current password was created, represented in SAS DateTime format.
datePasswordExpiration	The date on which the current password expires, represented in SAS DateTime format.
createdDate	The date on which is account was created.
dateCreated	The date on which is account was created, represented in SAS DateTime format.
createdBy	The identifier of the user who created the account.
lastModifiedDate	The date on which the account was last modified.
dateLastModified	The date on which the account was last modified, represented in SAS DateTime format.
lastModifiedBy	The identifier of the user who last modified the account.

Note: See "Format Date Values" on page 4 for information about date formats.

%LSAF_GETUSERAUTHPROVIDERS Macro

Gets all of the user authentication providers available to the system. The providers are stored in a SAS data set.

Category: UserService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETUSERAUTHPROVIDERS(<SAS_DSNAME=*name*>);

Optional Argument

SAS_DSNAME=*name*

The name of the output data set that contains the authentication providers. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETUSERAUTHPROVIDERS.

The data set is sorted by name and contains the following columns:

Column Name	Description
providerId	The unique identifier of the authentication provider.
name	The name of the authentication provider.
description	The description of the authentication provider.
isPrimary	Indicates whether this is the primary authentication provider.

%LSAF_GETUSERLICENSETYPES Macro

Gets all of the user license types. The types are stored in a SAS data set.

Category: UserService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETUSERLICENSETYPES(<SAS_DSNAME=*name*>);

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the user license types. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETLICENSETYPES.

The data set is sorted by name and contains the following columns:

Column Name	Description
licenseId	The unique identifier of the license.
name	The name of the license.
description	The description of the license.

%LSAF_GETUSERPROPERTIES Macro

Gets all of the properties for the specified user account. The properties are stored in a SAS data set.

Category: UserService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETUSERPROPERTIES(LSAF_USERID=*user-id*<, SAS_DSNAME=*name*>);

Required Argument

LSAF_USERID=*user-id*
The user identifier of the account.

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the properties for the user account. See "Data Set Macros" on page 5 for information about specifying the

name of the data set. The default value is WORK.LSAFGETUSERPROPERTIES.

The data set is sorted by name and contains the following columns:

Column Name	Description
name	The name of the property.
value	The value of the property.
displayName	The name of the property as displayed as a label within SAS Life Science Analytics Framework.
type	The type of the property. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=AttributeType`.
isReadonly	Indicates whether the property cannot be modified.
isExtended	Indicates whether the property is an extended attribute.
isRequired	Indicates whether the property is required.

%LSAF_ISUSERLOCKED Macro

Indicates whether a user account is locked.

Category: UserService Module

Note: This macro sets the standard macro variables and _lsafIsUserLocked_, which indicates whether the user account is locked.

Syntax

%LSAF_ISUSERLOCKED(LSAF_USERID=*user-id*);

Required Argument

LSAF_USERID=*user-id*
The user identifier of the account.

Details

Sets the Boolean macro variable _lsafIsUserLocked_, which indicates whether a user account is locked. See “Boolean Macros” on page 4 for information about return values.

%LSAF_LOCKUSER Macro

Locks a user account.

Category: UserService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_LOCKUSER(LSAF_USERID=*user-id*);

Required Argument

LSAF_USERID=*user-id*
The user identifier of the account.

Details

Locks a user account.

You must have the user account privilege Manage User Accounts.

%LSAF_RESETPASSWORD Macro

Resets the password for a user account.

Category: UserService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_RESETPASSWORD(LSAF_USERID=*user-id*, LSAF_NEWPASSWORD=*password*);

Required Arguments

LSAF_USERID=*user-id*
The user identifier of the account.

LSAF_NEWPASSWORD=*password*
The new, temporary password.

Details

Resets the password for a user account.

You must have the user account privilege Manage User Accounts. After the password has been successfully reset, you can immediately log on with the new password. But, you are required to immediately change the password.

%LSAF_SETUSERDEFAULTAUTHENTICATOR Macro

Sets the default authenticator for a user account.

Category: UserService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_SETUSERDEFAULTAUTHENTICATOR(LSAF_USERID=*user-id*, LSAF_PASSWORD=*password*);

Required Arguments

LSAF_USERID=*user-id*
The user identifier of the account.

LSAF_PASSWORD=*password*
The unique password for the user account that overwrites the existing password. Valid password rules are enforced for this argument.

%LSAF_SETUSEREXTAUTHENTICATOR Macro

Sets the external authenticator for the user account.

Category: UserService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_SETUSEREXTAUTHENTICATOR(LSAF_USERID=*user-id*, LSAF_AUTHENTICATORID=*authenticator-id*<, LSAF_AUTHENTICATORNAME=*name*>);

Required Arguments

LSAF_USERID=*user-id*
The user identifier of the account.

LSAF_AUTHENTICATORID=*authenticator-id*
The unique identifier of the external authenticator. To get the identifier, call %LSAF_GETUSERAUTHPROVIDERS.

Optional Argument

LSAF_AUTHENTICATORNAME=*name*
The user name known to the external authenticator.

Details

Sets the external authenticator for the user account.

To change a user account back to a default state, call %LSAF_SETUSERDEFAULTAUTHENTICATOR.

%LSAF_UNLOCKUSER Macro

Unlocks a user account.

Category: UserService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_UNLOCKUSER(LSAF_USERID=*user-id*);

Required Argument

LSAF_USERID=*user-id*
The user identifier of the account.

Details

Unlocks a user account.

You must have the user account privilege Manage User Accounts.

%LSAF_UPDATEUSERPROPERTIES Macro

Updates the editable properties of a user account using a SAS data set as input.

Category: UserService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_UPDATEUSERPROPERTIES(LSAF_USERID=*user-id*, SAS_DSNAME=*name*);

Required Arguments

LSAF_USERID=*user-id*
The user identifier of the account.

SAS_DSNAME=*name*
The name of the data set that contains the property values to set. See “Data Set Macros” on page 5 for information about specifying the name of the data set.

The data set must contain at least the following columns. Additional variables are ignored.

Column Name	Description
name	The name of the property.
value	The value of the property.

Details

Updates the editable properties of a user account using a SAS data set as input.

The output data set from the %LSAF_GETUSERPROPERTIES macro. contains the necessary variables and data needed as the input data set for the %LSAF_UPDATEUSERPROPERTIES macro. The column isReadonly indicates whether the property can be updated: 0 allows updates, 1 does not. Modify the data set with changes prior to calling %LSAF_UPDATEUSERPROPERTIES.

- Data in the input data set that do not correspond to a property of the user account to update are ignored.
- Values that are specified for properties that are not editable are ignored.
- An input data value that is the same as the current property value is ignored and does not update the property. No audit record is created for the property.

%LSAF_USEREXISTS Macro

Indicates whether a user account exists.

Category: UserService Module

Note: This macro sets the standard macro variables and _lsafUserExists_, which indicates whether the user account exists.

Syntax

%LSAF_USEREXISTS(LSAF_USERID=*user-id*);

Required Argument

LSAF_USERID=*user-id*
The user identifier of the account.

Details

Sets the Boolean macro variable _lsafUserExists_, which indicates whether a user account exists. See “Boolean Macros” on page 4 for information about return values.

PART 9

Workflow Macros

36

AutoflowService Module

Overview

Macros to create autoflow process flows.

Macro Summary Table

Category	Language Elements	Description
AutoflowService Module	%LSAF_CREATEAUTOFLOW Macro (p. 432)	Automatically creates a process flow based on a process definition whose elements have been mapped to source (such as TLF) metadata.

Dictionary

%LSAF_CREATEAUTOFLOW Macro

Automatically creates a process flow based on a process definition whose elements have been mapped to source (such as TLF) metadata.

Category: AutoflowService Module

Note: This macro sets the standard macro variables and _lsafcreateautoflow_, which specifies the name of the process flow that is created by the autoflow operation.

Syntax

%LSAF_CREATEAUTOFLOW(LSAF_CONTEXTID=*context-id*, LSAF_PROCESSDEFID=*definition-id*, LSAF_NAMEPREFIX=*prefix*, LSAF_SOURCEID=*source-id*<, LSAF_AUTOFLOWTYPE=*type*>);

Required Arguments

LSAF_CONTEXTID=*context-id*
The identifier of the study context in which to create the process flow. To get the identifier, call %LSAF_GETSTUDYID.

LSAF_PROCESSDEFID=*definition-id*
The case-sensitive identifier of the process flow definition. To get the identifier, call %LSAF_GETALLPROCESSDEFINITIONS.

LSAF_NAMEPREFIX=*prefix*
The prefix to add to the name (name of the source) of the process flow to create.

LSAF_SOURCEID=*source-id*
The alphanumeric identifier of the source from which to create the process flow, such as Study TLF. To get the identifier, call %LSAF_GETSTUDYTLFSINFO.

Optional Argument

LSAF_AUTOFLOWTYPE=*type*
The autoflow type. Valid value: TLF.

Details

Sets the value of the macro variable _lsafcreateautoflow_ to the name of the process flow that is created by the autoflow operation. See “Value Macros” on page 5 for information about value macros.

37

ProcessDefinitionMappingService Module

Overview

Macros to manipulate process flow definition autoflow mapping.

The metadata for autoflow mappings for a process flow definition are configured at each site. To determine the metadata that is configured at your site, call %LSAF_GETDEFINITIONMAPPINGS.

Macro Summary Table

Category	Language Elements	Description
ProcessDefinitionMappingService Module	%LSAF_CLEARDEFELEMENTMAPPINGS Macro (p. 436)	Removes all autoflow mappings from the specified process flow definition element.
	%LSAF_CLEARDEFINTIONMAPPINGS Macro (p. 437)	Removes all autoflow mappings for the specified process flow definition.
	%LSAF_DEFAUTOFLOWENABLED Macro (p. 438)	Indicates whether autoflow has been enabled for the specified process definition.
	%LSAF_DEFMAPPINGISCOMPLETE Macro (p. 439)	Indicates whether all of the auto flow attributes that are required for a process flow to be successfully created and activated are mapped for the specified process definition.
	%LSAF_GETDEFINITIONMAPPINGS Macro (p. 439)	Gets all of the available autoflow mappings and their associated metadata for the specified process flow definition or the process flow definition element. The metadata is stored in a SAS data set.
	%LSAF_GETDEFSMAPPEDFORAUTOFLOW Macro (p. 441)	Gets the metadata for the process flow definitions that are enabled and mapped for automatic process flow creation. The metadata is stored in a SAS data set.
	%LSAF_SETDEFAUTOFLOWENABLED Macro (p. 443)	Sets the autoflow flag for the specified process definition.
	%LSAF_UPDATEDEFINITIONMAPPINGS Macro (p. 443)	Updates the autoflow mappings for the specified process definition elements using a data set as input.

Dictionary

%LSAF_CLEARDEFELEMENTMAPPINGS Macro

Removes all autoflow mappings from the specified process flow definition element.

Category: ProcessDefinitionMappingService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_CLEARDEFELEMENTMAPPINGS(LSAF_PROCESSDEFID=*definition-id*, LSAF_ELEMENTID=*element-id*<, LSAF_AUTOFLOWTYPE=*autoflow-type*>);

Required Arguments

LSAF_PROCESSDEFID=*definition-id*
The case-sensitive identifier of the process flow definition. To get the identifier, call %LSAF_GETALLPROCESSDEFINITIONS.

LSAF_ELEMENTID=*element-id*
The unique identifier of the process flow definition element. To get the identifier, call %LSAF_GETPROCESSDEFELEMENTS.

Optional Argument

LSAF_AUTOFLOWTYPE=*autoflow-type*
The autoflow type. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=AutoFlowType`.

Details

Removes all autoflow mappings from the specified process flow definition element.

You must have the global privilege PRIVILEGE_MANAGE_PROCESS_FLOW_DEFINITIONS.

%LSAF_CLEARDEFINTIONMAPPINGS Macro

Removes all autoflow mappings for the specified process flow definition.

Category: ProcessDefinitionMappingService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_CLEARDEFINTIONMAPPINGS(LSAF_PROCESSDEFID=*definition-id*<, LSAF_AUTOFLOWTYPE=*autoflow-type*>);

Required Argument

LSAF_PROCESSDEFID=*definition-id*
The case-sensitive identifier of the process flow definition. To get the identifier, call %LSAF_GETALLPROCESSDEFINITIONS.

Optional Argument

LSAF_AUTOFLOWTYPE=*autoflow-type*
The autoflow type. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=AutoFlowType`.

Details

Removes all autoflow mappings for the specified process flow definition.

You must have the global privilege PRIVILEGE_MANAGE_PROCESS_FLOW_DEFINITIONS.

%LSAF_DEFAUTOFLOWENABLED Macro

Indicates whether autoflow has been enabled for the specified process definition.

Category: ProcessDefinitionMappingService Module

Note: This macro sets the standard macro variables and _lsafdefautoflowenabled_, which indicates whether the autoflow is enabled.

Syntax

%LSAF_DEFAUTOFLOWENABLED(LSAF_PROCESSDEFID=*definition-id*<, LSAF_AUTOFLOWTYPE=*autoflow-type*>);

Required Argument

LSAF_PROCESSDEFID=*definition-id*
The case-sensitive identifier of the process flow definition. To get the identifier, call %LSAF_GETALLPROCESSDEFINITIONS.

Optional Argument

LSAF_AUTOFLOWTYPE=*autoflow-type*
The autoflow type. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=AutoFlowType`.

Details

Sets the Boolean macro variable _lsafdefautoflowenabled_, which indicates whether autoflow is enabled for the specified process definition. See “Boolean Macros” on page 4 for information about return values.

%LSAF_DEFMAPPINGISCOMPLETE Macro

Indicates whether all of the auto flow attributes that are required for a process flow to be successfully created and activated are mapped for the specified process definition.

Category: ProcessDefinitionMappingService Module

Note: This macro sets the standard macro variables and _lsafdefmappingiscomplete_, which indicates whether the autoflow mapping is complete.

Syntax

%LSAF_DEFMAPPINGISCOMPLETE(LSAF_PROCESSDEFID=*definition-id*<, LSAF_AUTOFLOWTYPE=*autoflow-type*>);

Required Argument

LSAF_PROCESSDEFID=*definition-id*
The case-sensitive identifier of the process flow definition. To get the identifier, call %LSAF_GETALLPROCESSDEFINITIONS.

Optional Argument

LSAF_AUTOFLOWTYPE=*autoflow-type*
The autoflow type. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=AutoFlowType`.

Details

Sets the Boolean macro variable _lsafdefmappingsiscomplete_, which indicates whether all of the auto flow attributes that are required for a process flow to be successfully created and activated are mapped for the specified process definition. See “Boolean Macros” on page 4 for information about return values.

Mappings for timer or signal elements are required.

%LSAF_GETDEFINITIONMAPPINGS Macro

Gets all of the available autoflow mappings and their associated metadata for the specified process flow definition or the process flow definition element. The metadata is stored in a SAS data set.

Category: ProcessDefinitionMappingService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETDEFINITIONMAPPINGS(LSAF_PROCESSDEFID=*definition-id*<, LSAF_AUTOFLOWTYPE=*autoflow-type*><, LSAF_ELEMENTID=*element-id*><, SAS_DSNAME=*name*>);

Required Argument

LSAF_PROCESSDEFID=*definition-id*
The case-sensitive identifier of the process flow definition. To get the identifier, call %LSAF_GETALLPROCESSDEFINITIONS.

Optional Arguments

LSAF_AUTOFLOWTYPE=*autoflow-type*
The autoflow type. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=AutoFlowType`.

LSAF_ELEMENTID=*element-id*
The unique identifier for the process definition element. To get the identifier, call %LSAF_GETPROCESSDEFELEMENTS. If omitted, all mappings for all elements for the specified process definition are returned.

SAS_DSNAME=*name*
The name of the output data set that contains the autoflow mapping metadata. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETDEFINITIONMAPPINGS.

The data set contains a row for each mapping and columns with the following names:

Column Name	Description
defID	The identifier of the process flow definition.
elementId	The identifier of the process definition element.
elementAttrName	The name of the element attribute that can be mapped, such as Location or Recipients.
elementAttrDisplayName	The display name of the element attribute.
elementAttrType	The type of the element attribute, such as TEXT, USER, or FILE. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=FlowAttributeType`.

Column Name	Description
elementAttrIsMultivalued	Indicates whether the attribute allows more than one mapping.
mappingName	The name of the mapping to associate with the element attribute.
mappingType	The type of the mapping, such as TEXT, USER, or FILE. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=FlowAttributeType`.
mappingIsDerived	Indicates whether the mapping is derived. Derived mappings cannot be updated.
isMapped	Indicates whether the mapping is set for the element attribute.

Details

Gets all of the available autoflow mappings and their associated metadata for the specified process flow definition or the process flow definition element. The metadata is stored in a SAS data set.

You must have the global privilege PRIVILEGE_MANAGE_PROCESS_FLOW_DEFINITIONS.

%LSAF_GETDEFSMAPPEDFORAUTOFLOW Macro

Gets the metadata for the process flow definitions that are enabled and mapped for automatic process flow creation. The metadata is stored in a SAS data set.

Category: ProcessDefinitionMappingService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETDEFSMAPPEDFORAUTOFLOW(<LSAF_CONTEXTTYPEID=*type-id*><, LSAF_AUTOFLOWTYPE=*autoflow-type*><, LSAF_ACTIVEONLY=0 | 1><, SAS_DSNAME=*name*>);

Optional Arguments

LSAF_CONTEXTTYPEID=*type-id*
The context type identifier in which the process definition is deployed. To get the identifier, call %LSAF_GETCONTEXTTYPES.

LSAF_AUTOFLOWTYPE=*autoflow-type*
The autoflow type. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=AutoFlowType`.

LSAF_ACTIVEONLY=0 (Default) | 1
Indicates whether to include only process definitions with a status of ACTIVE.

SAS_DSNAME=*name*
The name of the output data set that contains the list of process flow definitions. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETDEFSMAPPEDFORAUTOFLOW.

The data set contains a row for each process flow definition and columns with the following names:

Column Name	Description
defID	The identifier of the process flow definition.
name	The name of the process flow definition.
description	The process flow definition description.
deployedVersion	The version of the process flow definition.
sourceFilePath	The path of the source file from which the process flow definition was deployed.
sourceFileVersion	The version of the source file from which the process flow definition was deployed.
contexts	A comma-delimited string that lists the context types for which the process flow definition is deployed.
status	The status of the process flow definition.
comment	The comment for audit trail purposes when the process definition was deployed.

%LSAF_SETDEFAUTOFLOWENABLED Macro

Sets the autoflow flag for the specified process definition.

Category: ProcessDefinitionMappingService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_SETDEFAUTOFLOWENABLED(LSAF_PROCESSDEFID=*process-id*<, LSAF_AUTOFLOWTYPE=*autoflow-type*><, LSAF_ENABLEAUTOFLOW=0 | 1 (Default)>);

Required Argument

LSAF_PROCESSDEFID=*process-id*
The case-sensitive identifier of the process flow definition. To get the identifier, call %LSAF_GETALLPROCESSDEFINITIONS.

Optional Arguments

LSAF_AUTOFLOWTYPE=*autoflow-type*
The autoflow type. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=AutoFlowType`.

LSAF_ENABLEAUTOFLOW=0 | 1 (Default)
Indicates whether to enable autoflow.

Details

Sets the autoflow flag for the specified process definition.

You must have the global privilege PRIVILEGE_MANAGE_PROCESS_FLOW_DEFINITIONS.

%LSAF_UPDATEDEFINITIONMAPPINGS Macro

Updates the autoflow mappings for the specified process definition elements using a data set as input.

Category: ProcessDefinitionMappingService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_UPDATEDEFINITIONMAPPINGS(LSAF_PROCESSDEFID=*definition-id*, SAS_DSNAME=*name*<, LSAF_AUTOFLOWTYPE=*autoflow-type*>);

Required Arguments

LSAF_PROCESSDEFID=*definition-id*
The case-sensitive identifier of the process flow definition. To get the identifier, call %LSAF_GETALLPROCESSDEFINITIONS.

SAS_DSNAME=*name*
The name of the input data set that contains the details for the process definition mappings. See "Data Set Macros" on page 5 for information about specifying the name of the data set.

The following columns are required for processing, all other columns are ignored:

Column Name	Description
elementId	The identifier of the process definition element.
elementAttrName	The name of the element attribute to be mapped, such as Location or Recipients.
mappingName	The name of the mapping to be associated with the element attribute.

Optional Argument

LSAF_AUTOFLOWTYPE=*autoflow-type*
The autoflow type. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=AutoFlowType`.

Details

Updates the autoflow mappings for the specified process definition elements, using a data set as input.

The operation maps only the data that is included in the data set. Existing mappings for the elements that are included in the data set are overwritten. In the event of macro failure, no updates are made.

To get the values that are needed for the input data set, call %LSAF_GETDEFINITIONMAPPINGS.

You must have the global privilege PRIVILEGE_MANAGE_PROCESS_FLOW_DEFINITIONS.

38

ProcessDefinitionService Module

Overview

Macros to manage process flow definitions.

Macro Summary Table

Category	Language Elements	Description
ProcessDefinition Service Module	%LSAF_ACTIVATEPROCESSDEFINITION Macro (p. 446)	Activates a process flow definition.

Category	Language Elements	Description
	%LSAF_DEPLOYPROCESSDEFINITION Macro (p. 447)	Deploys a process flow definition using a BPMN file from the repository.
	%LSAF_GETALLPROCESSDEFINITIONS Macro (p. 448)	Gets the metadata for all of the process flow definitions that are available to the user. The metadata is stored in a SAS data set.
	%LSAF_GETPROCESSDEFCONTEXTS Macro (p. 449)	Gets the context types in which the specified process flow definition is deployed. The metadata is stored in a SAS data set.
	%LSAF_GETPROCESSDEFELEMENTS Macro (p. 450)	Gets the elements of the specified process flow definition.
	%LSAF_GETPROCESSDEFINITIONSTATUS Macro (p. 451)	Gets the status of a process flow definition.
	%LSAF_GETPROCESSDEFSBYTYPE Macro (p. 452)	Gets the metadata for all of the process flow definitions that are available at the specified context type. The metadata is stored in a SAS data set.
	%LSAF_PROCESSDEFDEPLOYEDATCONTEXT Macro (p. 453)	Indicates whether a process flow definition is deployed at the specified context type.
	%LSAF_SUSPENDPROCESSDEFINITION Macro (p. 454)	Suspends a process flow definition.

Dictionary

%LSAF_ACTIVATEPROCESSDEFINITION Macro

Activates a process flow definition.

Category: ProcessDefinitionService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_ACTIVATEPROCESSDEFINITION(LSAF_PROCESSDEFID=*definition-id*);

Required Argument

LSAF_PROCESSDEFID=*definition-id*
The case-sensitive identifier of the process flow definition. To get the identifier, call %LSAF_GETALLPROCESSDEFINITIONS.

Details

Activates a process flow definition. You must have the PRIVILEGE_MANAGE_PROCESS_FLOW_DEFINITIONS global privilege.

%LSAF_DEPLOYPROCESSDEFINITION Macro

Deploys a process flow definition using a BPMN file from the repository.

Category: ProcessDefinitionService Module

Note: This macro sets the standard macro variables and _lsafProcessDefinitionId_ , which is the identifier of the deployed process flow definition.

Syntax

%LSAF_DEPLOYPROCESSDEFINITION(LSAF_PATH=*path*, SAS_DSNAME=*name*<, LSAF_VERSION=*version*><, LSAF_COMMENT=*comment*><, LSAF_ACTIVATE=0 | 1><, LSAF_OVERWRITE=0 | 1>);

Required Arguments

LSAF_PATH=*path*
The case-sensitive path to the process BPMN file.

SAS_DSNAME=*name*
The name of the output data set that contains the context type identifiers in which to deploy the process flow definition. See “Data Set Macros” on page 5 for information about specifying the name of the data set.

The data set must contain at least the following columns. Additional variables are ignored.

Column Name	Description
id	The context type identifier. To get the identifier, call %LSAF_GETCONTEXTTYPES.

Optional Arguments

LSAF_VERSION=*version*
The version of the BPMN file. See “LSAF_VERSION=version” on page 6 for more information about this argument.

LSAF_COMMENT=*comment*
The audit trail comment for the deployment.

LSAF_ACTIVATE=0 | 1 (Default)
Indicates whether to activate the process flow definition.

LSAF_OVERWRITE=0 (Default) | 1
Indicates whether an existing process flow definition with same identifier is overwritten. See “LSAF_OVERWRITE=0 (Default) | 1” on page 6 for more information about this argument.

Details

Deploys a process flow definition using a BPMN file from the repository, and sets the value of the macro variable _lsafProcessDefinitionId_ to the process flow definition identifier that is set as part of the deployment. You must have the global privilege PRIVILEGE_MANAGE_PROCESS_FLOW_DEFINITIONS. See “Value Macros” on page 5 for information about value macros.

%LSAF_GETALLPROCESSDEFINITIONS Macro

Gets the metadata for all of the process flow definitions that are available to the user. The metadata is stored in a SAS data set.

Category: ProcessDefinitionService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETALLPROCESSDEFINITIONS(<SAS_DSNAME=*name*>);

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the process flow definitions. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETALLPROCESSDEFINITIONS.

The data set contains a row for each process flow definition deployed, sorted by the process flow definition identifier, and the following columns:

Column Name	Description
defID	The identifier of the process flow definition.
name	The name of the process flow definition.
description	The process flow definition description.
deployedVersion	The version of the process flow definition.
sourceFilePath	The path of the source file from which the process flow definition was deployed.
sourceFileVersion	The version of the source file from which the process flow definition was deployed.
contexts	A comma-delimited string that lists the context types for which the process flow definition is deployed.
status	The status of the process flow definition.
comment	Comment for audit trail purposes when the process flow definition was deployed.

%LSAF_GETPROCESSDEFCONTEXTS Macro

Gets the context types in which the specified process flow definition is deployed. The metadata is stored in a SAS data set.

Category: ProcessDefinitionService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETPROCESSDEFCONTEXTS(LSAF_PROCESSDEFID=*definition-id*<, SAS_DSNAME=*name*>);

Required Argument

LSAF_PROCESSDEFID=*definition-id*
The case-sensitive identifier of the process flow definition. To get the identifier, call %LSAF_GETALLPROCESSDEFINITIONS.

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the current context types for the process flow definition. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETPROCESSDEFCONTEXTS.

The data set contains a row for each context type, sorted by processDefContextType, and the following columns:

Column Name	Description
processDefID	The unique identifier of the process flow definition.
processDefName	The name of the process flow definition.
processDefDeployedVersion	The deployed version of the process flow definition.
processDefContextType	The context type of the process flow definition.

%LSAF_GETPROCESSDEFELEMENTS Macro

Gets the elements of the specified process flow definition.

Category: ProcessDefinitionService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETPROCESSDEFELEMENTS(LSAF_PROCESSDEFID=*definition-id*<, SAS_DSNAME=*name*>);

Required Argument

LSAF_PROCESSDEFID=*definition-id*
The unique identifier of the process flow definition. To get the identifier, call %LSAF_GETALLPROCESSDEFINITIONS.

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the list of process flow elements. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETPROCESSDEFELEMENTS.

The data set contains a row for each process flow element, sorted by elementId, and the following columns:

Column Name	Description
processDefID	The unique identifier of the process flow definition.
elementId	The unique identifier of the process definition element.
elementName	The name of the process definition element.
elementDescription	The description of the process definition element.
elementType	The type of the process definition element.

%LSAF_GETPROCESSDEFINITIONSTATUS Macro

Gets the status of a process flow definition.

Category: ProcessDefinitionService Module

Note: This macro sets the standard macro variables and _lsafProcessDefinitionStatus_, which is the process flow definition status.

Syntax

%LSAF_GETPROCESSDEFINITIONSTATUS(LSAF_PROCESSDEFID=*definition-id*);

Required Argument

LSAF_PROCESSDEFID=*definition-id*
The case-sensitive identifier of the process flow definition. To get the identifier, call %LSAF_GETALLPROCESSDEFINITIONS.

Details

Gets the status of a process flow definition. The Manage Process Flows privilege is required. Sets the value of the macro variable _lsafProcessDefinitionStatus_ to the status value (SUSPENDED or ACTIVE). See "Value Macros" on page 5 for information about value macros.

%LSAF_GETPROCESSDEFSBYTYPE Macro

Gets the metadata for all of the process flow definitions that are available at the specified context type. The metadata is stored in a SAS data set.

Category: ProcessDefinitionService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETPROCESSDEFSBYTYPE(LSAF_TYPEID=*type-id*<, SAS_DSNAME=*name*>);

Required Argument

LSAF_TYPEID=*type-id*
The context type identifier. To get the identifier, call %LSAF_GETCONTEXTTYPES.

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the process flow definitions. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETPROCESSDEFSBYTYPE.

The data set contains a row for each process flow definition deployed at the context type level, sorted by the process flow definition identifier, and the following columns:

Column Name	Description
defID	The identifier of the process flow definition.
name	The name of the process flow definition.
description	The process flow definition description.
deployedVersion	The version of the process flow definition.
sourceFilePath	The path of the source file from which the process flow definition was deployed.
sourceFileVersion	The version of the source file from which the process flow definition was deployed.
contexts	A comma-delimited string that lists the context types for which the process flow definition is deployed.
status	The status of the process flow definition.
comment	The comment.

%LSAF_PROCESSDEFDEPLOYEDATCONTEXT Macro

Indicates whether a process flow definition is deployed at the specified context type.

Category: ProcessDefinitionService Module

Note: This macro sets the standard macro variables and _lsafPFDefDeployedAtContext_, which indicates whether the process flow definition is deployed at the specified context.

Syntax

%LSAF_PROCESSDEFDEPLOYEDATCONTEXT(LSAF_PROCESSDEFID=*definition-id*, LSAF_TYPEID=*type-id*);

Required Arguments

LSAF_PROCESSDEFID=*definition-id*
The identifier of the process flow definition.

LSAF_TYPEID=*type-id*
The context type identifier. To get the identifier, call %LSAF_GETCONTEXTTYPES.

Details

Sets the Boolean macro variable _lsafPFDefDeployedAtContext_, which indicates whether a process flow definition is deployed for the context type at the specified context type. See “Boolean Macros” on page 4 for information about return values.

%LSAF_SUSPENDPROCESSDEFINITION Macro

Suspends a process flow definition.

Category: ProcessDefinitionService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_SUSPENDPROCESSDEFINITION(LSAF_PROCESSDEFID=*definition-id*);

Required Argument

LSAF_PROCESSDEFID=*definition-id*
The identifier of the process flow definition. To get the identifier, call %LSAF_GETALLPROCESSDEFINITIONS.

Details

Suspends a process flow definition. The PRIVILEGE_MANAGE_PROCESS_FLOW_DEFINITIONS global privilege is required.

39

ProcessFlowService Module

Overview

Macros to manage process flows.

Macro Summary Table

Category	Language Elements	Description
ProcessFlowService Module	%LSAF_ACTIVATEPROCESSFLOW Macro (p. 456)	Activates a process flow.

Category	Language Elements	Description
	%LSAF_COPYPROCESSFLOW Macro (p. 457)	Copies a process flow. The new process flow is created in the same context as the source process flow.
	%LSAF_CREATEPROCESSFLOW Macro (p. 458)	Creates a process flow in the specified context.
	%LSAF_CREATEPROCESSFLOWMANIFEST Macro (p. 459)	Creates a process flow manifest file in the repository.
	%LSAF_DELETEPROCESSFLOW Macro (p. 460)	Deletes a process flow.
	%LSAF_GETPROCESSFLOWDATA Macro (p. 461)	Gets the process data for the specified process flow within the specified context. The metadata is stored in a SAS data set.
	%LSAF_GETPROCESSFLOWPROPERTIES Macro (p. 462)	Gets the properties for the specified process flow. The properties are stored in a SAS data set.
	%LSAF_GETPROCESSFLOWSBYQUERY Macro (p. 463)	Gets the metadata for process flows that match the specified search criteria and to which you have access permission. The metadata is stored in a SAS data set.
	%LSAF_PROCESSFLOWEXISTS Macro (p. 465)	Indicates whether a process flow is defined for the specified context.
	%LSAF_SUSPENDPROCESSFLOW Macro (p. 466)	Suspends a process flow.
	%LSAF_UPDATEPROCESSFLOWDATA Macro (p. 467)	Updates the process data for the specified process flow.
	%LSAF_UPDATEPROCESSFLOWPROPERTIES Macro (p. 468)	Updates the properties for the specified process flow.

Dictionary

%LSAF_ACTIVATEPROCESSFLOW Macro

Activates a process flow.

Category: ProcessFlowService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_ACTIVATEPROCESSFLOW(LSAF_PATH=*path*,
LSAF_PROCESSFLOW=*name*);

Required Arguments

LSAF_PATH=*path*
The path of the context in which the process flow is located.

LSAF_PROCESSFLOW=*name*
The case-sensitive name of the process flow. To get the name, call %LSAF_GETPROCESSFLOWSBYQUERY.

%LSAF_COPYPROCESSFLOW Macro

Copies a process flow. The new process flow is created in the same context as the source process flow.

Category: ProcessFlowService Module

Note: This macro sets the standard macro variables and _lsafCopiedProcessFlowId_, which is the identifier for the new process flow.

Syntax

%LSAF_COPYPROCESSFLOW(LSAF_PATH=*path*,
LSAF_PROCESSFLOW=*source-name*, LSAF_NAME=*copy-name*<,
LSAF_USESOURCEOWNER=0 | 1><, LSAF_INCLUDEATTACHMENTS=0 | 1><,
LSAF_INCLUDEPROCESSDATA=0 | 1><,
LSAF_ACTIVATENEWPROCESSFLOW=0 | 1>);

Required Arguments

LSAF_PATH=*path*
The case-sensitive path of the context in which the source process flow is located.

LSAF_PROCESSFLOW=*source-name*
The case-sensitive name of the source process flow. To get the name, call %LSAF_GETPROCESSFLOWSBYQUERY.

LSAF_NAME=*copy-name*
The name of the new process flow.

Optional Arguments

LSAF_USESOURCEOWNER=0 (Default) | 1
Indicates whether the owner of the new process flow is the same as the source process flow.

LSAF_INCLUDEATTACHMENTS=0 (Default) | 1
Indicates whether to copy the attachments of the source to the new process flow.

LSAF_INCLUDEPROCESSDATA=0 (Default) | 1
Indicates whether to copy the values of the process data from the source to the new process flow.

LSAF_ACTIVATENEWPROCESSFLOW=0 (Default) | 1
Indicates whether to activate the new process flow.

Details

Sets the value of the macro variable _lsafCopiedProcessFlowId_ to the identifier for the new process flow. See “Value Macros” on page 5 for information about value macros.

%LSAF_CREATEPROCESSFLOW Macro

Creates a process flow in the specified context.

Category: ProcessFlowService Module

Note: This macro sets the standard macro variables and _lsafProcessFlowId_, which is the identifier for the newly created process flow.

Syntax

%LSAF_CREATEPROCESSFLOW(LSAF_PATH=*path*, LSAF_PROCESSFLOW=*name*, LSAF_PROCESSDEFID=*definition-id*);

Required Arguments

LSAF_PATH=*path*
The path of the context in which to create the process flow.

LSAF_PROCESSFLOW=*name*
The name of the new process flow.

LSAF_PROCESSDEFID=*definition-id*
The identifier of the process flow definition to use to create the process flow. To retrieve the process flow definition identifier, call %LSAF_GETPROCESSDEFSBYTYPE.

Details

Sets the value of the macro variable _lsafProcessFlowId_ to the identifier for the newly created process flow. See “Value Macros” on page 5 for information about value macros.

%LSAF_CREATEPROCESSFLOWMANIFEST Macro

Creates a process flow manifest file in the repository.

Category: ProcessFlowService Module

Note: This macro sets the standard macro variables and _lsafCreatedProcessFlowManifest_, which is the full path of the process flow manifest file.

Syntax

%LSAF_CREATEPROCESSFLOWMANIFEST(LSAF_PATH=*path*, LSAF_PROCESSFLOW=*name*, LSAF_MANIFEST_PATH=*path*<, LSAF_OVERWRITE=0 | 1><, LSAF_ENABLEVERSIONING=0 | 1><, LSAF_VERSIONTYPE=MINOR | MAJOR | CUSTOM><, LSAF_CUSTOMVERSION=*verision*><, LSAF_COMMENT=*comment*>);

Required Arguments

LSAF_PATH=*path*
The context path in which the process flow is located.

LSAF_PROCESSFLOW=*name*
The name of the process flow.

LSAF_MANIFEST_PATH=*path*
The repository location for the manifest file. The location must be a container and must exist in the repository.

Optional Arguments

LSAF_OVERWRITE=0 (Default) | 1
Indicates whether an existing unversioned repository file is overwritten by a new file with a same path. See “LSAF_OVERWRITE=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_ENABLEVERSIONING=0 (Default) | 1
Indicates whether a new file is versioned. See “LSAF_ENABLEVERSIONING=0 (Default) | 1” on page 6 for more information about this argument.

LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM
If the file is versioned, the version type to create. See “LSAF_VERSIONTYPE=MAJOR | MINOR | CUSTOM” on page 6 for more information about this argument.

LSAF_CUSTOMVERSION=*verision*
The version number to assign to a versioned file. See “LSAF_CUSTOMVERSION=version” on page 6 for more information about this argument.

LSAF_COMMENT=*comment*
The audit trail comment for the manifest file creation.

Details

Sets the value of the macro variable _lsafCreatedProcessFlowManifest_ to the full path of the process flow manifest file, which includes the file extension. See “Value Macros” on page 5 for information about value macros.

If the manifest file exists, a new version of the file is created. If the existing file is not versioned, it is overwritten by the new file.

%LSAF_DELETEPROCESSFLOW Macro

Deletes a process flow.

Category: ProcessFlowService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_DELETEPROCESSFLOW(LSAF_PATH=*path*, LSAF_PROCESSFLOW=*name*);

Required Arguments

LSAF_PATH=*path*
The path of the context in which the process flow is located.

LSAF_PROCESSFLOW=*name*
The name of the process flow.

%LSAF_GETPROCESSFLOWDATA Macro

Gets the process data for the specified process flow within the specified context. The metadata is stored in a SAS data set.

Category: ProcessFlowService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETPROCESSFLOWDATA(LSAF_PATH=*path*, LSAF_PROCESSFLOW=*name*<, SAS_DSNAME=*name*>);

Required Arguments

LSAF_PATH=*path*
The path of the context in which the process flow is located.

LSAF_PROCESSFLOW=*name*
The name of the process flow.

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the process flow data. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETPROCESSFLOWDATA.

The data set contains a row for each process flow data element and columns with the following names. The data set is not sorted.

Column Name	Description
processFlowPath	The path of the context in which the process flow is located.
processFlowName	The name of the process flow.
dataID	The identifier of the process flow data element.
dataName	The name of the process flow data element.
dataType	The type of the process flow data element.

Column Name	Description
dataValue	The value of the process flow data element, represented as a String.
dataEnumValues	A comma-delimited list of the defined values for the elements of the type ENUM, displayed in the format key(value).

%LSAF_GETPROCESSFLOWPROPERTIES Macro

Gets the properties for the specified process flow. The properties are stored in a SAS data set.

Category: ProcessFlowService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETPROCESSFLOWPROPERTIES(LSAF_PATH=*path*, LSAF_PROCESSFLOW=*name*<, SAS_DSNAME=*name*>);

Required Arguments

LSAF_PATH=*path*
The path of the context in which the process flow is located.

LSAF_PROCESSFLOW=*name*
The name of the process flow.

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the properties and attributes for the process flow data. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETPROCESSFLOWPROPERTIES.

The data set contains a row for each property or attribute and columns with the following names. The data set is not sorted.

Column Name	Description
processFlowPath	The path of the context in which the process flow is located.
processFlowName	The name of the process flow.
name	The name of the property.
displayName	The name of the property, displayed as a label in SAS Life Science Analytics Framework. Derived properties (such as name and owner) do not have a value for displayName.
value	The value of the property.
type	The type of the property. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=AttributeType`.
isReadonly	Indicates whether the property cannot be modified.
isExtended	Indicates whether the property is an extended attribute.

%LSAF_GETPROCESSFLOWSBYQUERY Macro

Gets the metadata for process flows that match the specified search criteria and to which you have access permission. The metadata is stored in a SAS data set.

Category: ProcessFlowService Module

Note: This macro sets the standard macro variables and _lsafPfQueryLimitReached_, which indicates whether the search limit was reached.

Syntax

%LSAF_GETPROCESSFLOWSBYQUERY(<LSAF_STATUS=*flow-status*><, LSAF_NAME=*name*><, LSAF_CONTEXTPATH=*path*><, LSAF_PROCESSDEFID=*definition-id*><, LSAF_OWNER=*user-id*><, LSAF_LASTMODIFIEDBY=*name*><, LSAF_FROMLASTMODIFIEDDATE=*date*><, LSAF_TOLASTMODIFIEDDATE=*date*><, LSAF_CREATIONMETHOD=*creation-method*><, SAS_RESULT_DSNAME=*name*>);

Optional Arguments

LSAF_STATUS=*flow-status*
The status of the process flow. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=FlowStatus`.

LSAF_NAME=*name*
The name of the process flow.

LSAF_PROCESSDEFID=*definition-id*
The identifier of the process flow definition that was used to create the process flow. To retrieve the process flow definition identifier, call %LSAF_GETPROCESSDEFSBYTYPE.

LSAF_OWNER=*user-id*
The identifier of the user who owns the process flow.

LSAF_LASTMODIFIEDBY=*name*
The identifier of the user who last modified the process flow.

LSAF_FROMLASTMODIFIEDDATE=*date*
The beginning last modified date to query. See “Format Date Values” on page 4 for information about date formats.

LSAF_TOLASTMODIFIEDDATE=*date*
The ending last modified date to query. See “Format Date Values” on page 4 for information about date formats.

LSAF_CREATIONMETHOD=*creation-method*
The creation method of the process flow. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=CreationMethod`.

SAS_RESULT_DSNAME=*name*
The name of the output data set that contains the metadata for the process flows. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETPROCESSFLOWSBYQUERY.

The data set contains a row for each process flow, sorted by the process flow name, and the following columns:

Column Name	Description
processFlowId	The identifier for the process flow.
processFlowName	The name of the process flow.
processFlowStatus	The status of the process flow.
creationMethod	The creation method of the process flow.
owner	The identifier of the user who owns the process flow.
processDefinitionId	The identifier of the process flow definition used to create the process flow.

Column Name	Description
lastModifiedBy	The identifier of the user who last modified the process flow.
lastModifiedDate	The date on which the process flow was last modified, represented as a String.
lastModifiedDateSASFormat	The date on which the process flow was last modified, represented as a SAS Datetime format.
path	The path of the context in which the process flow is located.

Note: See "Format Date Values" on page 4 for information about date formats.

Details

Gets the metadata for the process flows that matches the specified search criteria and to which you have access permission. The metadata is stored in a SAS data set.

Sets the Boolean macro variable _lsafPfQueryLimitReached_, which indicates whether the maximum limit on the number of items that can be returned by a query is reached. If the limit is reached, there might be additional process flows that meet the specified criteria. A more refined search is needed to retrieve those records. See "Boolean Macros" on page 4 for information about return values.

All arguments are optional.

Wildcards can be used in LSAF_NAME. For example: LSAF_NAME=myflow* searches for any task that starts with 'myflow'.

%LSAF_PROCESSFLOWEXISTS Macro

Indicates whether a process flow is defined for the specified context.

Category: ProcessFlowService Module

Note: This macro sets the standard macro variables and _lsafProcessFlowExists_, which indicates whether the process flow exists in the context.

Syntax

%LSAF_PROCESSFLOWEXISTS(LSAF_PATH=*path*, LSAF_PROCESSFLOW=*name*);

Required Arguments

LSAF_PATH=*path*
The path of the context to search for the process flow.

LSAF_PROCESSFLOW=*name*
The name of the process flow.

Details

Sets the Boolean macro variable _lsafProcessFlowExists_, which indicates whether a process flow is defined for the specified context. See “Boolean Macros” on page 4 for information about return values.

%LSAF_SUSPENDPROCESSFLOW Macro

Suspends a process flow.

Category: ProcessFlowService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_SUSPENDPROCESSFLOW(LSAF_PATH=*path*, LSAF_PROCESSFLOW=*name*);

Required Arguments

LSAF_PATH=*path*
The path of the context in which the process flow is located.

LSAF_PROCESSFLOW=*name*
The name of the process flow. To get the name,call %LSAF_GETPROCESSFLOWSBYQUERY.

%LSAF_UPDATEPROCESSFLOWDATA Macro

Updates the process data for the specified process flow.

Category: ProcessFlowService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_UPDATEPROCESSFLOWDATA(LSAF_PATH=*path*, LSAF_PROCESSFLOW=*name*, SAS_DSNAME=*name*);

Required Arguments

LSAF_PATH=*path*
The path of the context in which the process flow is located.

LSAF_PROCESSFLOW=*name*
The name of the process flow.

SAS_DSNAME=*name*
The name of the data set that contains the process flow data values to set on the process flow. See “Data Set Macros” on page 5 for information about specifying the name of the data set.

The data set must contain at least the following columns. Additional variables are ignored.

Column Name	Description
dataID	The identifier for the data element property.
dataValue	The value. The output data set from the %LSAF_GETPROCESSFLOWDATA macro can be used to determine the identifier and current values of the process flow data.

Details

Updates the process data for the specified process flow.

Rows in which the identifier matches an existing process flow data element are used to attempt to update the element. If the specified value is invalid for the element type, the update is stopped and the appropriate error message is printed to the SAS log file.

See “Format Date Values” on page 4 for information about date formats.

%LSAF_UPDATEPROCESSFLOWPROPERTIES Macro

Updates the properties for the specified process flow.

Category: ProcessFlowService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_UPDATEPROCESSFLOWPROPERTIES(LSAF_PATH=*path*, LSAF_PROCESSFLOW=*name*, SAS_DSNAME=*name*);

Required Arguments

LSAF_PATH=*path*
The path of the context in which the process flow is located.

LSAF_PROCESSFLOW=*name*
The name of the process flow.

SAS_DSNAME=*name*
The name of the data set that contains the property values to set on the process flow. See “Data Set Macros” on page 5 for information about specifying the name of the data set.

The data set must contain at least the following columns. Additional variables are ignored.

Column Name	Description
name	The name of the property.
value	The value.

Details

Updates the properties for the specified process flow.

The output data set from the %LSAF_GETPROCESSFLOWPROPERTIES macro can be used to determine the name and current values of the process flow properties. Rows in which the name matches an existing process flow property are used to attempt to update the property. If the specified value is invalid for the property type, the update is stopped and the appropriate error message is printed to the SAS log file.

See “Format Date Values” on page 4 for information about date formats.

40

ProcessFlowSetupService Module

Overview

Macros to manage process flow setup elements.

Macro Summary Table

Category	Language Elements	Description
ProcessFlowSetupService Module	%LSAF_GETPFSETUPELEMENTS Macro (p. 473)	Gets the metadata for the flow setup elements for the specified process flow within the specified context. The metadata is stored in a SAS data set.
	%LSAF_GETPFSETUPJOBINFO Macro (p. 474)	Gets the properties for the job setup elements for the process flow within the specified context. The properties are stored in a SAS data set.
	%LSAF_GETPFSETUPJOBPARAMETERS Macro (p. 475)	Gets the override parameters for the job setup elements for the process flow within the specified context. The parameters are stored in a SAS data set.
	%LSAF_GETPFSETUPNOTIFINFO Macro (p. 477)	Gets the properties for the notification setup elements for the process flow within the specified context. The properties are stored in a SAS data set.
	%LSAF_GETPFSETUPNOTIFRECIPS Macro (p. 478)	Gets the recipients for notification setup elements for the process flow within the specified context. The recipients are stored in a SAS data set.
	%LSAF_GETPFSETUPSIGNALLOCS Macro (p. 480)	Gets the locations for the signal setup elements for the process flow within the specified context. The locations are stored in a SAS data set.
	%LSAF_GETPFSETUPTIMERS Macro (p. 481)	Gets the data for the timer setup elements for the process flow within the specified context. The data is stored in a SAS data set.
	%LSAF_GETPFSETUPUSERCANDIDATES Macro (p. 482)	Gets the candidates for the user setup elements for the process flow within the specified context. The candidates are stored in a SAS data set.
	%LSAF_GETPFSETUPUSERINFO Macro (p. 483)	Gets the properties for the user setup elements for the process flow within the specified context. The properties are stored in a SAS data set.
	%LSAF_UPDATEPFSETUPJOBINFO Macro (p. 485)	Updates the properties for process flow job elements using a SAS data set as input.
	%LSAF_UPDATEPFSETUPJOBPARAMETERS Macro (p. 486)	Updates the job parameter overrides for process flow job elements using a SAS data set as input.
	%LSAF_UPDATEPFSETUPNOTIFINFO Macro (p. 488)	Updates the properties for process flow notification elements using a SAS data set as input.

Category	Language Elements	Description
	%LSAF_UPDATEPFSETUPNOTIFRECIPS Macro (p. 490)	Updates the recipients for process flow notification elements using a SAS data set as input.
	%LSAF_UPDATEPFSETUPSIGNALLOCS Macro (p. 492)	Updates the locations for process flow signal elements using a SAS data set as input.
	%LSAF_UPDATEPFSETUPTIMERS Macro (p. 493)	Updates the values of process flow timer elements using a SAS data set as input.
	%LSAF_UPDATEPFSETUPUSERCANDIDATES Macro (p. 495)	Updates the candidates for process flow user task elements using a SAS data set as input.
	%LSAF_UPDATEPFSETUPUSERINFO Macro (p. 497)	Updates the properties for process flow user elements using a SAS data set as input.

Dictionary

%LSAF_GETPFSETUPELEMENTS Macro

Gets the metadata for the flow setup elements for the specified process flow within the specified context. The metadata is stored in a SAS data set.

Category: ProcessFlowSetupService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETPFSETUPELEMENTS(LSAF_PATH=*path*, LSAF_PROCESSFLOW=*name*<, SAS_DSNAME=*name*>);

Required Arguments

LSAF_PATH=*path*
The path of the context in which the process flow is located.

LSAF_PROCESSFLOW=*name*
The name of the process flow.

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the elements for the process flow data. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETPFSETUPELEMENTS.

The data set contains a row for each element and columns with the following names. The data set is not sorted.

Column Name	Description
processFlowPath	The path of the context in which the process flow is located.
processFlowName	The name of the process flow.
elementType	The type of the element.
elementID	The unique identifier of the element.
elementName	The name of the element.
elementDescription	The description of the element.

%LSAF_GETPFSETUPJOBINFO Macro

Gets the properties for the job setup elements for the process flow within the specified context. The properties are stored in a SAS data set.

Category: ProcessFlowSetupService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETPFSETUPJOBINFO(LSAF_PATH=*path*, LSAF_PROCESSFLOW=*name*<, SAS_DSNAME=*name*>);

Required Arguments

LSAF_PATH=*path*
The path of the context in which the process flow is located.

LSAF_PROCESSFLOW=*name*
The name of the process flow.

Optional Argument

SAS_DSNAME=*name*

The name of the output data set that contains the properties for the job setup elements for the process flow. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETPFSETUPJOBINFO.

The data set contains a row for each property or attribute and columns with the following names. The data set is not sorted.

Column Name	Description
processFlowPath	The path of the context in which the process flow is located.
processFlowName	The name of the process flow.
elementID	The identifier for the element applicable for the property.
name	The name of the property.
displayName	The name of the property as displayed as a label within SAS Life Science Analytics Framework. Derived properties (such as name) do not have a value for displayName.
value	The value of the property.
type	The type of the property. Valid values: ■ STRING ■ LONG ■ DATE ■ BOOLEAN
isReadonly	Indicates whether the property cannot be modified.
isExtended	Indicates whether the property is an extended attribute.

%LSAF_GETPFSETUPJOBPARAMETERS Macro

Gets the override parameters for the job setup elements for the process flow within the specified context. The parameters are stored in a SAS data set.

Category: ProcessFlowSetupService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETPFSETUPJOBPARAMETERS(LSAF_PATH=*path*, LSAF_PROCESSFLOW=*name*<, SAS_DSNAME=*name*>);

Required Arguments

LSAF_PATH=*path*
The path of the context in which the process flow is located.

LSAF_PROCESSFLOW=*name*
The name of the process flow.

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the override parameters for the process flow element. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETPFSETUPJOBPARAMETERS.

The data set contains a row for each override parameter, sorted by the element identifier and parameter name, and the following columns:

Column Name	Description
processFlowPath	The path of the context in which the process flow is located.
processFlowName	The name of the process flow.
elementID	The identifier for the element applicable for the parameter.
jobPath	The path for the job.
jobVersion	The version of the job.
name	The name of the parameter.
type	The type of the parameter. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=ParameterType`.
value	The value of the parameter.

Column Name	Description
fileVersion	If the parameter is type FILE, the file version. A blank value specifies that either the file is not versioned or the file is versioned but no version was specified.
includeSubFolders	Indicates whether the subfolders of a container are included during job execution.

Details

Gets the override parameters for the job setup elements for the process flow within the specified context. The parameters are stored in a SAS data set.

Job setup elements maintain a list of the parameters that are used to override the default parameters when the job task executes. To get the full list and metadata for the job parameters, use the output data set from the %LSAF_GETJOBPARAMETERS macro.

%LSAF_GETPFSETUPNOTIFINFO Macro

Gets the properties for the notification setup elements for the process flow within the specified context. The properties are stored in a SAS data set.

Category: ProcessFlowSetupService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETPFSETUPNOTIFINFO(LSAF_PATH=*path*, LSAF_PROCESSFLOW=*name*<, SAS_DSNAME=*name*>);

Required Arguments

LSAF_PATH=*path*
The path of the context in which the process flow is located.

LSAF_PROCESSFLOW=*name*
The name of the process flow.

Optional Argument

SAS_DSNAME=*name*

The name of the output data set that contains the properties and attributes for the notification setup elements for the process flow. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETPFSETUPNOTIFINFO.

The data set contains a row for each property or attribute and columns with the following names. The data set is not sorted.

Column Name	Description
processFlowPath	The path of the context in which the process flow is located.
processFlowName	The name of the process flow.
elementID	The identifier for the element applicable for the property.
name	The name of the property.
displayName	The name of the property as displayed as a label within SAS Life Science Analytics Framework. Derived properties (such as name) do not have a value for displayName.
value	The value of the property.
type	The type of the property. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=AttributeType`.
isReadonly	Indicates whether the property cannot be modified.
isExtended	Indicates whether the property is an extended attribute.

%LSAF_GETPFSETUPNOTIFRECIPS Macro

Gets the recipients for notification setup elements for the process flow within the specified context. The recipients are stored in a SAS data set.

Category: ProcessFlowSetupService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETPFSETUPNOTIFRECIPS(LSAF_PATH=*path*, LSAF_PROCESSFLOW=*name*<, SAS_DSNAME=*name*>);

Required Arguments

LSAF_PATH=*path*
The path of the context in which the process flow is located.

LSAF_PROCESSFLOW=*name*
The name of the process flow.

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the metadata for the recipients. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETPFSETUPNOTIFRECIPS.

The data set contains a row for each recipient, sorted by the element identifier, recipient type, and principalId, and the following columns:

Column Name	Description
processFlowPath	The path of the context in which the process flow is located.
processFlowName	The name of the process flow.
elementID	The identifier for the element.
type	The recipient type. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=AttributeType`.
principalId	The name of the recipient. The returned value is either a user group name or a user identifier.
description	The description of the recipient. The returned value is either a user group name or a user display name.
grpSrcCtxt	If the recipient type is GROUP, the context path in which the user group is defined.

%LSAF_GETPFSETUPSIGNALLOCS Macro

Gets the locations for the signal setup elements for the process flow within the specified context. The locations are stored in a SAS data set.

Category: ProcessFlowSetupService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETPFSETUPSIGNALLOCS(LSAF_PATH=*path*, LSAF_PROCESSFLOW=*name*<, SAS_DSNAME=*name*>);

Required Arguments

LSAF_PATH=*path*
The path of the context in which the process flow is located.

LSAF_PROCESSFLOW=*name*
The name of the process flow.

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the locations for the process flow signal element. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETPFSETUPSIGNALLOCS.

The data set contains a row for each signal location, sorted by element identifier and location, and the following columns:

Column Name	Description
processFlowPath	The path of the context in which the process flow is located.
processFlowName	The name of the process flow.
elementID	The identifier for the element.
signalName	The name of the signal element
location	The path for the signal location.

%LSAF_GETPFSETUPTIMERS Macro

Gets the data for the timer setup elements for the process flow within the specified context. The data is stored in a SAS data set.

Category: ProcessFlowSetupService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETPFSETUPTIMERS(LSAF_PATH=*path*, LSAF_PROCESSFLOW=*name*<, SAS_DSNAME=*name*>);

Required Arguments

LSAF_PATH=*path*
The path of the context in which the process flow is located.

LSAF_PROCESSFLOW=*name*
The name of the process flow.

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the data for the process flow timer setup elements. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETPFSETUPTIMERS.

The data set contains a row for each timer element and columns with the following names. The data set is not sorted.

Column Name	Description
processFlowPath	The path of the context in which the process flow is located.
processFlowName	The name of the process flow.
elementID	The identifier for the element.
name	The name of the element.
description	The description of the timer element.
type	The type of data the timer represents. For valid values, call

Column Name	Description
	%LSAF_GETSYSTEMCONSTANTS, where `name=TimerType`.
value	The value of the timer.
years	The years value that is represented by the duration timer.
months	The months value that is represented by the duration timer.
days	The days value that is represented by the duration timer.
hours	The hours value that is represented by the duration timer.
minutes	The minutes value that is represented by the duration timer.
seconds	The seconds value that is represented by the duration timer.

%LSAF_GETPFSETUPUSERCANDIDATES Macro

Gets the candidates for the user setup elements for the process flow within the specified context. The candidates are stored in a SAS data set.

Category: ProcessFlowSetupService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETPFSETUPUSERCANDIDATES(LSAF_PATH=*path*, LSAF_PROCESSFLOW=*name*<, SAS_DSNAME=*name*>);

Required Arguments

LSAF_PATH=*path*
The path of the context in which the process flow is located.

LSAF_PROCESSFLOW=*name*
The name of the process flow.

Optional Argument

SAS_DSNAME=*name*

The name of the output data set that contains the metadata for the candidates. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETPFSETUPUSERCANDIDATES.

The data set contains a row for each candidate, sorted by the element identifier, candidate type, and principalId, and the following columns:

Column Name	Description
processFlowPath	The path of the context in which the process flow is located.
processFlowName	The name of the process flow.
elementID	The identifier for the element.
type	The candidate type. Valid values: ■ USER ■ GROUP
principalId	The name of the candidate. The returned value is either a user group name or a user identifier.
description	The description of the candidate. The returned value is either a user group description or a user display name.
grpSrcCtxt	If the candidate is a user group, the context path in which the user group is defined.

%LSAF_GETPFSETUPUSERINFO Macro

Gets the properties for the user setup elements for the process flow within the specified context. The properties are stored in a SAS data set.

Category: ProcessFlowSetupService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETPFSETUPUSERINFO(LSAF_PATH=*path*, LSAF_PROCESSFLOW=*name*<, SAS_DSNAME=*name*>);

Required Arguments

LSAF_PATH=*path*
The path of the context in which the process flow is located.

LSAF_PROCESSFLOW=*name*
The name of the process flow.

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the property attributes for the user setup elements. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETPFSETUPUSERINFO.

The data set contains a row for each property or attribute and columns with the following names. The data set is not sorted.

Column Name	Description
processFlowPath	The path of the context in which the process flow is located.
processFlowName	The name of the process flow.
elementID	The identifier for the element applicable for the property.
name	The name of the property.
value	The value of the property.
displayName	The name of the property as displayed as a label within SAS Life Science Analytics Framework. Derived properties (such as name) do not have a value for displayName.
type	The type of the property. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=AttributeType`.
isReadonly	Indicates whether the property cannot be modified.

Column Name	Description
isExtended	Indicates whether the property is an extended attribute.

%LSAF_UPDATEPFSETUPJOBINFO Macro

Updates the properties for process flow job elements using a SAS data set as input.

Category: ProcessFlowSetupService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_UPDATEPFSETUPJOBINFO(SAS_DSNAME=*name*);

Required Argument

SAS_DSNAME=*name*

The name of the data set that contains the properties information for all of the job element properties to modify. See “Data Set Macros” on page 5 for information about specifying the name of the data set.

The data set must contain at least the following columns. Additional variables are ignored.

Column Name	Description
processFlowPath	The path of the context in which the process flow is located.
processFlowName	The name of the process flow.
elementID	The identifier for the element applicable for the property.
name	The name of the property.
value	The value of the property.

Details

Updates the properties for process flow job elements using a SAS data set as input.

Properties for multiple job elements can be modified using a single SAS data set. Each property is represented as a single row in the input data set.

To determine the name and current values of the job element properties, use the output data set from the %LSAF_GETPFSETUPJOBINFO macro. Rows in which the name matches an existing job element property are used to attempt to update the property. If the specified value is invalid for the property type, the update is stopped and the appropriate error message is written to the SAS log file.

Failure and Message Handling

The macro performs a series of validation procedures to determine whether the input data set has the necessary structure for processing. If any of these validation procedures fail, an appropriate failure message is written to the SAS log file and the macro stops processing.

In addition to other modifications, the macro sorts the input data set and stores the modified data in a temporary data set called WORK.__SASMACRO_JOBINFOTEMP__. In the event that a failure occurs during processing, this temporary data set is retained as a reference. If all updates are performed successfully, this data set is deleted from the work library. A message is written to the SAS log file, which indicates whether the data set was retained. Regardless of success or failure of previous runs, this data set is deleted at the beginning of each new run of the macro. The data values and sort order of the original input data set are maintained.

All records that pertain to the same job element are processed as a single transaction by the macro. Messages are written to the SAS log file as each job element is processed. If the update of a job element fails, the macro stops processing and a failure message is written to the SAS log file. Successful updates to job elements made prior to the observation that contains the failure are maintained in SAS Life Science Analytics Framework.

%LSAF_UPDATEPFSETUPJOBPARAMETERS Macro

Updates the job parameter overrides for process flow job elements using a SAS data set as input.

Category: ProcessFlowSetupService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_UPDATEPFSETUPJOBPARAMETERS(SAS_DSNAME=*name*);

Required Argument

SAS_DSNAME=*name*

The name of the data set that contains the override parameters for the job elements to modify. See "Data Set Macros" on page 5 for information about specifying the name of the data set.

The data set must contain at least the following columns. Additional variables are ignored.

Column Name	Description
processFlowPath	The path of the context in which the process flow is located.
processFlowName	The name of the process flow.
elementID	The identifier for the element applicable for the property.
name	The name of the job parameter.
type	The parameter type. For a list of valid types, call %LSAF_GETSYSTEMCONSTANTS where `name=ParameterType`.
value	The parameter value to use when the job is executed. The value must match the parameter type. See "Format Date Values" for information about date formats.
fileVersion	If the parameter type is FILE, this parameter specifies the file version to use. For versioned files, a blank value means that the latest file version is used. If the file is unversioned, is a workspace file, or the parameter type is not FILE, this value is ignored.
includeSubFolders	If the parameter type is FOLDER, this parameter indicates whether the subfolders of the container are included during job execution. If the parameter type is not FOLDER, this value is ignored.

Details

Updates the job parameter overrides for process flow job elements using a SAS data set as input.

The override parameters for multiple job elements can be modified using a single SAS data set. Each parameter is represented as a single row in the input data set.

The job element stores only the values to use as overrides at the time the job is run. To determine the name and default values of all of the parameters for the job associated with the process flow element, use the output data set from the %LSAF_GETPFSETUPJOBPARAMETERS macro. You can also use the same output data set to determine the name and value for the overrides that are currently set on the job element. To determine the job elements for a process flow and the job that is associated with the elements, use the output data set from the %LSAF_GETPFSETUPJOBINFO macro. If the job path has not been set on the job element, parameter overrides cannot be set.

The job parameter overrides that are represented in the input data set replace the list of parameter overrides currently saved on the job element. A single row in which both NAME and TYPE are blank results in clearing all of the job parameter overrides for the specified element. If an invalid value is specified for a parameter, the updates are stopped and the appropriate error message is written to the SAS log file.

If duplicate valid parameters are specified for the same element, only the first value is maintained.

Failure and Message Handling

The macro performs a series of validation procedures to determine whether the input data set has the necessary structure for processing. If any of these validation procedures fail, an appropriate failure message is written to the SAS log file and the macro stops additional processing.

In addition to other modifications, the macro sorts the input data set and stores the modified data in a temporary data set called WORK.__SASMACRO_JOBELEMENTPARMSTEMP__. In the event that a failure occurs during processing, this temporary data set is retained as a reference. If all updates are performed successfully, this data set is deleted from the work library. A message is written to the SAS log file, which indicates whether the data set was retained. Regardless of the success or failure of previous runs, this data set is deleted at the beginning of each new run of the macro. The data values and sort order of the original input data set are maintained.

All records that pertain to the same job element are processed as a single transaction by the macro. Messages are written to the SAS log file as each job element is processed. If the update of a job element fails, the macro stops processing and a failure message is written to the SAS log file. Successful updates to job elements made prior to the observation that contains the failure are maintained in SAS Life Science Analytics Framework.

%LSAF_UPDATEPFSETUPNOTIFINFO Macro

Updates the properties for process flow notification elements using a SAS data set as input.

Category: ProcessFlowSetupService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_UPDATEPFSETUPNOTIFINFO(SAS_DSNAME=*name*);

Required Argument

SAS_DSNAME=*name*

The name of the data set that contains the properties information for all of the notification element properties to modify. See "Data Set Macros" on page 5 for information about specifying the name of the data set.

The data set must contain at least the following columns. Additional variables are ignored.

Column Name	Description
processFlowPath	The path of the context in which the process flow is located.
processFlowName	The name of the process flow.
elementID	The identifier for the element applicable for the property.
name	The name of the property.
value	The value of the property.

Details

Updates the properties for process flow notification elements using a SAS data set as input.

Properties for multiple notification elements can be modified using a single SAS data set. Each property is represented as a single row in the input data set.

To determine the name and current values of the notification element properties, use the output data set from the %LSAF_GETPFSETUPNOTIFINFO macro. Rows in which the name matches an existing notification element property are used to attempt to update the property. If the specified value is invalid for the property type, the update is stopped and the appropriate error message is written to the SAS log file.

Failure and Message Handling

The macro performs a series of validation procedures to determine whether the input data set has the necessary structure for processing. If any of these validation

procedures fail, an appropriate failure message is written to the SAS log file and the macro stops additional processing.

In addition to other modifications, the macro sorts the input data set and stores the modified data in a temporary data set called WORK.__SASMACRO_NOTIFICATIONSTEMP__. In the event that a failure occurs during processing, this temporary data set is retained as a reference. If all updates are performed successfully, this data set is deleted from the work library. A message is written to the SAS log file, which indicates whether the data set was retained. Regardless of success or failure of previous runs, this data set is deleted at the beginning of each new run of the macro. The data values and sort order of the original input data set are maintained.

All records that pertain to the same notification element are processed as a single transaction by the macro. Messages are written to the SAS log file as each notification element is processed. If the update of a notification element fails, the macro stops processing and a failure message is written to the SAS log file. Successful updates to notification elements made prior to the observation that contains the failure are maintained in SAS Life Science Analytics Framework.

%LSAF_UPDATEPFSETUPNOTIFRECIPS Macro

Updates the recipients for process flow notification elements using a SAS data set as input.

Category: ProcessFlowSetupService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_UPDATEPFSETUPNOTIFRECIPS(SAS_DSNAME=*name*);

Required Argument

SAS_DSNAME=*name*

The name of the data set that contains the recipients for the notification elements to modify. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETACLS.

The data set must contain at least the following columns. Additional variables are ignored.

Column Name	Description
processFlowPath	The path of the context in which the process flow is located.
processFlowName	The name of the process flow.
elementID	The identifier for the element applicable for the property.

Column Name	Description
type	The recipient type. Valid values: ■ USER ■ GROUP
principalId	The name of the recipient. The value is either a user group name or a user identifier.
grpSrcCtxt	The context path in which a user group recipient is defined. If the recipient is type USER, the value is ignored.

Details

Updates the recipients for process flow notification elements using a SAS data set as input.

The recipients for multiple notification elements can be modified using a single SAS data set. Each recipient is represented as a single row in the input data set.

To determine the recipients currently set on the notification element, use the output data set from the %LSAF_GETPFSETUPNOTIFRECIPS macro. To determine the principals (user accounts and user groups) that are available at the context level eligible to be recipients, use the output data set from the %LSAF_GETASSIGNEDMEMBERS macro. To determine the identifiers for all notification elements for a process flow, use the output data set from the %LSAF_GETPFSETUPELEMENTS macro.

The notification recipients that are represented in the input data set replace the list of recipients currently saved on the notification element. A single row in which TYPE and PRINCIPALID are blank result in clearing all of the notification recipients for the specified element.

Failure and Message Handling

The macro performs a series of validation procedures to determine whether the input data set has the necessary structure for processing. If any of these validation procedures fail, an appropriate failure message is written to the SAS log file and the macro stops additional processing.

In addition to other modifications, the macro sorts the input data set and stores the modified data in a temporary data set called WORK._SASMACRO_NOTIRECIPIENTSTEMP__. In the event that a failure occurs during processing, this temporary data set is retained as a reference. If all updates are performed successfully, this data set is deleted from the work library. A message is written to the SAS log file, which indicates whether the data set was retained. Regardless of success or failure of previous runs, this data set is deleted at the beginning of each new run of the macro. The data values and sort order of the original input data set are maintained.

All records that pertain to the same notification element are processed as a single transaction by the macro. Messages are written to the SAS log file as each

notification element is processed. If the update of a notification element fails, the macro stops processing and a failure message is written to the SAS log file. Successful updates to notification elements made prior to the observation that contains the failure are maintained in SAS Life Science Analytics Framework.

If duplicate valid recipients are specified for the same element, only one of the values is maintained.

%LSAF_UPDATEPFSETUPSIGNALLOCS Macro

Updates the locations for process flow signal elements using a SAS data set as input.

Category: ProcessFlowSetupService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_UPDATEPFSETUPSIGNALLOCS(SAS_DSNAME=*name*);

Required Argument

SAS_DSNAME=*name*

The name of the data set that contains the locations for the signal elements to modify. See “Data Set Macros” on page 5 for information about specifying the name of the data set.

The data set must contain at least the following columns. Additional variables are ignored.

Column Name	Description
processFlowPath	The path of the context in which the process flow is located.
processFlowName	The name of the process flow.
elementID	The identifier for the element applicable for the property.
location	The path for the signal location.

Details

Updates the locations for process flow signal elements using a SAS data set as input.

The locations for multiple signal elements can be modified using a single SAS data set. Each location is represented as a single row in the input data set.

To determine the locations that are currently set on the signal element, use the output data set from the %LSAF_GETPFSETUPSIGNALLOCS macro. To determine the signal elements for a process flow, use the output data set from the %LSAF_GETPFSETUPELEMENTS macro.

The signal locations that are represented in the input data set replace the list of locations currently saved on the signal element. A single row in which LOCATION is blank result in clearing all signal locations for the specified element.

Failure and Message Handling

The macro performs a series of validation procedures to determine whether the input data set has the necessary structure for processing. If any of these validation procedures fail, an appropriate failure message is written to the SAS log file and the macro stops additional processing.

Signal locations must adhere to the valid format of a SAS Life Science Analytics Framework repository item with the exception of container paths, which must end in slash star "/*". If an invalid value is specified for a location, the updates are stopped and the appropriate error message is written to the SAS log file.

In addition to other modifications, the macro sorts the input data set and stores the modified data in a temporary data set called WORK.__SASMACRO_SIGNALLOCATIONSTEMP__. In the event that a failure occurs during processing, this temporary data set is retained as a reference. If all updates are performed successfully, this data set is deleted from the work library. A message is written to the SAS log file, which indicates whether the data set was retained. Regardless of success or failure of previous runs, this data set is deleted at the beginning of each new run of the macro. The data values and sort order of the original input data set are maintained.

All records that pertain to the same signal element are processed as a single transaction by the macro. Messages are written to the SAS log file as each signal element is processed. If the update of a signal element fails, the macro stops processing and a failure message is written to the SAS log file. Successful updates to signal elements made prior to the observation that contains the failure are maintained in SAS Life Science Analytics Framework.

If duplicate valid locations are specified for the same element, only the first value is maintained.

%LSAF_UPDATEPFSETUPTIMERS Macro

Updates the values of process flow timer elements using a SAS data set as input.

Category: ProcessFlowSetupService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_UPDATEPFSETUPTIMERS(SAS_DSNAME=*name*);

Required Argument

SAS_DSNAME=*name*

The name of the data set that contains the timer values to set on the timer elements. See "Data Set Macros" on page 5 for information about specifying the name of the data set.

The data set must contain at least the following columns. Additional variables are ignored.

Column Name	Description
processFlowPath	The path of the context in which the process flow is located.
processFlowName	The name of the process flow.
elementID	The identifier for the element.
value	The value of a date timer. This is ignored when timer type is DURATION.
years	The years value for a duration timer. This is ignored when timer type is DATE.
months	The months value for a duration timer. This is ignored when timer type is DATE.
days	The days value a duration timer. This is ignored when timer type is DATE.
hours	The hours value for a duration timer. This is ignored when timer type is DATE.
minutes	The minutes value for a duration timer. This is ignored when timer type is DATE.
seconds	The seconds value for a duration timer. This is ignored when timer type is DATE.

Details

Updates the values of process flow timer elements using a SAS data set as input. Each row of the data set represents a timer element.

To determine the identifier and current values of the process flow timer elements, use the output data set from the %LSAF_GETPFSETUPTIMERS macro. Rows in which the identifier matches an existing process flow timer element are used to attempt to update the element. If the specified value is invalid for the element type, the update is stopped and the appropriate error message is written to the SAS log file.

See "Format Date Values" on page 4 for information about date formats.

Failure and Message Handling

The macro performs a series of validation procedures to determine whether the input data set has the necessary structure for processing. If any of these validation procedures fail, an appropriate failure message is written to the SAS log file and the macro stops additional processing.

If a failure occurs while processing a specific timer element, the macro stops processing and a failure message is written to the SAS log file. Successful updates made to elements prior to the element that contains the failure is maintained in SAS Life Science Analytics Framework.

%LSAF_UPDATEPFSETUPUSERCANDIDATES Macro

Updates the candidates for process flow user task elements using a SAS data set as input.

Category: ProcessFlowSetupService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_UPDATEPFSETUPUSERCANDIDATES(SAS_DSNAME=*name*);

Required Argument

SAS_DSNAME=*name*

The name of the data set that contains the candidates for the user task elements to modify. See "Data Set Macros" on page 5 for information about specifying the name of the data set.

The data set must contain at least the following columns. Additional variables are ignored.

Column Name	Description
processFlowPath	The path of the context in which the process flow is located.
processFlowName	The name of the process flow.

Column Name	Description
elementID	The identifier for the element applicable for the property.
type	The candidate type. Valid values: ■ USER ■ GROUP
principalId	The name of the candidate. The value is either a user group name or a user identifier.
grpSrcCtxt	The context path in which a user group candidate is defined. If the candidate is type user, the value is ignored.

Details

Updates the candidates for process flow user task elements using a SAS data set as input.

The candidates for multiple user task elements can be modified using a single SAS data set. Each candidate is represented as a single row in the input data set. To determine the candidates that are currently set on the user task element, use the output data set from the %LSAF_GETPFSETUPUSERCANDIDATES macro.

To determine the principals (users and user groups) available at the context level that are eligible to be candidates, use the output data set from the %LSAF_GETASSIGNEDMEMBERS macro. To determine the identifiers for all of user elements for a process flow, use the output data set from the %LSAF_GETPFSETUPELEMENTS macro.

The user task candidates that are represented in the input data set replace the list of candidates that is currently saved on the user task element. A single row in which TYPE and PRINCIPALID are blank result in clearing all user task candidates for the specified element.

Failure and Message Handling

The macro performs a series of validation procedures to determine whether the input data set has the necessary structure for processing. If any of these validation procedures fail, an appropriate failure message is written to the SAS log file and the macro stops additional processing.

In addition to other modifications, the macro sorts the input data set and stores the modified data in a temporary data set called WORK.__SASMACRO_USERCANDIDATESTEMP__. In the event that a failure occurs during processing, this temporary data set is retained as a reference. If all updates are performed successfully, this data set is deleted from the work library. A message is written to the SAS log file, which indicates whether the data set was retained. Regardless of success or failure of previous runs, this data set is deleted

at the beginning of each new run of the macro. The data values and sort order of the original input data set are maintained.

All records that pertain to the same user task element are processed as a single transaction by the macro. Messages are written to the SAS log file as each user task element is processed. If the update of a user task element fails, the macro stops processing and a failure message is written to the SAS log file. Successful updates to user task elements made prior to the observation that contains the failure are maintained in SAS Life Science Analytics Framework.

If duplicate valid candidates are specified for the same element, only one of the values is maintained.

%LSAF_UPDATEPFSETUPUSERINFO Macro

Updates the properties for process flow user elements using a SAS data set as input.

Category: ProcessFlowSetupService Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_UPDATEPFSETUPUSERINFO(SAS_DSNAME=*name*);

Required Argument

SAS_DSNAME=*name*

The name of the data set that contains the properties information for all of the user element properties to modify. See “Data Set Macros” on page 5 for information about specifying the name of the data set.

The data set must contain at least the following columns. Additional variables are ignored.

Column Name	Description
processFlowPath	The path of the context in which the process flow is located.
processFlowName	The name of the process flow.
elementID	The identifier for the element applicable for the property.
name	The name of the property.
value	The value of the property.

Details

Updates the properties for process flow user elements using a SAS data set as input.

Properties for multiple user elements can be modified using a single SAS data set. Each property is represented as a single row in the input data set.

To determine the name and current values of the user element properties, use the output data set from the %LSAF_GETPFSETUPUSERINFO macro. Rows in which the name matches an existing user element property are used to attempt to update the property. If the specified value is invalid for the property type, the update is stopped and the appropriate error message is written to the SAS log file.

Failure and Message Handling

The macro performs a series of validation procedures to determine whether the input data set has the necessary structure for processing. If any of these validation procedures fail, an appropriate failure message is written to the SAS log file and the macro stops additional processing.

In addition to other modifications, the macro sorts the input data set and stores the modified data in a temporary data set called WORK.__SASMACRO_USERELEMENTSTEMP__. In the event that a failure occurs during processing, this temporary data set is retained as a reference. If all updates are performed successfully, this data set is deleted from the work library. A message is written to the SAS log file, which indicates whether the data set was retained. Regardless of success or failure of previous runs, this data set is deleted at the beginning of each new run of the macro. The data values and sort order of the original input data set are maintained.

All records that pertain to the same user element are processed as a single transaction by the macro. Messages are written to the SAS log file as each user element is processed. If the update of a user element fails, the macro stops processing and a failure message is written to the SAS log file. Successful updates to user elements made prior to the observation that contains the failure are maintained in SAS Life Science Analytics Framework.

41

ProcessFlowTask Module

Overview

Macros to manage process flow active tasks and elements.

Macro Summary Table

Category	Language Elements	Description
ProcessFlowTask Module	%LSAF_ASSIGNPFUSERTASK Macro (p. 501)	Assigns a task to a user. If no user is specified, the task is unassigned.
	%LSAF_CLAIMPFUSERTASK Macro (p. 502)	Claims a user task by assigning the task to the current user. Tasks in a started state cannot be claimed.
	%LSAF_CLEARPFUSERTASKCANDIDATES Macro (p. 502)	Removes all of the candidates from the specified process flow user task.
	%LSAF_COMPLETEPFUSERTASK Macro (p. 503)	Completes a process flow user task.
	%LSAF_GETPFCOMPLETEDDETAILS Macro (p. 504)	Gets the details for the completed elements of the process flow that is in the specified context. The data is stored in a SAS data set.
	%LSAF_GETPFCOMPLETEDELEMENTS Macro (p. 506)	Gets the metadata for the completed elements for the specified process flow within the specified context. The metadata is stored in a SAS data set.
	%LSAF_GETPFCURRENTELEMENTS Macro (p. 507)	Gets the metadata for the current elements for the specified process flow within the specified context. The metadata is stored in a SAS data set.
	%LSAF_GETPFTASKSBYQUERY Macro (p. 508)	Gets the active process flow tasks based on the specified query.
	%LSAF_GETPFUSERTASKCANDIDATES Macro (p. 510)	Gets the candidates of the specified task. The data is stored in a SAS data set.
	%LSAF_GETPFUSERTASKCOMPLETIONDATA Macro (p. 512)	Gets the completion data for the specified task. The metadata is stored in a SAS data set.
	%LSAF_GETPFUSERTASKHISTORY Macro (p. 513)	Gets the task assignment history for the specified task. The metadata is stored in a SAS data set.
	%LSAF_GETPFUSERTASKPROPERTIES Macro (p. 514)	Gets the properties of the specified task. The data is stored in a SAS data set.
	%LSAF_GETPFUSERTASKSTATUS Macro (p. 515)	Gets the status of a process flow user task.

Category	Language Elements	Description
	%LSAF_SETPFUSERTASKCOMPLETIONDATA Macro (p. 516)	Updates the process flow user task completion data using a SAS data set as input.
	%LSAF_STARTPFUSERTASK Macro (p. 517)	Starts a user task. A task must be assigned to the current user before it can be started.
	%LSAF_UPDATEPFUSERTASKCANDIDATES Macro (p. 517)	Updates the process flow user task candidates using a SAS data set as input.
	%LSAF_UPDATEPFUSERTASKPROPERTIES Macro (p. 519)	Updates the process flow user task properties using a SAS data set as input.

Dictionary

%LSAF_ASSIGNPFUSERTASK Macro

Assigns a task to a user. If no user is specified, the task is unassigned.

Category: ProcessFlowTask Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_ASSIGNPFUSERTASK(LSAF_TASKID=*task-id*<, LSAF_USERID=*user-id*>);

Required Argument

LSAF_TASKID=*task-id*
The identifier of the task. To get the identifier, call %LSAF_GETPFTASKSBYQUERY.

Optional Argument

LSAF_USERID=*user-id*
The identifier of the user. If this argument is blank or not specified, the task is unassigned. To find the identifier, call %LSAF_GETALLUSERS.

%LSAF_CLAIMPFUSERTASK Macro

Claims a user task by assigning the task to the current user. Tasks in a started state cannot be claimed.

Category: ProcessFlowTask Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_CLAIMPFUSERTASK(LSAF_TASKID=*task-id*);

Required Argument

LSAF_TASKID=*task-id*
The identifier of the task. To get the identifier, call %LSAF_GETPFTASKSBYQUERY.

%LSAF_CLEARPFUSERTASKCANDIDATES Macro

Removes all of the candidates from the specified process flow user task.

Category: ProcessFlowTask Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_CLEARPFUSERTASKCANDIDATES(LSAF_TASKID=*task-id*);

Required Argument

LSAF_TASKID=*task-id*
The identifier of the process flow user task. To get the identifier, call %LSAF_GETPFTASKSBYQUERY.

%LSAF_COMPLETEPFUSERTASK Macro

Completes a process flow user task.

Category: ProcessFlowTask Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_COMPLETEPFUSERTASK(LSAF_TASKID=*task-id*<, LSAF_ACTUALHOURS=*actual-hours*><, LSAF_COMMENT=*comment*><, SAS_DSNAME=*name*>);

Required Argument

LSAF_TASKID=*task-id*
The identifier of the process flow user task. To get the identifier, call %LSAF_GETPFTASKSBYQUERY.

Optional Arguments

LSAF_ACTUALHOURS=*actual-hours*
The actual hours represented as *n.n*. If no value is specified, 0.0 is used.

LSAF_COMMENT=*comment*
The comment for completing a user task.

SAS_DSNAME=*name*
The name of the input data set that contains the details for the process flow user task completion data. See “Data Set Macros” on page 5 for information about specifying the name of the data set.

The following columns are required for processing, all other columns are ignored:

Column Name	Description
dataId	The identifier of the process flow data completion element.
dataValue	The value of the process flow completion data element.

Details

Completes the process flow user task. An optional data set can be used as input to update completion attributes. The output data set from the %LSAF_GETPFUSERTASKCOMPLETIONDATA macro can be used as input to the macro.

%LSAF_GETPFCOMPLETEDDETAILS Macro

Gets the details for the completed elements of the process flow that is in the specified context. The data is stored in a SAS data set.

Category: ProcessFlowTask Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETPFCOMPLETEDDETAILS(LSAF_PATH=*path*, LSAF_PROCESSFLOW=*name*<, LSAF_ELEMENTID=*element-id*><, SAS_DSNAME=*name*>);

Required Arguments

LSAF_PATH=*path*
The path of the context in which the process flow is located.

LSAF_PROCESSFLOW=*name*
The name of the process flow.

Optional Arguments

LSAF_ELEMENTID=*element-id*
The identifier of the completed element for which to report the details. To get the identifier,call %LSAF_GETPFCOMPLETEDELEMENTS. If this argument is not specified, the details for all of the completed elements for the process flow are returned.

SAS_DSNAME=*name*
The name of the output data set that contains the details for the process flow completed task data. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETPFCOMPLETEDDETAILS.

The data set contains a row for each detail, sorted in descending order by elementCompleted, taskId, recordTypeSortOrder, id, and name, and the following columns:

Column Name	Description
processFlowPath	The path of the context in which the process flow is located.
processFlowName	The name of the process flow.
elementID	The identifier of the element.
elementName	The name of the element.
elementType	The type for the element.
taskID	The identifier of the task.
elementCreated	The date on which the element was created.
elementCreatedSasFormat	The date on which the element was created, in SAS format.
elementCompleted	The date on which the element was completed.
elementCompletedSasFormat	The date on which the element was completed in SAS format.
elementCompletedBy	The user who completed the element.
elementCompletedState	The completion status of the element.
recordType	The type of element detail for the row. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=RecordType`.
recordTypeSortOrder	The record type sort order.
id	The identifier of the element detail, when applicable. This value is repeated for details that span across rows, such as user task history.
name	The name of the element detail.
value	The value of the element detail.

Note: See “Format Date Values” on page 4 for information about date formats.

%LSAF_GETPFCOMPLETEDELEMENTS Macro

Gets the metadata for the completed elements for the specified process flow within the specified context. The metadata is stored in a SAS data set.

Category: ProcessFlowTask Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETPFCOMPLETEDELEMENTS(LSAF_PATH=*path*, LSAF_PROCESSFLOW=*name*<, SAS_DSNAME=*name*>);

Required Arguments

LSAF_PATH=*path*
The path of the context in which the process flow is located.

LSAF_PROCESSFLOW=*name*
The name of the process flow.

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the completed elements for the process flow. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETPFCOMPLETEDELEMENTS.

The data set contains a row for each element and columns with the following names. The data set is not sorted.

Column Name	Description
processFlowPath	The path of the context in which the process flow is located.
processFlowName	The name of the process flow.
elementType	The type of the element.
elementID	The unique identifier of the element.
elementName	The name of the element.
elementDescription	The description of the element.

Column Name	Description
elementCreated	The date on which the element was created, in SAS format.
elementCompleted	The date on which the element was completed.
elementCompletedBy	The user who completed the element.

Note: See "Format Date Values" on page 4 for information about date formats.

%LSAF_GETPFCURRENTELEMENTS Macro

Gets the metadata for the current elements for the specified process flow within the specified context. The metadata is stored in a SAS data set.

Category: ProcessFlowTask Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETPFCURRENTELEMENTS(LSAF_PATH=*path*, LSAF_PROCESSFLOW=*name*<, SAS_DSNAME=*name*>);

Required Arguments

LSAF_PATH=*path*
The path of the context in which the process flow is located.

LSAF_PROCESSFLOW=*name*
The name of the process flow.

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the current elements for the process flow. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETPFCURRENTELEMENTS.

The data set contains a row for each element and columns with the following names. The data set is not sorted.

Column Name	Description
processFlowName	The name of the process flow.
elementType	The type of the element.
elementID	The unique identifier of the element.
elementName	The name of the element.
elementDescription	The description of the element.
elementStatus	The status of the element.
elementCreated	The date on which the element was created, in SAS format. See "Format Date Values" on page 4 for information about date formats.

%LSAF_GETPFTASKSBYQUERY Macro

Gets the active process flow tasks based on the specified query.

Category: ProcessFlowTask Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETPFTASKSBYQUERY(<LSAF_STATUS=*task-status*><, LSAF_ELEMENTID=*element-id*><, LSAF_ELEMENTNAME=*name*><, LSAF_PROCESSFLOWNAME=*name*><, LSAF_ASSIGNEETYPE=*assignee-type*><, LSAF_ASSIGNEES=*name*><, LSAF_FROMDUEDATE=*date*><, LSAF_TODUEDATE=*date*><, LSAF_PRIORITY=*priority*><, LSAF_TASKLIMIT=*limit*><, SAS_RESULT_DSNAME=*name*>);

Optional Arguments

LSAF_STATUS=*task-status*
The case-insensitive status of the user task. If no status is specified, all are queried. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=TaskStatus`.

LSAF_ELEMENTID=*element-id*
The element identifier of the user task.

LSAF_ELEMENTNAME=*name*
The name of the user task.

LSAF_PROCESSFLOWNAME=*name*
The name of the process flow.

LSAF_ASSIGNEETYPE=*assignee-type*
The case-insensitive assignee type. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=AssigneeType`.

LSAF_ASSIGNEES=*name*
The case-sensitive assignees associated with the user task. This can be specified as a comma-delimited string. A comma separated list must be used with the %STR() macro function. For example, to search for two users, use the following format: LSAF_ASSIGNEES=%str(user1, user2).

Required when LSAF_ASSIGNEETYPE is SPECIFIC_ASSIGNEE.

LSAF_FROMDUEDATE=*date*
The beginning due date to query. See “Format Date Values” on page 4 for information about date formats.

LSAF_TODUEDATE=*date*
The ending due date to query. See “Format Date Values” on page 4 for information about date formats.

LSAF_PRIORITY=*priority*
The case-insensitive priority of the user task. If no priority is entered all are queried. For valid values, call %LSAF_GETSYSTEMCONSTANTS, where `name=Priority`.

LSAF_TASKLIMIT=*limit*
The maximum number of user tasks to return at one time. The default is 5000. A message is printed to the SAS log file to indicate whether the limit is reached. If true, there might be more tasks that meet the search criteria. A more refined search is needed to retrieve those records.

SAS_RESULT_DSNAME=*name*
The name of the output data set that contains the queried tasks. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETPFTASKSBYQUERY.

The data set contains the following columns:

Column Name	Description
processFlowId	The process flow identifier.
processFlowStatus	The flow status of the process flow.
elementId	The element identifier of the user task
elementName	The element name of the item.
elementType	The element type.
taskId	The task identifier.
taskDescription	The description of the user task.

Column Name	Description
taskCreated	The creation date of the user task.
taskStatus	The status of the user task.
assignees	The assignee for the user task.
priority	The priority level of the user task.
complexity	The complexity level of the user task.
dateStarted	The date on which the user task was last started.
dateStartedSASFormat	The dateStarted represented as a SAS Datetime format.
dueDate	The date on which the user task is due.
dueDateSASFormat	The dueDate represented as a SAS Datetime format.

Note: See “Format Date Values” on page 4 for information about date formats.

Details

Gets a list of process flow tasks based on the specified query criteria.

The number of items that are returned by a search is limited to 5000. A message is printed to the SAS log file to indicate whether the limit was reached. If True, there might be more tasks that meet the search criteria. A more refined search is needed to retrieve those records.

Wildcards can be used in both LSAF_NAME and LSAF_PROCESSFLOWNAME. For example:

- LSAF_NAME=task* searches for a task that starts with 'task'.
- LSAF_PROCESSFLOWNAME=*flow* searches for a task in which the process flow name contains 'flow'.

See “Format Date Values” on page 4 for information about date formats.

%LSAF_GETPFUSERTASKCANDIDATES Macro

Gets the candidates of the specified task. The data is stored in a SAS data set.

Category: ProcessFlowTask Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETPFUSERTASKCANDIDATES(LSAF_TASKID=*task-id*<, SAS_DSNAME=*name*>);

Required Argument

LSAF_TASKID=*task-id*

The identifier of the task. To get the identifier, call %LSAF_GETPFTASKSBYQUERY.

Optional Argument

SAS_DSNAME=*name*

The name of the output data set that contains the details for the process flow user task candidates. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETPFUSERTASKCANDIDATES.

The data set contains a row for each candidate and columns with the following names. The data set is not sorted.

Column Name	Description
taskId	The identifier of the task.
candidateType	The candidate type. Valid values: ■ USER ■ GROUP
candidateId	The identifier of the candidate. The returned values are either a user group name or a user identifier.
description	The description of the candidate. The returned values are either a user group name or a user display name.
grpSrcCxt	The context in which the user group is defined. The value is blank when candidateType is USER.

%LSAF_GETPFUSERTASKCOMPLETIONDATA Macro

Gets the completion data for the specified task. The metadata is stored in a SAS data set.

Category: ProcessFlowTask Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETPFUSERTASKCOMPLETIONDATA(LSAF_TASKID=*task-id*<, SAS_DSNAME=*name*>);

Required Argument

LSAF_TASKID=*task-id*
The identifier of the task. To get the identifier, call %LSAF_GETPFTASKSBYQUERY.

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the details for the process flow user task completion data. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETPFUSERTASKCOMPLETIONDATA.

The data set contains a row for each completion data element and columns with the following names. The data set is not sorted.

Column Name	Description
taskId	The task identifier
dataID	The identifier of the process flow completion data element.
dataName	The name of the process flow data completion element.
dataType	The type of the process flow data completion element.

Column Name	Description
dataValue	The value of the process flow data completion element, represented as a String.
dataEnumValues	The comma-delimited list of the defined values for the completion data of the type ENUM.

%LSAF_GETPFUSERTASKHISTORY Macro

Gets the task assignment history for the specified task. The metadata is stored in a SAS data set.

Category: ProcessFlowTask Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETPFUSERTASKHISTORY(LSAF_TASKID=*task-id*<, SAS_DSNAME=*name*>);

Required Argument

LSAF_TASKID=*task-id*
The identifier of the task. To get the identifier, call %LSAF_GETPFTASKSBYQUERY.

Optional Argument

SAS_DSNAME=*name*
The name of the output data set that contains the details for the process flow user task history data. See "Data Set Macros" on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETPFUSERTASKHISTORY.

The data set contains a row for each history entity and columns with the following names. The data set is not sorted.

Column Name	Description
taskId	The task identifier.
elementID	The element identifier.

Column Name	Description
dateAssigned	The date on which the task was assigned.
dateAssignedSASFormat	The date on which the task was assigned (SAS Format).
assignee	The user identifier of the assignee or <unassigned>.
assigneeDisplayName	If the task was assigned, the display name of the assignee.
dateStarted	The date on which the task was started.
dateStartedSASFormat	The date on which the task was started, in SAS format.
actualHours	The actual hours for the task assignment.

Note: See "Format Date Values" on page 4 for information about date formats.

%LSAF_GETPFUSERTASKPROPERTIES Macro

Gets the properties of the specified task. The data is stored in a SAS data set.

Category: ProcessFlowTask Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_GETPFUSERTASKPROPERTIES(LSAF_TASKID=*task-id*<, SAS_DSNAME=*name*>);

Required Argument

LSAF_TASKID=*task-id*

The identifier of the task. To get the identifier, call %LSAF_GETPFTASKSBYQUERY.

Optional Argument

SAS_DSNAME=*name*

The name of the output data set that contains the details for the process flow user task properties. See “Data Set Macros” on page 5 for information about specifying the name of the data set. The default value is WORK.LSAFGETPFUSERTASKPROPERTIES.

The data set contains a row for each property and columns with the following names. The data set is not sorted.

Column Name	Description
taskId	The task identifier.
propertyName	The name of the property.
propertyDisplayName	The display name of the property.
propertyType	The type of the property.
propertyValue	The value of the property.
isReadOnly	Indicates whether the property is Read-Only.
isExtended	Indicates whether the property is an extended attribute.
isRequired	Indicates whether the property value is required.

%LSAF_GETPFUSERTASKSTATUS Macro

Gets the status of a process flow user task.

Category: ProcessFlowTask Module

Note: This macro sets the standard macro variables and _lsafpfusertaskstatus_, which is the status.

Syntax

%LSAF_GETPFUSERTASKSTATUS(LSAF_TASKID=*task-id*);

Required Argument

LSAF_TASKID=*task-id*
The identifier of the task. To get the identifier, call %LSAF_GETPFTASKSBYQUERY.

Details

Sets the value of the macro variable _lsafpfusertaskstatus_ to the status. See "Value Macros" on page 5 for information about value macros.

%LSAF_SETPFUSERTASKCOMPLETIONDATA Macro

Updates the process flow user task completion data using a SAS data set as input.

Category: ProcessFlowTask Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_SETPFUSERTASKCOMPLETIONDATA(LSAF_TASKID=*task-id*<, SAS_DSNAME=*name*>);

Required Argument

LSAF_TASKID=*task-id*
The identifier of the task. To get the identifier, call %LSAF_GETPFTASKSBYQUERY.

Optional Argument

SAS_DSNAME=*name*
The name of the data set that contains the completion data values to set on the task. See "Data Set Macros" on page 5 for information about specifying the name of the data set.

The data set must contain at least the following columns. Additional variables are ignored.

Column Name	Description
dataID	The identifier of the process flow completion data element.

Column Name	Description
dataName	The name of the process flow data completion element.

Details

Updates the process flow user task completion data using a SAS data set as input.

Use the output data set from the %LSAF_GETPFUSERTASKCOMPLETIONDATA macro to determine the name and current values of the process flow properties and to use as input to the macro. Rows in which the dataId matches an existing completion data property are used to attempt to update the property. If the specified value is invalid for the data type, the update is stopped and the appropriate error message is printed to the SAS log file.

See "Format Date Values" on page 4 for information about date formats.

%LSAF_STARTPFUSERTASK Macro

Starts a user task. A task must be assigned to the current user before it can be started.

Category: ProcessFlowTask Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_STARTPFUSERTASK(LSAF_TASKID=*task-id*);

Required Argument

LSAF_TASKID=*task-id*
The identifier of the task. To get the identifier, call %LSAF_GETPFTASKSBYQUERY.

%LSAF_UPDATEPFUSERTASKCANDIDATES Macro

Updates the process flow user task candidates using a SAS data set as input.

Category: ProcessFlowTask Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_UPDATEPFUSERTASKCANDIDATES(LSAF_TASKID=*task-id*<, SAS_DSNAME=*name*>);

Required Argument

LSAF_TASKID=*task-id*
The identifier of the task. To get the identifier, call %LSAF_GETPFTASKSBYQUERY.

Optional Argument

SAS_DSNAME=*name*
The name of the input data set that contains the details for the process flow user task candidates. See "Data Set Macros" on page 5 for information about specifying the name of the data set.

The following columns are required for processing, all other columns are ignored:

Column Name	Description
candidateType	The case-insensitive type of the candidate. Valid values: ■ User ■ Group
candidateID	The case-sensitive identifier of the flow candidate, either a user group name or user identifier.
grpSrcCtxt	The full path of the context in which the user group is defined. This value is ignored when candidateType is USER.

Details

Updates the process flow user task candidates using a SAS data set as input. The output data set from the %LSAF_GETPFUSERTASKCANDIDATES macro can be used as input to the macro. The process flow user task candidates that are represented in the input data set replace the list of candidates that is currently saved on the specified process flow user task. To clear all candidates from the task, call %LSAF_CLEARPFUSERTASKCANDIDATES.

%LSAF_UPDATEPFUSERTASKPROPERTIES Macro

Updates the process flow user task properties using a SAS data set as input.

Category: ProcessFlowTask Module

Note: This macro sets the standard macro variables.

Syntax

%LSAF_UPDATEPFUSERTASKPROPERTIES(LSAF_TASKID=*task-id*, SAS_DSNAME=*name*);

Required Arguments

LSAF_TASKID=*task-id*

The identifier of the process flow user task. To get the identifier, call %LSAF_GETPFTASKSBYQUERY.

SAS_DSNAME=*name*

The name of the input data set that contains the details for the process flow user task properties. See “Data Set Macros” on page 5 for information about specifying the name of the data set.

The following columns are required for processing, all other columns are ignored:

Column Name	Description
propertyName	The name of the process flow user task property.
propertyValue	The value for the process flow user task property.

Details

Updates the process flow user task properties using a SAS data set as input. The output data set from the %LSAF_GETPFUSERTASKPROPERTIES macro can be used as input to the macro.