Java and JavaScript Public APIs

Gargantua 8 Public API

This document is the property of SIATEL.
All rights reserved to SIATEL.

Table of contents

Glossary of terms 13

Introduction 14

Object types 15

Object identification 16

Generalities 18

Interaction with the Gargantua server 18

Data transfer objects 18

Services and commands 19

How to use the public API 20

Java API Reference 21

Package com.siatel.defs 21

Interfaces 21

Archive 21

Commented 21

Dated 22

DatedEx 23

Deletable 24

Deployable 25

DeployableEx 26

Described 27

Expirable 28

Flagged 30

Identified 30

Labeled 32

Lifecycled 33

Managed 33

Modified 34

Numbered 35

Owned 36

Packageable 37

Parsable 38

Presented 39

Prioritized 39
Realized 40
RealizedEx 41
Referenced 42
Replicable 43
Sampled 43
Securable 44
Shared 46
Tagged 48
Timeboxed 49
TimeboxedEx 50
Timed 51
UUIDEnabled 52
Version 53
Versioned 53
Viewable 54
ViewableEx 58
Package com.siatel.domain 59
Interfaces 59
ArchiveComment 59
ArchiveDocument 60
ArchiveDocumentVersion 60
ArchiveFolder 61
ArchiveNote 62
Assignment 62
Attribute 64
Certificate 64
ClassificationElement 65
ClassificationTreeElement 65
Comment 66
Database 66
Dependency 67
DistributionListEntry 68
Document 68
DocumentVersion 69
ExpirableDownloadLink 69
ExtendedProperty 70
Folder 71
Form 71
Grid 72
Group 72
HierarchyCreator 73

InterventionReport ... 73
IsJob ... 74
Layouted ... 74
Lifecycle ... 75
Manager ... 75
ModelOCR ... 75
Note ... 76
ObjectExtendedProperty ... 77
Package ... 77
Planner ... 78
PlannerDay ... 79
Portal ... 79
Process ... 80
ProcessDefinition ... 80
ProcessDefinitionVersion ... 81
Resource ... 82
Role ... 82
Rule ... 82
Script ... 83
Storage ... 83
Task ... 83
Type ... 83
User ... 84
UserDelegation ... 84
View ... 85
ViewElement ... 86
ViewElementVersion ... 88
ViewVersion ... 90
Classes ... 90
AttributeModifier ... 90
AttributeProp ... 91
Attributes ... 91
FilterCriterion ... 92
Flag ... 93
Prefix ... 139
Siatel_0 ... 172
Siatel_1 ... 173
Siatel_1_1 ... 173
Siatel_1_2 ... 174
Siatel_1_3 ... 175
Siatel_1_4 ... 175
Siatel_2 ... 176

Siatel_3 ..... *177*
Tag *178*
UniqueCriterion *179*
ValueToken *181*
Package com.siatel.framework.service.admin 181
Classes 181
AdministrationLibrary *181*
Package com.siatel.framework.service.audit 208
Classes 208
AuditLibrary *208*
Package com.siatel.framework.service.auth 211
Classes 211
AuthenticationLibrary *211*
Package com.siatel.framework.service.design 214
Classes 215
DesignLibrary *215*
Package com.siatel.framework.service.flow 218
Classes 218
FlowLibrary *218*
Package com.siatel.framework.service.ged 223
Classes 223
GedLibrary *223*
Package com.siatel.framework.service.misc 260
Classes 260
MiscLibrary *260*
Package com.siatel.framework.service.plugins 268
Classes 269
PluginsLibrary *269*
Package com.siatel.framework.service.search 270
Classes 270
SearchLibrary *270*
Package com.siatel.framework.service.statistics 280
Classes 280
StatisticsLibrary *280*
Package com.siatel.proxy.http 281
Classes 281
SiatelClient *281*

JAVA API EXAMPLES 339
The Toolbox object 339
Managing the connection cache 340

Getting server information 341
Opening a user connection to the server 342
Closing a user connection 343
Reading user information by label 344
Update a user 345
List users 346
Create a new user 346
Assign a user to a group 347
Complete Toolbox class source code 348
Using the Toolbox class 358

GARGANTUA API FOR FORMS 360

JavaScript functions for form elements 361
gfxGetAttrIdByLabel 361
gfxGetAttrLabelById 362
getElementValue 363
getElementKey 364
gfxVal 366
gfxEnable 367
gfxDisable 368
gfxVisible 369
getElementValueEx 370
setElementValue 372
setElementKey 373
getElementColor 374
setElementValueEx 375
setElementColor 376
setElementColor2 377
setElementTitle 378
setElementVisible 379
setElementVisible2 380
setElementEdit 381
setElementEdit2 382
setElementEnabled 383
isElementEnabled 384
disabledEditor 385
enabledEditor 386
form_getValues 387
form_getAttributeType 388
form_getAttributeTypeId 389
form_getPage 390

Miscellaneous JavaScript functions 391
form_getVersion 391
form_lang 392
form_getDisplayMode 393
form_enableGroup 394
form_setGroupVisible 395
form_expandSection 396
form_collapseSection 397
form_toggleSection 398
form_toggleAllSections 399
form_isSectionExpanded 400
form_addSectionIcon 401
form_toggleSectionIcon 402
form_removeSectionIcon 403
form_setSectionIconStyle 404
form_stickySection 405
form_sectionsToTabs 406
form_sectionGroupsToTabs 407
form_refresh 409
form_escapeHtml 410
form_refreshViewer 411
form_cleanupHtml 412
form_log 413
form_info 414
form_warn 415
form_error 416
form_waitstart 417
form_waitstop 418
form_formatDate 419
form_parseDate 420
form_formatUserName 421
form_hasDocuments 422
form_getDocuments 423
form_filterList 424
form_setDocTemplateOptions 425
JavaScript functions for customisizing the user interface 426
flashElement 427
form_displayButtonBar 428
form_displayDocumentListToggle 429
form_displayDocumentList 430
form_displaySelectorArea 431
form_displayDefaultButtonBar 432

form_setCustomButtonEnabled 433
form_displayCustomButtonBar 434
form_setCustomButtonVisible 435
form_setCustomButtonHidden 436
form_refreshDocumentList 437
form_displayDocumentListOnReturn 438
form_processAreaVerticalMaximize 439
form_processAreaResizeToggle 440
form_processAreaResize 441
form_displayAlert 442
form_closeAlert 443
form_displayDialog 444
form_displayCustomDialog 446
form_closeDialog 450
form_getDialogControl 451
form_getDialogElement 451
Button 451
Buttons 451
form_loadScript 452
form_loadScripts 453
form_loadStylesheet 454
form_loadStylesheets 455
JavaScript functions for triggering actions 456
resetElement 457
runServerScript 458
runServerScript2 459
form_runScript 460
form_apiRequest 461
form_scanDocuments 462
form_scanDocumentsExt 463
form_scanDocumentsExt2 464
form_scanDocumentsExt3 465
form_createDocument 466
form_addDocuments 467
form_addDocumentsDirect 468
form_certifyDocument 469
form_editDocument 470
form_viewDocument 471
form_signDocument 473
form_viewDocuments 474
form_downloadDocuments 474
form_editObject 475

form_save ... 475
form_saveImmediate ... 476
form_saveDraft ... 477
form_saveDraftImmediate ... 478
form_saveTaskImmediate ... 479
form_saveTask ... 480
form_cancelDraft ... 481
form_cancel ... 481
form_finish ... 481
form_startProcess ... 482
form_cancelTask ... 483
form_finishTask ... 484
form_executeSearch ... 485
form_escalateTask ... 486
doValidateForm ... 487
handler_onSwitchPage ... 488
form_switchForm ... 489
form_gotoTasks ... 490
form_invokePlugin ... 491
Advanced JavaScript functions ... 492
form_newControl ... 493
form_destroyControl ... 494
form_getControl ... 495
form_getElement ... 496
getElementKeyAndValue ... 497
setElementKeyAndValue ... 498
form_saveGlobalGrids ... 499
form_validateLatency ... 500
form_getValidateLatency ... 501
form_validateLatencyOffset ... 502
form_validateMode ... 503
form_getValidateMode ... 504
form_getValidationMessage ... 505
form_setSubmitOnEnter ... 506
Controls ... 507
General behavior ... 507
Simple input fields ... 515
Lists ... 526
Grid ... 538
Other input fields ... 598
Form callback functions ... 599
fun_form_pre_load ... 600

fun_form_load 601
fun_form_post_load 602
fun_form_displayControls 603
fun_form_pre_loadattrs 604
fun_form_post_loadattrs 605
fun_form_pending_validate 606
fun_form_pre_validateattrs 607
fun_form_pre_validatereponse 608
fun_form_validate 610
fun_form_post_validatereponse 611
fun_form_get_validationmessage 612
fun_form_pre_submitform 614
fun_form_confirm_submitform 615
fun_form_post_submitform 617
fun_form_pre_sendtask 618
fun_form_get_editnative_options 619
fun_form_refresh_docs 620
fun_form_pre_escalatetask 621
fun_form_get_ocr_attributes 622
fun_form_set_ocr_attributes 623
fun_form_post_sign_documents 624
fun_form_reset 625
Services available through AJAX requests 626
ajax_syncServerRequestXML 626
/api/search/attr?request 627
/api/get?request 628
Information accessible from the form 629
Deprecated APIs 629

Glossary of terms

Gargantua server

The Gargantua server is a software application that implements document management and workflow functionalities.

Service

A service is a component of the Gargantua server that responds to commands.

Command

A request being made to a service within the Gargantua server; a command is identified by a command code; parameters may be sent along with the command code and a result may be expected; the documentation for each individual command lists the command code, the input parameters, the outcome, exceptions thrown and preconditions, each being subject to availability within the given command.

Command code

An integer value that identifies a Gargantua server command

Command parameters

A data structure given on input when executing a Gargantua server command; parameters are always built using a data transfer object (DTO), which is implemented as a string-keyed map of objects, the key being the parameter name

Command result

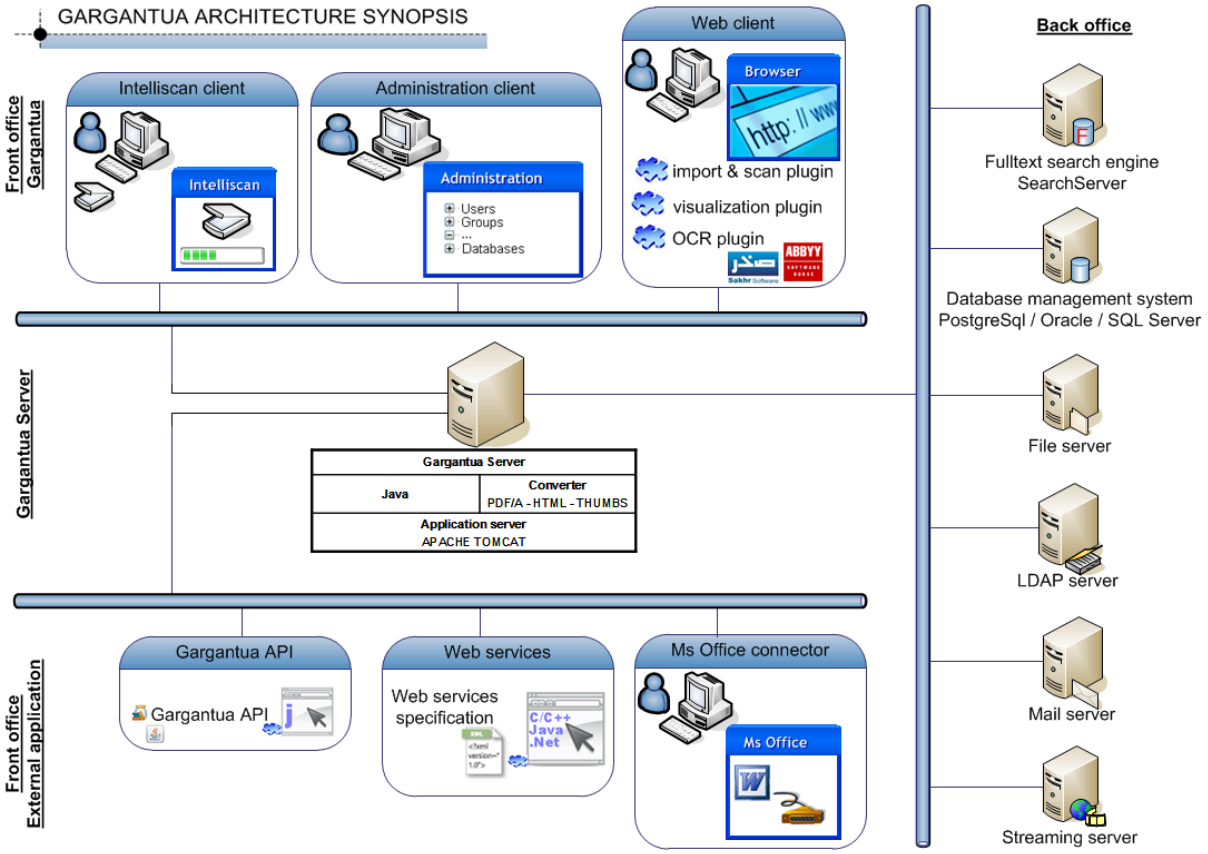
A data structure that is the result of executing a Gargantua server command.

Data transfer object

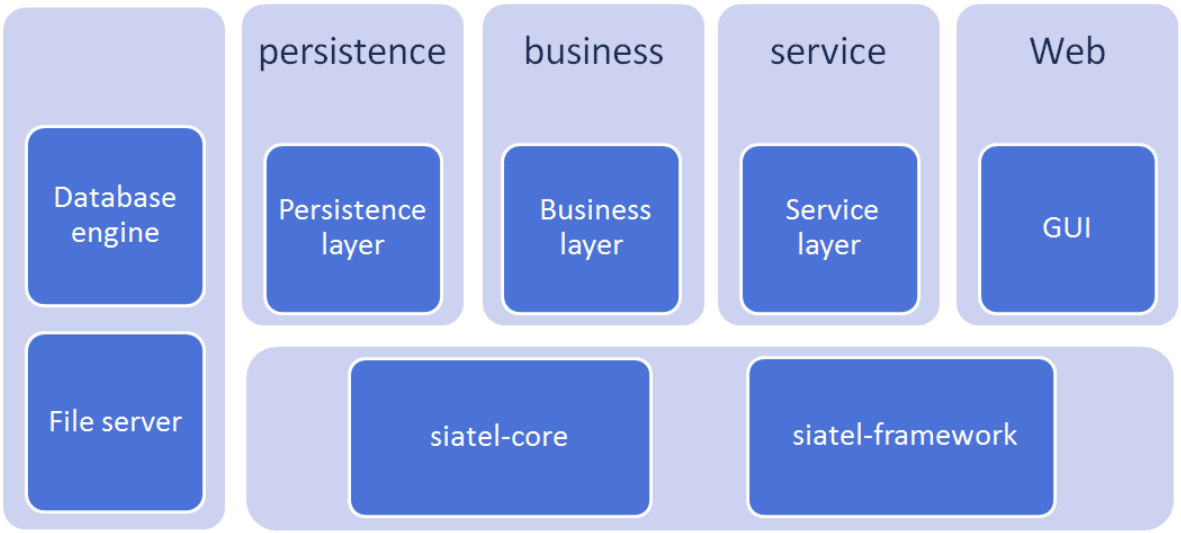
A map data structure used to pass parameters towards the Gargantua server or retrieve the result of a command.

Introduction

This document is provided as a reference to the public API of the Gargantua 7 application. The general architecture of a Gargantua system is shown in the image below.



Internally, Gargantua 7 follows a layered architectural approach, as described in the following schema.



Object types

In Gargantua 7, there are three major object types

System objects

The system objects include, for instance, users, groups, their assignments, databases, etc. They are needed to manage the Gargantua system; as a visual clue, these are the objects that are displayed on the first level of the administration tree in the Gargantua Administrator.

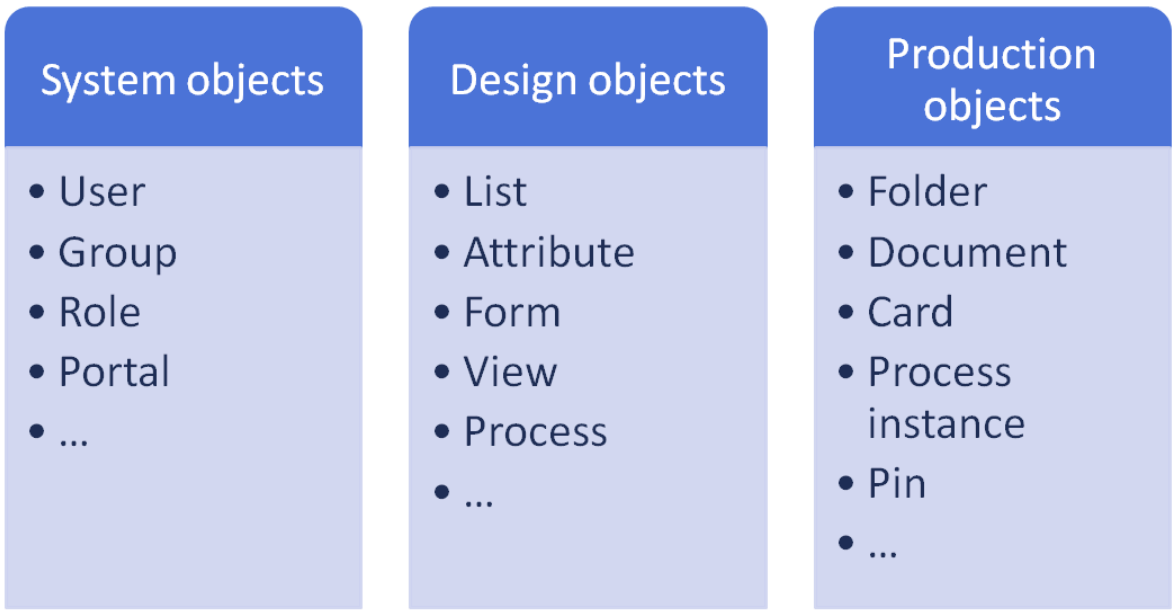
Design objects

Design objects are managed on a per-database basis. These objects are the ones that govern the use of a specific Gargantua database; as a visual clue, these are the objects that are displayed as the first level objects below the database object in the Gargantua Administrator

Production objects

Production objects are the objects that users manipulate directly using the user interface, either web or custom tools like the IntelliScan.

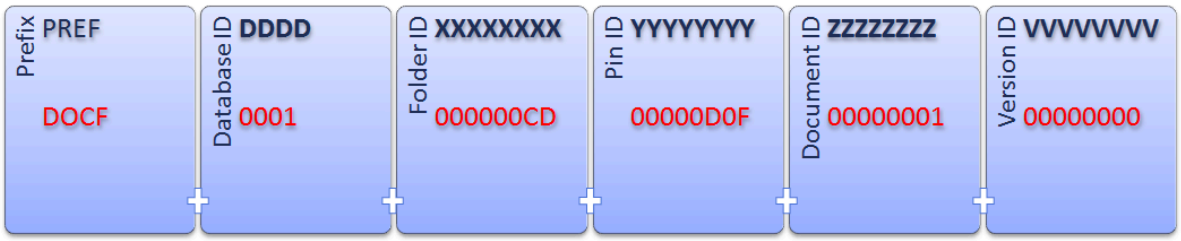
The schema below outlines these object types.



One can also define the design and production objects types as follows : design objects are templates used to create actual objects, which are the production objects. Design objects are like class definitions in high level languages like C++ or Java, while production objects are instances of these classes.

Object identification

To help identify objects, in Gargantua a fixed length string is used. Any object's ID is a 40 character sequence; it is always 40 characters long, and this fact can be used to validate IDs. Each object ID can be broken down in several elements, as outlined in the schema below.



Size and type :

- **PREF** : 4 letter code
- **DDDD** : 4 hex digits representing the Gargantua database ID, zero for system objects
- **XXXXXXXX** : 8 hex digits
- **YYYYYYYY** : 8 hex digits
- **ZZZZZZZZ** : 8 hex digits
- **VVVVVVVV** : 8 hex digits
- All unused part of the ID must be set to zero, as it is ignored.

There are always 6 components of the ID, and their meaning vary by object type. However, the first two elements are common to all objects :

Prefix

The prefix helps tell objects apart; it is a unique 4 letter code that is assigtned to each object type. It uniquely identifies the object type. Below are a few examples of object prefixes, grouped by object type. For the full list of object types please review the API reference.

System objects

- User - USER
- Group - GROU
- Role - ROLE
- Portal - PORT
- ...

Design objects

- List - DFTY
- Attribute - DFAT
- Form - DFFR
- View - DFVW
- Process - DFPR
- ...

Production objects

- Folder - FLDR
- Document - DOCF
- Card - FCHF
- Process instance - PROC
- Pin – PINF
- ...

Database ID

The database ID tells in which database the object resides. For system objects, this is always set to zero.

Please note that IDs are always 40 characters long. First 4 characters are alphanumeric; currently only letters are used.

All ID parts except the object prefix are stored in hexadecimal notation (digits 0 through 9 and letters 'A' through 'F')

All ID parts that are not used for a given object must me set to zero; failing to do so may result in improper handling of objects by the core API; please carefully review the ID structure of any object before using it, as in some cases setting a certain part to a non-zero value may make the ID point to a different object.

Generalities

Interaction with the Gargantua server

Interaction with the Gargantua server is done through commands; one such very simple command is related to authentication.

The client application identifies itself to the Gargantua server by providing an username and a password, and receives back a connection token

Data transfer objects

A data transfer object (or DTO) is an object that carries data between processes. The original motivation for its use is that communication between processes is usually done resorting to remote interfaces (e.g. web services), where each call is an expensive operation. Because the majority of the cost of each call is related to the round-trip time between the client and the server, one way of reducing the number of calls is to use an object (the DTO) that aggregates the data that would have been transferred by the several calls, but that is served by one call only.

One important side effect of the use of DTO objects is that only one parameter is passed, and thus the signature of a call does not change over time.

Another important side effect is that implementation of optional or mutually exclusive parameters is very easy to implement.

The difference between data transfer objects and business objects or data access objects is that a DTO does not have any behavior except for storage and retrieval of its own data (mutators and accessors). DTOs are simple objects that should not contain any business logic.

In the Gargantua world, DTOs are implemented as string-keyed maps of objects. Also, a number of helper classes and methods are provided with the Gargantua API, which leads to easy to follow client-side code, like in the example below.

```
List<Document> docList = (List<Document>)clientProxy.genericGedInvoke(
          GedLibrary.CMD_ENUMERATE_DOCUMENTS,
            DTOFactory.newDTO(
               K.CRITERIA, criteria,
                K.FLAGS, Flag.DOCUMENT_FULL,
                K.OFFSET, 0,
                K.COUNT, -1))
            .get(K.LIST);
```

The code above will retrieve a full list of complete document objects (offset = 0, count = -1) with some criteria

from the server;

Services and commands

How to use the public API

Java API Reference

Package com.siatel.defs

This package contains interface definitions that group domain object properties.

Interfaces

Archive

```
public interface Archive
```

Accessor wrapper interface for archive enabled objects (ArchiveFolder, ArchiveDocument, ArchiveDocumentVersion, ArchiveNote).

Since :

7.5

Method summary :

Modifier and type	Method
`public java.util.Date`	`getDateArchive ()`
`public void`	`setDateArchive (java.util.Date d)`

Method detail :

```
public Date getDateArchive ()
```

Accessor.

Returns :

the date when object was archived

```
public void setDateArchive (Date d)
```

Accessor.

Parameters :

`d` - the value to set for the date when object was archived.

Commented

```
public interface Commented
```

Commented objects are objects that have a comment; this comment has no other purpose than help the administrator or users understand the purpose of the object.

Since :

7.5

See also :

Labeled

Author :

stefanu

Method summary :

Modifier and type	Method
public java.lang.String	getComment ()
public void	setComment (java.lang.String comment)

Method detail :

```
public String getComment ()
```

Returns :

the comment for the specified object

```
public void setComment (String comment)
```

Parameters :

comment - the new comment for the specified object

DATED

```
public interface Dated
```

Dated interface is implemented by all objects that can have a creation date and a modification date; the Dated objects can be traced.

Since :

7.0

Author :

stefanu

See also :

DatedEx

Method summary :

Modifier and type	Method
public java.util.Date	getDateC ()
public void	setDateC (java.util.Date dateC)
public java.util.Date	getDateM ()
public void	setDateM (java.util.Date dateM)

Method detail :

```
public Date getDateC ()
```

Returns :

the creation date and time of the specified object

```
public void setDateC (Date dateC)
```

Parameters :

dateC - the new creation date and time for the specified object

```
public Date getDateM ()
```

Returns :

the last modification date and time of the specified object

```
public void setDateM (Date dateM)
```

Parameters :

dateM - the new last modification date and time of the specified object

DatedEx

```
public interface DatedEx
```

DatexEx objects keep track of the last access date and time, as an extension to the Dated objects that keep track of the creation and last modification date and time

All Superinterfaces :

Dated

Since :

7.0

See also :

Dated

Author :

stefanu

Method summary :

Modifier and type	Method
public java.util.Date	getDateA ()
public void	setDateA (java.util.Date dateA)

Method detail :

```
public Date getDateA ()
```

Returns :

the last access date and time for the specified object

```
public void setDateA (Date dateA)
```

Parameters :

dateA - the new last access date and time for the specified object

DELETABLE

public interface Deletable

The Deletable interface is used by all objects that have a flag for delete.

Since :

7.9

Author :

ying

Method summary :

Modifier and type	Method
public boolean	getDeleted ()
public void	setDeleted (boolean deleted)
public java.util.Date	getDateDel ()
public void	setDateDel (java.util.Date dateDel)
public int	getDeleteBy ()
public void	setDeleteBy (int deleteBy)

Method detail :

```
public boolean getDeleted ()
```

Returns :

true if object is in trash

```
public void setDeleted (boolean deleted)
```

Parameters :

`deleted` - the new 'deleted to trash' status for the object

```
public Date getDateDel ()
```

Returns :

the date and time when the object was moved to trash

```
public void setDateDel (Date dateDel)
```

Parameters :

`dateDel` - the date and time when the object was moved to trash

```
public int getDeleteBy ()
```

Returns :

numeric id of user who moved the object yp trash

```
public void setDeleteBy (int deleteBy)
```

Parameters :

`deleteBy` - the numeric id of user who moved the object to trash

DEPLOYABLE

`public interface Deployable`

Deployable objects are design objects; most of the design objects are not meant to be put in production immediately; this means that creating such an object may require significant time and effort from the system administrator, and the system state may become unstable if such objects are made available for the users immediately after they've been saved into the database. Such objects are, for instance, processes. processes are complex objects that must not be put into production immediately; the processes are created (which may require even more than one week's work from the system administrator), then 'deployed' once the process is ready to work as intended. Users have access to Deployable objects only when then are deployed; otherwise, these objects are only available in the administrative tools. Some objects may require validation or compilation as they are deployed. Validation errors will block the deploy command and leave the object in an undeployed state.

Since :

7.0

See also :

DeployableEx

Author :

stefanu

Method summary :

Modifier and type	Method
`public boolean`	`isDeployed ()`
`public void`	`setDeployed (boolean deployed)`

Method detail :

```
public boolean isDeployed ()
```

Returns :

the deployment status of the specified object

```
public void setDeployed (boolean deployed)
```

Parameters :

`deployed` - the new deployment status for the specified object

DeployableEx

```
public interface DeployableEx
```

DeployableEx objects are objects that take an alternate form when deployed; an example are the form objects; when deployed, these objects are 'compiled' into JSP pages that can be displayed by the JSP engine. When DeployableEx objects are deployed, a secondary file is created and stored for use in production.

All Superinterfaces :

Deployable

Since :

7.0

See also :

Deployable

Author :

stefanu

Method summary :

Modifier and type	Method
public java.lang.String	getDeploymentId ()
public void	setDeploymentId (java.lang.String deploymentId)
public long	getDeploymentTimestamp ()
public void	setDeploymentTimestamp (long millis)

Method detail :

```
public String getDeploymentId ()
```

Returns :

the deployment ID for the specified object

```
public void setDeploymentId (String deploymentId)
```

Parameters :

`deploymentId` - the new deployment ID for the specified object

```
public long getDeploymentTimestamp ()
```

Returns :

the deployment time stamp for the specified object; this time stamp is required by the deployment operation to assess the need of transformation, checking that the deployed object has been modified since last deployment command

```
public void setDeploymentTimestamp (long millis)
```

Parameters :

`millis` - the new deployment time stamp for the specified object

DESCRIBED

```
public interface Described
```

Described objects are objects that have a description; this description has no other purpose than help the administrator or users understand the purpose of the object.

Descriptions may be displayed as tooltips by the user interface.

Since :

7.0

See also :

Labeled

Author :

stefanu

Method summary :

Modifier and type	Method
public java.lang.String	getDescription ()
public void	setDescription (java.lang.String description)

Method detail :

```
public String getDescription ()
```

Returns :

the description for the specified object

```
public void setDescription (String description)
```

Parameters :

description - the new description for the specified object

EXPIRABLE

public interface Expirable

Expirable objects are part of the workflow related objects; these objects have an due date and an expiration date; when the due date is reached, the object is not yet expired, but it will be signaled by the user interface differently; the due date and the expiration date are needed by the task and process state transition algorithm

Since :

7.0

Author :

stefanu

Method summary :

Modifier and type	Method

public java.util.Date	getDateD ()
public void	setDateD (java.util.Date dateD)
public boolean	isOverdue ()
public void	setOverdue (boolean overdue)
public void	setDateX (java.util.Date dateX)
public boolean	isExpired ()
public void	setExpired (boolean expired)

Method detail :

```
public Date getDateD ()
```

Returns :

the due date of the specified object

```
public void setDateD (Date dateD)
```

Parameters :

`dateD` - the new due date for the specified object

```
public boolean isOverdue ()
```

Returns :

the overdue status of the specified object

```
public void setOverdue (boolean overdue)
```

Parameters :

`overdue` - the new overdue status of the specified object

```
public void setDateX (Date dateX)
```

Parameters :

`dateX` - the new expiration date for the specified object

```
public boolean isExpired ()
```

Returns :

the expiration status for the specified object

```
public void setExpired (boolean expired)
```

Parameters :

`expired` - the new expiration status for the specified object

Flagged

`public interface Flagged`

Flagged objects are objects that may have a set of flags associated with them. Flags are defined by the Flag constants, and are ORed constants

Since :

7.0

See also :

Flag

Author :

stefanu

Method summary :

Modifier and type	Method
`public long`	`getFlags ()`
`public void`	`setFlags (long flags)`

Method detail :

```
public long getFlags ()
```

Returns :

the flags for the specified object

```
public void setFlags (long flags)
```

Parameters :

`flags` - the new flags for the specified object

Identified

`public interface Identified`

The Identified interface is the very heart of the the Gargantua system. It sets an identification system that all objects handled by the system must follow.

The Identified interface is implemented by almost all objects within the Gargantua system, as all the objects handled by the server must have a unique identifier. here are two types of identifiers for each type of object : a numeric identifier and a string identifier.

The numeric identifier is used to uniquely identify objects of the same type.

Because unique numeric identifiers cannot globally and uniquely identify objects, a global identifier system was designed. the purpose of such an identifying system is, first, to uniquely identify objects throughout the system, and secondly, to be able to get a maximum amount of information about an object position within the database just be examining the object identification string. These string identifiers have always the same character length, thus enabling consistent handling, and always hold al least the object type, the database ID of the database that stores the object and the object numeric identification; additionally, some objects contain other numeric identifications such as, for a document, the numeric container folder identification.

The structure of the string identification is specific to each object and is explained for each and every one of the objects within the object's documentation

As of release 7.4 this interface extends Parsable which wraps the 'prefix' property and the 'Parsable#parse' method. These methods have been pulled up from previous version of `Identified`.

All Superinterfaces :

Parsable

Since :

7.0

See also :

IdFormat , Prefix

Author :

stefanu

Method summary :

Modifier and type	Method
public java.lang.String	getId ()
public void	setId (java.lang.String id)
public int	getDbId ()
public void	setDbId (int id)
public int	getNumeric (int i)
public void	setNumeric (int i, int numeric)

Method detail :

```
public String getId ()
```

Returns :

the string identification of the specified object

```
public void setId (String id)
```

Parameters :

`id` - the new string identification for the object

```
public int getDbId ()
```

Returns :

the numeric database ID that stores the specified object

```
public void setDbId (int id)
```

Parameters :

`id` - the new numeric database ID that stores the specified object

```
public int getNumeric (int i)
```

Parameters :

`i` - the numeric ID index

Returns :

the numeric ID at the specified index

```
public void setNumeric (int i, int numeric)
```

Parameters :

`i` - the numeric ID index

`numeric` - the new numeric ID for the specified index

Labeled

`public interface Labeled`

The Labeled interface is used by all objects that have a label, which is almost every public Gargantua object.

Since :

7.0

See also :

Described

Author :

stefanu

Method summary :

Modifier and type	Method
public java.lang.String	getLabel ()
public void	setLabel (java.lang.String label)

Method detail :

```
public String getLabel ()
```

Returns :

the label for the specified object

```
public void setLabel (String label)
```

Parameters :

`label` - the new label for the specified object

LIFECYCLED

```
public interface Lifecycled
```

Lifecycled objects are objects that may change automatically over time. The lifecycle of such objects may be customized by the system administrator, as well as the events that trigger a lifecycle state change. For instance, objects may be automatically archived after a specified period of inactivity or documents may be published if a certain attribute has changed to a specific value.

This interface is obsolete as of 7.8

All Superinterfaces :

Timed

Deprecated :

Author :

stefanu

MANAGED

```
public interface Managed
```

Managed objects are objects that have an associated user that acts as a manager. Such objects are, for instance, processes.

Since :

7.0

Author :

stefanu

Method summary :

Modifier and type	Method
public java.lang.String	getManagerId ()
public void	setManagerId (java.lang.String managerId)

Method detail :

```
public String getManagerId ()
```

Returns :

the manager identification for the specified object

```
public void setManagerId (String managerId)
```

Parameters :

managerId - the new manager identification for the specified object

MODIFIED

public interface Modified

Modified objects are objects that keep track of the user identification that modified them; (viewable objects)

ModifiedBy are always physical users, never groups or roles.

Since :

7.10

Author :

stefanu

Method summary :

Modifier and type	Method
public java.lang.String	getModifiedBy ()
public void	setModifiedBy (java.lang.String modifiedBy)

public int	getModifiedByNumeric ()
public void	setModifiedByNumeric (int modifiedBy)
public java.util.Date	getDateM ()
public void	setDateM (java.util.Date dateM)

Method detail :

```
public String getModifiedBy ()
```

Returns :

the modifiedBy the user who modify the specified object

```
public void setModifiedBy (String modifiedBy)
```

Parameters :

`modifiedBy` - the user who modify the specified object

```
public int getModifiedByNumeric ()
```

Returns :

the numeric of the modifiedById

```
public void setModifiedByNumeric (int modifiedBy)
```

Parameters :

`the` - numeric of the modifiedById

```
public Date getDateM ()
```

Returns :

the last modification date and time of the specified object

```
public void setDateM (Date dateM)
```

Parameters :

`dateM` - the new last modification date and time of the specified object

Numbered

`public interface Numbered`

The Numbered interface is used by all objects that have an index within a certain context (for example any object version within object history).

Since :

7.5

Author :

andreea

Method summary :

Modifier and type	Method
public int	getNumber ()
public void	setNumber (int number)

Method detail :

```
public int getNumber ()
```

Returns :

the index of the specified object

```
public void setNumber (int number)
```

Parameters :

index - the new index for the specified object

OWNED

```
public interface Owned
```

Owned objects are objects that keep track of the user identification that created them; most objects of the Gargantua system are owned, since the security mechanism uses owners to give them specific rights.

Owners are always physical users, never groups or roles.

Since :

7.0

Author :

stefanu

Method summary :

Modifier and type	Method
public java.lang.String	getOwnerId ()
public void	setOwnerId (java.lang.String ownerId)
public int	getOwnerIdNumeric ()
public void	setOwnerIdNumeric (int ownerId)

Method detail :

```
public String getOwnerId ()
```

Returns :

the owner identification for the specified object

```
public void setOwnerId (String ownerId)
```

Parameters :

`ownerId` - the new owner identification for the specified object

```
public int getOwnerIdNumeric ()
```

Returns :

the numeric owner identification for the specified object

```
public void setOwnerIdNumeric (int ownerId)
```

Parameters :

`ownerId` - the new numeric owner identification for the specified object

PACKAGEABLE

`public interface Packageable`

The Packageable interface is used by all objects that can be grouped in a package.

Since :

7.10

See also :

Described

Author :

danieln

Method summary :

Modifier and type	Method
`public java.util.List`	`getPackages ()`
`public void`	`setPackages (java.util.List packages)`

Method detail :

```
public List getPackages ()
```

Obtains the packages ids that the object is part of

```
public void setPackages (List packages)
```

Sets the ids of packages that the object is part of

Parameters :

`packages` - the list of packages

PARSABLE

```
public interface Parsable
```

Wrapper for parsable objects requirements.

Since :

7.4

Author :

elf

Method summary :

Modifier and type	Method
public java.lang.String	getPrefix ()
public void	setPrefix (java.lang.String prefix)
public void	parse (java.lang.String tokens)

Method detail :

```
public String getPrefix ()
```

Returns :

returns the prefix (type) of the specified object

```
public void setPrefix (String prefix)
```

Parameters :

`prefix` - the new prefix (type) for the specified object

```
public void parse (String tokens)
```

Initialize using string parsed tokens.

Parameters :

`tokens` - string parsed tokens used for initialization when object initialized from string

PRESENTED

`public interface Presented`

Presented objects are objects that have presentation options. They specify which attributes must be displayed

Since :

7.0

See also :

SearchTemplate

Author :

alexis

Method summary :

Modifier and type	Method
public java.lang.String	getPresentationFileId ()
public void	setPresentationFileId (java.lang.String fileId)

Method detail :

```
public String getPresentationFileId ()
```

Returns :

the presentation file ID for the specified object

```
public void setPresentationFileId (String fileId)
```

Parameters :

`fileId` - the presentation file ID for the specified object

PRIORITIZED

`public interface Prioritized`

Prioritized objects are objects that may be sorted according to a user set priority

Since :

7.0

See also :

Priority

Author :

stefanu

Method summary :

Modifier and type	Method
public int	getPriority ()
public void	setPriority (int priority)

Method detail :

```
public int getPriority ()
```

Returns :

the priority for the specified object

```
public void setPriority (int priority)
```

Parameters :

priority - the new priority for the specified object

REALIZED

```
public interface Realized
```

Realized objects are objects that have an associated file; it is the responsibility of the business code to store and retrieve the files from the Gargantua file server accordingly.

Since :

7.0

See also :

RealizedEx , com.siatel.framework.service.storage.StorageLibrary

Author :

stefanu

Method summary :

Modifier and type	Method
public java.lang.String	getFileId ()
public void	setFileId (java.lang.String fileId)

Method detail :

```
public String getFileId ()
```

Returns :

the file ID for the specified object

```
public void setFileId (String fileId)
```

Parameters :

`fileId` - the new file ID for the specified object

RealizedEx

```
public interface RealizedEx
```

RealizedEx objects are objects than need to keep track of the file size and file date that is stored within the Gargantua file server. In order to accelerate access to this information, these objects store this information internally, rather than retrieving it from the file server on a need-to-access basis. It is the responsibility of the business code to keep this information up to date and consistent with the actual file information.

All Superinterfaces :

Realized

Since :

7.0

See also :

Realized , com.siatel.framework.service.storage.StorageLibrary

Author :

stefanu

Method summary :

Modifier and type	Method
public long	getFileSize ()
public void	setFileSize (long size)
public java.lang.String	getFileType ()
public void	setFileType (java.lang.String fileType)

Method detail :

```
public long getFileSize ()
```

Returns :

the file size for the specified object

```
public void setFileSize (long size)
```

Parameters :

`size` - new the file size for the specified object

```
public String getFileType ()
```

Returns :

the file type for the specified object

```
public void setFileType (String fileType)
```

Parameters :

`fileType` - the new file type for the specified object

REFERENCED

```
public interface Referenced
```

Referenced objects are objects that must keep a reference count per actual object.

Since :

7.0

Author :

stefanu

Method summary :

Modifier and type	Method
public int	getRefCount ()
public void	setRefCount (int refCount)

Method detail :

```
public int getRefCount ()
```

Returns :

the reference count for the specified object

```
public void setRefCount (int refCount)
```

Parameters :

`refCount` - the new reference count for the specified object.

REPLICABLE

```
public interface Replicable
```

Replicable objects are objects that may be copied.

Since :

7.0

Author :

stefanu

Method summary :

Modifier and type	Method
public void	replicate (com.siatel.defs.Identified replica)
public void	replicateFillOnly (com.siatel.defs.Identified replica)

Method detail :

```
public void replicate (Identified replica)
```

Parameters :

`replica` - the object that will be set to be a replica of the specified object.

```
public void replicateFillOnly (Identified replica)
```

Parameters :

`replica` - the object that will be set to be a replica of the specified object. Only null-value properties will be replicated, hence 'fillOnly'

SAMPLED

```
public interface Sampled
```

Sampled objects are objects that have an associated sample file; it is the responsibility of the business code to store and retrieve the files from the Gargantua file server accordingly.

Since :

7.5

See also :

com.siatel.framework.service.storage.StorageLibrary

Author :

elf

Method summary :

Modifier and type	Method
public java.lang.String	getSampleFileId ()
public void	setSampleFileId (java.lang.String fileId)

Method detail :

```
public String getSampleFileId ()
```

Returns :

the sample file ID for the specified object

```
public void setSampleFileId (String fileId)
```

Parameters :

`fileId` - the new sample file ID for the specified object

SECURABLE

`public interface Securable`

Securable interface is implemented by all objects that can have a security profile.

Since :

7.3

Author :

manuelam, elf

Method summary :

Modifier and type	Method
public java.lang.String	getSecurityProfileId ()
public void	setSecurityProfileId (java.lang.String

	securityProfile)
public int	getSecurityProfileIdNumeric ()
public void	setSecurityProfileIdNumeric (int securityProfileId)
public long	getSecurityMask ()
public void	setSecurityMask (long mask)

Method detail :

```
public String getSecurityProfileId ()
```

Accessor.

Returns :

the security profile of the specified object

```
public void setSecurityProfileId (String securityProfile)
```

Accessor.

Parameters :

`securityProfile` - the new security profile for the specified object

```
public int getSecurityProfileIdNumeric ()
```

Accessor.

Returns :

the numeric relevant part of the security profile id

```
public void setSecurityProfileIdNumeric (int securityProfileId)
```

Accessor.

Parameters :

`securityProfileId` - the numeric part of the security profile id

```
public long getSecurityMask ()
```

Accessor.

Returns :

the rendered security mask of the specified object (by connection)

```
public void setSecurityMask (long mask)
```

Accessor.

Parameters :

`securityMask` - the security mask for the specified object

SHARED

```
public interface Shared
```

Sharing support field accessors declarations.

Since :

7.0

Author :

victorc

Field summary :

Modifier and type	Field
public static final int	UNLOCK
public static final int	LOCK
public static final int	CHECK_OUT

Method summary :

Modifier and type	Method
public java.lang.String	getLockedBy ()
public void	setLockedBy (java.lang.String userId)
public int	getLockType ()
public void	setLockType (int lockType)
public boolean	isLocked ()
public java.util.Date	getDateL ()
public void	setDateL (java.util.Date dateL)
public java.lang.Long	getValidMillis ()
public void	setValidMillis (java.lang.Long millis)

Field detail :

```
public static final int UNLOCK = -1
```

Lock type constant defining the 'unlocked' object state.

```
public static final int LOCK = 0
```

Lock type constant defining the 'locked' object state.

```
public static final int CHECK_OUT = 1
```

Lock type constant defining the 'exclusive lock' object state.

Method detail :

```
public String getLockedBy ()
```

Accessor.

Returns :

the id of the user responsible for the 'locked' or 'locked exclusively' state of the object

```
public void setLockedBy (String userId)
```

Accessor.

Parameters :

`userId` - the id of the user responsible for the 'locked' or 'locked exclusively' state of the object

```
public int getLockType ()
```

Accessor.

Returns :

the lock type/state

```
public void setLockType (int lockType)
```

Accessor.

Parameters :

`lockType` - the lock type/state

```
public boolean isLocked ()
```

Accessor

Returns :

`true` if current object lock type is not {@link #UNLOCK}, `false` otherwise

```
public Date getDateL ()
```

Accessor.

Returns :

the lock date if current object lock type is not {@link #UNLOCK}

```
public void setDateL (Date dateL)
```

Accessor.

Parameters :

`dateL` - the lock date if current object lock type is not #UNLOCK

```
public Long getValidMillis ()
```

Accessor.

Returns :

the validity duration of the current object lock if lock type not {@link #UNLOCK}

```
public void setValidMillis (Long millis)
```

Accessor

Parameters :

`millis` - the validity duration of the current object lock if lock type not #UNLOCK

TAGGED

```
public interface Tagged
```

Tagged objects are objects that may have a set of tags associated with them. Tags are defined by the Tag constants, and are ORed constants

Since :

7.0

See also :

Tag

Author :

stefanu

Method summary :

Modifier and type	Method
public long	getTags ()
public void	setTags (long tags)

Method detail :

```
public long getTags ()
```

Returns :

the tags for the specified object

```
public void setTags (long tags)
```

Parameters :

`tags` - the new tags for the specified object

TIMEBOXED

```
public interface Timeboxed
```

This interface is for objects that have a due (estimated) and an expiry (validity) duration.

Since :

7.9

Author :

victorc

Method summary :

Modifier and type	Method
public java.lang.Long	getDelayD ()
public void	setDelayD (java.lang.Long delayD)
public java.lang.Long	getDelayX ()
public void	setDelayX (java.lang.Long delayX)

Method detail :

```
public Long getDelayD ()
```

Returns :

due (estimated) delay value in millis

```
public void setDelayD (Long delayD)
```

Parameters :

`delayD` - the new due delay value in millis

```
public Long getDelayX ()
```

Returns :

expiry (validity) delay value in millis

```
public void setDelayX (Long delayX)
```

Parameters :

`delayX` - the new expiry delay value in millis

TimeboxedEx

```
public interface TimeboxedEx
```

This interface is for Timeboxed objects that can specify the due (estimated) and expiry (validity) duration in a attributes specified by id.

All Superinterfaces :

Timeboxed

Since :

7.9

Author :

victorc

Method summary :

Modifier and type	Method
public java.lang.String	getDelayDAttrId ()
public void	setDelayDAttrId (java.lang.String delayDAttrId)
public java.lang.String	getDelayXAttrId ()
public void	setDelayXAttrId (java.lang.String delayXAttrId)

Method detail :

```
public String getDelayDAttrId ()
```

Returns :

due delay attribute id

```
public void setDelayDAttrId (String delayDAttrId)
```

Parameters :

`delayDAttrId` - id of attribute whose value will specify the delay

```
public String getDelayXAttrId ()
```

Returns :

expiry delay attribute id

```
public void setDelayXAttrId (String delayXAttrId)
```

Parameters :

`delayXAttrId` - id of attribute whose value will specify the delay

TIMED

`public interface Timed`

Wrapper for timed objects (processes, tasks) support accessors.

Since :

7.0

Author :

stefanu

Method summary :

Modifier and type	Method
`public java.util.Date`	`getDateS ()`
`public void`	`setDateS (java.util.Date dateS)`
`public java.util.Date`	`getDateE ()`
`public void`	`setDateE (java.util.Date dateE)`

Method detail :

```
public Date getDateS ()
```

Accessor.

Returns :

the start date for the specified object

```
public void setDateS (Date dateS)
```

Accessor.

Parameters :

`dateS` - the new start date for the specified object

```
public Date getDateE ()
```

Accessor.

Returns :

the end date for the specified object

```
public void setDateE (Date dateE)
```

Accessor.

Parameters :

`dateE` - dateS the new end date for the specified object

UUIDEnabled

`public interface UUIDEnabled`

This interface defines requirements for objects that support UUIDs as means of identification. A default support implementation is provided in UUIDDelegate. Default implementations of Folder, Document, Process support this interface through aggregating an UUIDDelegate and delegating to it.

Since :

7.8 / 2012.05

See also :

"http://en.wikipedia.org/wiki/Uuid"

Author :

victorc

Method summary :

Modifier and type	Method
`public java.util.UUID`	`getUUID ()`
`public void`	`setUUID (java.util.UUID uuidSrc)`
`public java.lang.String`	`getUUIDString ()`
`public void`	`setUUIDString (java.lang.String uuidStr)`

Method detail :

```
public UUID getUUID ()
```

Accessor.

Returns :

the object {@link UUID}

```
public void setUUID (UUID uuidSrc)
```

Accessor.

Parameters :

`uuidSrc` - the source UUID

```
public String getUUIDString ()
```

Accessor.

Returns :

the 36-character string formated object {@link UUID}. Short for {@link #getUUID()}`.toString()`.

```
public void setUUIDString (String uuidStr)
```

Accessor.

Parameters :

`uuidStr` - the source UUID in 36-character string format. Short for #setUUID(`UUID#fromString`(uuidStr))

Version

`public interface Version`

Public definition to group identified, realized and commented objects.

All Superinterfaces :

Identified , Realized , Commented

Since :

7.5

Author :

mariusz

Versioned

`public interface Versioned`

Versioned objects are objects for which the database keeps track of previous versions. Versions are

always represented as a long, but actually they keep track of two components : the major version number and the minor version number; these numbers are meant to be used as with software versions : the major number is incremented once some significant modifications are made to the object, while the minor version number is incremented if non-significant modifications are made.

When versioned objects are created the version number is set to 1.0; the user interface does not display 1.0 version numbers, as these are the defaults. Some objects may use the version number 0 (zero) as a default 1.0 version number.

Since :

7.0

See also :

VersionedEx

Author :

stefanu

Method summary :

Modifier and type	Method
public long	getVersion ()
public void	setVersion (long version)

Method detail :

```
public long getVersion ()
```

Accessor.

Returns :

the version number for the specified object

```
public void setVersion (long version)
```

Accessor.

Parameters :

`version` - the new version number for the specified object

VIEWABLE

```
public interface Viewable
```

Viewable objects are objects that have a set of user-defined attributes. The attributes are set using forms, and may be explicit or default attributes. Default attributes are set by the business components at runtime according to the current running context.

Since :

7.0

See also :

Attribute , Form , Type

Author :

stefanu

Method summary :

Modifier and type	Method
public java.lang.String	getFormId ()
public void	setFormId (java.lang.String formId)
public int	getFormIdNumeric ()
public void	setFormIdNumeric (int formId)
public java.util.List	getAttributes ()
public java.util.List	getCustomAttributes ()
public void	setAttributes (java.util.List attributes)
public java.lang.String	getAttributeValue (java.lang.String attributeId, java.util.Locale locale)
public void	setAttributeValue (java.lang.String attributeId, java.util.Locale locale, java.lang.String value)
public java.lang.Object	getAttributeValue (java.lang.String attributeId)
public void	setAttributeValue (java.lang.String attributeId, java.lang.Object value)
public java.util.List	setDefaultAttributes (java.util.List attributes, boolean bForced)
public java.util.List	getDefaultAttributes (java.util.List attributes)
public int	getIcon ()
public void	setIcon (int icon)

Method detail :

```
public String getFormId ()
```

Accessor.

Returns :

the form identification for the specified object

```
public void setFormId (String formId)
```

Accessor.

Parameters :

`formId` - the new form identification for the specified object

```
public int getFormIdNumeric ()
```

Accessor.

Returns :

the numeric form identification for the specified object

```
public void setFormIdNumeric (int formId)
```

Accessor.

Parameters :

`formId` - the new numeric form identification for the specified object

```
public List getAttributes ()
```

Accessor.

Returns :

the list of attributes for the specified object; these will always include the default attributes

```
public List getCustomAttributes ()
```

Accessor.

Returns :

the list of attributes for the specified object; these will always exclude the default attributes

```
public void setAttributes (List attributes)
```

Accessor.

Parameters :

`attributes` - the new attributes for the specified object; default attributes are always filtered out, as they are generally not user modifiable

```
public String getAttributeValue (String attributeId, Locale locale)
```

Accessor.

Parameters :

`attributeId` - the sttribute identification

`locale` - the i18n locale identifier for the current running context

Returns :

the value for the specified attribute, using the specified locale, for the specified object.

```
public void setAttributeValue (String attributeId, Locale locale, String value)
```

Accessor. Sets the attribute value for the specified attribute, using the specified locale identifier, for the specified object.

Parameters :

`attributeId` - the attribute identification

`locale` - the i18n locale identifier for the current running context

`value` - the attribute value

```
public Object getAttributeValue (String attributeId)
```

Accessor. Retrieves the attribute value; no locale conversion is applied

Parameters :

`attributeId` - the attribute identification

Returns :

the attribute value

```
public void setAttributeValue (String attributeId, Object value)
```

Accessor. Sets the attribute value for the specified object; no locale conversion is applied.

Parameters :

`attributeId` - the attribute identifier

`value` - the attribute value

```
public List setDefaultAttributes (List attributes, boolean bForced)
```

Modifies the input list of attributes, setting the default attribute values for the specified object.

Internal use only

Parameters :

`attributes` - the list of object attributes for the specified object

Returns :

the modified list of attributes, with all default attributes set.

```
public List getDefaultAttributes (List attributes)
```

Internal use only.

Parameters :

`attributes` -

Returns :

list of attributes

```
public int getIcon ()
```

Accessor.

Returns :

the icon index for the specified object

```
public void setIcon (int icon)
```

Accessor.

Parameters :

`icon` - the new icon index for the specified object

ViewableEx

```
public interface ViewableEx
```

ViewableEx extends the Viewable support by adding form page accessibility.

All Superinterfaces :

Viewable

Since :

7.0

See also :

Viewable , Form

Author :

victorc

Method summary :

Modifier and type	Method
public java.lang.String	getFormPage ()

public void	setFormPage (java.lang.String formPage)

Method detail :

```
public String getFormPage ()
```

Accessor.

Returns :

the form apge

```
public void setFormPage (String formPage)
```

Accessor.

Parameters :

`formPage` - the new form page

Package com.siatel.domain

This package contains the domain objects (data structures).

Interfaces

ArchiveComment

```
public interface ArchiveComment
```

The archive version of the comment object. Archive comments are attached to archived documents and folders;

The comment identification structure :

ID component	Value	Description
Prefix	ACDF, ACMF	See Prefix for more information on prefixes
Database ID	numeric	The database ID
Folder ID	numeric	The folder container numeric identification
Pin ID	numeric	The numeric pin ID, if the document is pinned; zero otherwise
Document ID	numeric	The numeric document ID if the comment is on a document; zero otherwise

Comment ID	numeric	The comment numeric identification

All Superinterfaces :

Comment , Archive

Author :

adinam

ARCHIVEDOCUMENT

```
public interface ArchiveDocument
```

The archive version of the document object.

The archive document identification structure :

ID component	Value	Description
Prefix	ADCF, AFCF, ARDF, ARFF	See Prefix for more information on prefixes
Database ID	numeric	The database ID
Folder ID	numeric	The folder container numeric identification
Pin ID	numeric	The numeric pin ID, if the document is pinned; zero otherwise
Document ID	numeric	The numeric document ID if the comment is on a document; zero otherwise
Comment ID	numeric	The comment numeric identification

All Superinterfaces :

Document , Archive

Author :

adinam

ARCHIVEDOCUMENTVERSION

```
public interface ArchiveDocumentVersion
```

The archived document version

The document version identification structure :

ID component	Value	Description
Prefix	VNDF, VNDP	See com.siatel.domain.Prefix for more information on prefixes
Database ID	numeric	The database ID
Folder or process ID	numeric	The folder or process container numeric identification
Pin ID	numeric	The numeric pin ID, if the document is pinned; zero otherwise
Document ID	numeric	The numeric document ID
Version ID	numeric	The numeric version ID

All Superinterfaces :

DocumentVersion , Archive

Author :

stefanu

ARCHIVEFOLDER

```
public interface ArchiveFolder
```

The archive version of a folder.

The archive folder identification structure :

ID component	Value	Description
Prefix	ARFL	See com.siatel.domain.Prefix for more information on prefixes
Database ID	numeric	The database ID
Folder ID	numeric	The numeric folder ID
Not used	0	-
Not used	0	-
Not used	0	-

All Superinterfaces :

Folder , Archive

Author :

stefanu

ARCHIVENOTE

```
public interface ArchiveNote
```

The assignment identification structure :

ID component	Value	Description
Prefix	ARFL	See com.siatel.domain.Prefix for more information on prefixes
Database ID	numeric	The database ID
Folder ID	numeric	The numeric folder ID
Not used	0	-
Not used	0	-
Not used	0	-

All Superinterfaces :

Note , Archive

Author :

stefanu

ASSIGNMENT

```
public interface Assignment
```

General assignment object; holds a source and destination identifier; used, for instance, to represent user assignments to groups. The assignment identification structure :

ID component	Value	Description
Prefix	ASGU, ASUR, ASGR, ASUD, ASRD, ASGD, ASUP, ASRP, ASGP, ASFA, ASUV, ASRV, ASGV, ASUF, ASGF, ASRF, ASUP, ASGP, ASRP, ASOP, ASFP, ASUT, ASGT, ASRT	The assignment object prefix; the prefix depends on the source and target object, but always starts with the letters "AS"; see com.siatel.domain.Prefix for more information on prefixes

Database ID	0	Always set to zero; the database identification is not required for these objects
Source ID	numeric	The numeric ID of the source object
Destination ID	numeric	The numeric ID of the destination object
Not used	0	Always set to zero
Not used	0	Always set to zero

All Superinterfaces :

Identified , Cloneable , Replicable

Author :

stefanu

Method summary :

Modifier and type	Method
public int	getDestIdNumeric ()
public void	setDestIdNumeric (int destId)
public int	getSourceIdNumeric ()
public void	setSourceIdNumeric (int sourceId)
public java.lang.Object	getData ()
public void	setData (java.lang.Object data)

Method detail :

```
public int getDestIdNumeric ()
```

Returns :

the destination numeric identification

```
public void setDestIdNumeric (int destId)
```

Parameters :

destId - the new destination numeric identification

```
public int getSourceIdNumeric ()
```

Returns :

the source numeric identification

```
public void setSourceIdNumeric (int sourceId)
```

Parameters :

`sourceId` - the new source numeric identification

```
public Object getData ()
```

Returns :

custom assignment data

```
public void setData (Object data)
```

Parameters :

`data` - custom assignment data

ATTRIBUTE

`public interface Attribute`

Attributes are assigned to objects that are Viewable. See the Viewable interface for information on how to retrieve and manipulate the Gargantua object attributes.

The attribute identification structure :

ID component	Value	Description
`Prefix`	`DFAT`	See com.siatel.domain.Prefix for more information on prefixes
`Database ID`	`numeric`	The database numeric ID for the Gargantua database that stores the object
`Attribute ID`	`numeric`	Numeric ID of the attribute within the specified database
`Not used`	`0`	-
`Not used`	`0`	-
`Not used`	`0`	-

All Superinterfaces :

Identified , Labeled , Owned , Described , Flagged , Dated , Cloneable , Replicable , Packageable , Cacheable

Author :

stefanu

CERTIFICATE

`public interface Certificate`

All Superinterfaces :

Identified , Labeled , Owned , Flagged , Described , Dated , Realized , Deployable , Replicable

Author :

olga

ClassificationElement

```
public interface ClassificationElement
```

Classification elements are used by Gargantua Views to create a virtual folder hierarchy based on attribute values.

The classification element identification structure :

ID component	Value	Description
Prefix	CELM	See com.siatel.domain.Prefix for more information on prefixes
Database ID	numeric	The database identification

All Superinterfaces :

Identified , Labeled , Cloneable , Replicable , Flagged

Author :

stefanu

ClassificationTreeElement

```
public interface ClassificationTreeElement
```

Classification elements are used by Gargantua Views to create a virtual folder hierarchy based on attribute values.

The classification element identification structure :

ID component	Value	Description
Prefix	CELM	See com.siatel.domain.Prefix for more information on prefixes
Database ID	numeric	The database identification

All Superinterfaces :

Identified , Labeled , Cloneable , Replicable , Flagged

Author :

stefanu

COMMENT

```
public interface Comment
```

The Comment object. Comments can be attached to documents ,folders or processes;

The comment identification structure :

ID component	Value	Description
Prefix	CMDF, CMDP , CMTF , CMTP	See com.siatel.domain.Prefix for more information on prefixes
Database ID	numeric	The database ID
Folder or process ID	numeric	The folder or process container numeric identification
Pin ID	numeric	The numeric pin ID, if the document is pinned; zero otherwise
Document ID	numeric	The numeric document ID if the comment is on a document; zero otherwise
Comment ID	numeric	The comment numeric identification

All Superinterfaces :

Labeled , Identified , Owned , Dated , Flagged , Replicable , Deletable

Author :

adinam

DATABASE

```
public interface Database
```

The 'database' concept.

The database identification structure :

ID component	Value	Description
Prefix	DATA	See com.siatel.domain.Prefix for more information on prefixes

Database ID	numeric	The database ID
Not used	0	-
Not used	0	-
Not used	0	-
Not used	0	-

All Superinterfaces :

Identified , Labeled , Owned , Dated , Described , Flagged , Cloneable , Replicable

Since :

15/03/2007

Author :

Victor Citiriga

Dependency

```
public interface Dependency
```

A **dependency** relationship concept definition. It specifies

- a **SOURCE** - the dependant, one who depends on another
- a **TARGET** - the dependency, one on which others depend

The intended usage of this concept is to cover the following common requirements:

- the **SOURCE** depends on the **TARGET** and so the **TARGET** may be required to be **LOCKED** for certain operations (change, deletion, etc)
- provide information on dependency relationships that prevent certain operations from taking place.

The requirements above have been extended in order to support basic-typed custom fields in which, based on specific usage requirements the user may store dependency-relevant information

All Superinterfaces :

Flagged

Since :

02/02/2010

Author :

victorc

DistributionListEntry

```
public interface DistributionListEntry
```

General distribution list object; holds all information needed to to represent an entry of distribution list
The distribution list entry identification structure :

ID component	Value	Description
Prefix	DEDF, DEPF, DEFL	The distribution list entry object prefix; the prefix depends on the object, but always starts with the letters "DE"; see com.siatel.domain.Prefix for more information on prefixes
Database ID	numeric	The database ID
object ID	numeric	The numeric ID of the object
accessLevel ID	numeric	The numeric ID of the access level object
actor ID	numeric	The numeric ID of the actor object (can be user,group,role,attribute)
actor type	numeric	The category type of the actor (can be user,group,role,userlist,groupList,roleList)

All Superinterfaces :

Identified , Labeled , Flagged , Timed , Cloneable , Replicable

Document

```
public interface Document
```

The Gargantua document.

The document identification structure :

ID component	Value	Description
Prefix	DOCF, DOCP, PINF, PINP, FCHF, FCHP	See com.siatel.domain.Prefix for more information on prefixes
Database ID	numeric	The database ID
Folder or process ID	numeric	The folder or process container numeric identification
Pin ID	numeric	The numeric pin ID, if the document is pinned; zero otherwise

Document ID	numeric	The numeric document ID
Not used	0	-

All Superinterfaces :

Identified , Labeled , Owned , DatedEx , Flagged , ViewableEx , RealizedEx , Versioned , Lifecycled , Shared , Cloneable , Replicable , Securable , UUIDEnabled , Tagged , Deletable , Modified

Author :

stefanu

DocumentVersion

```
public interface DocumentVersion
```

The Gargantua document version

The document version identification structure :

ID component	Value	Description
Prefix	AVDF	See com.siatel.domain.Prefix for more information on prefixes
Database ID	numeric	The database ID
Folder or process ID	numeric	The folder or process container numeric identification
Pin ID	numeric	The numeric pin ID, if the document is pinned; zero otherwise
Document ID	numeric	The numeric document ID
Version ID	numeric	The numeric version ID

All Superinterfaces :

Identified , Labeled , Owned , DatedEx , Flagged , RealizedEx , Cloneable , Replicable , Deletable

Author :

stefanu

ExpirableDownloadLink

```
public interface ExpirableDownloadLink
```

The expirable download link is a link that has either a maximum access times or a maximum validity

time.

The expirable download link identification structure :

ID component	Value	Description
Prefix	EXDL	The expirable download link prefix
Database ID	0	The database ID that stores the definition; since the links are stored in the administration database, this is always set to zero
User ID	numeric	The numeric ID of the expirable download link
Not used	0	Always set to zero
Not used	0	Always set to zero
Not used	0	Always set to zero
Not used	0	Always set to zero

All Superinterfaces :

Identified , Labeled , Flagged , Dated , Replicable , Cloneable

Author :

mariusz

ExtendedProperty

```
public interface ExtendedProperty
```

This interface represents the definition for a extended property. The extended property holds extra-data for an object and is characterized by a label, type and value.

The structure for any extended property is as follows:

ID component	Value	Description
Prefix	EXPR	The extended property prefix
Database ID	0	The database ID that stores the extened property definition; since the properties are stored in the administration database, this is always set to zero
Extended property ID	numeric	The numeric ID of the extended property object
Not used	0	Always set to zero
Not used	0	Always set to zero

Not used	0	Always set to zero
Not used	0	Always set to zero

All Superinterfaces :

Identified , Labeled , Flagged , Described , Dated , Replicable , Cloneable

Author :

mariusz

Folder

```
public interface Folder
```

The Folder object

The assignment identification structure :

ID component	Value	Description
Prefix	FLDR	See com.siatel.domain.Prefix for more information on prefixes
Database ID	numeric	The database ID
Folder ID	numeric	The numeric folder ID
Not used	0	-
Not used	0	-
Not used	0	-

All Superinterfaces :

Identified , Labeled , Owned , Flagged , DatedEx , ViewableEx , Lifecycled , Cloneable , Replicable , Securable , UUIDEnabled , Deletable , Modified

Author :

stefanu

Form

```
public interface Form
```

The Form object. Forms are used by users to assign attributes to objects

The form identification structure :

ID component	Value	Description

Prefix	DFFR	See com.siatel.domain.Prefix for more information on prefixes
Database ID	numeric	The database ID
Not used	0	-
Not used	0	-
Not used	0	-
Not used	0	-

All Superinterfaces :

Identified , Labeled , Owned , Flagged , Dated , Described , DeployableEx , Realized , Cloneable , Replicable , Lifecycle , Layouted , HierarchyCreator , Packageable , Cacheable

Author :

stefanu

Grid

```
public interface Grid
```

The grid object.

All Superinterfaces :

Identified , Labeled , Owned , Flagged , Dated , Described , Replicable , Realized , Cloneable , Packageable , Cacheable

See also :

Prefix#GRID for id structure

Group

```
public interface Group
```

The Group object. Groups statically group users that share the same access rights on Gargantua objects.

The group identification structure :

ID component	Value	Description
Prefix	GROU	See com.siatel.domain.Prefix for more information on prefixes
Not used	0	-

Group ID	numeric	The numeric identification of the group object
Not used	0	-
Not used	0	-
Not used	0	-

All Superinterfaces :

Identified , Labeled , Owned , Flagged , Described , Dated , Replicable , Cloneable , Cacheable

Author :

stefanu

HierarchyCreator

```
public interface HierarchyCreator
```

This interface is implemented by objects that generate a custom hierarchy after creation

Since :

7.10

Author :

mariusz

InterventionReport

```
public interface InterventionReport
```

The intervention report indicates an intervention done on the Gargantua server.

The intervention report object identification structure :

ID component	Value	Description
Prefix	INRE	The intervention report object prefix
Database ID	0	The database ID that stores the intervention report definition; since the delegations are stored in the administration database, this is always set to zero
Intervention ID	numeric	The numeric ID of the intervention report object
Not used	0	Always set to zero

Not used	0	Always set to zero
Not used	0	Always set to zero
Not used	0	Always set to zero

All Superinterfaces :

Identified , Labeled , Flagged , Described , Dated , Replicable , Cloneable , Realized

Author :

stefanu

ISJOB

```
public interface IsJob
```

The IsJob object.

The form identification structure :

ID component	Value	Description
Prefix	ISJB	See com.siatel.domain.Prefix for more information on prefixes
Database ID	numeric	The database ID
IsJob ID	numeric	The IS Job numeric identification
Not used	0	-
Not used	0	-
Not used	0	-

All Superinterfaces :

Identified , Labeled , Owned , Flagged , Dated , Described , Realized , DeployableEx , Cloneable , Replicable , Packageable , Cacheable

Author :

manuelam

LAYOUTED

```
public interface Layouted
```

The layouter interface is implemented by objects that have a custom presentation (layout), usually a fragment

Since :

7.10

Author :

mariusz

Lifecycle

```
public interface Lifecycle
```

Lifecycle enabling objects requirements.

Author :

elf

Manager

```
public interface Manager
```

This interface wraps the model requirements of the Manager object.

All Superinterfaces :

Identified , Labeled , Owned , Flagged , Described , Replicable , Cloneable

Since :

R562

Author :

victorc

ModelOCR

```
public interface ModelOCR
```

The ModelOCR object.

The form identification structure :

ID component	Value	Description
`Prefix`	`MOCR`	See com.siatel.domain.Prefix for more information on prefixes
`Database ID`	`numeric`	The database ID
`ModelOCR ID`	`numeric`	The Model OCR numeric identification

Not used	0	-
Not used	0	-
Not used	0	-

All Superinterfaces :

Identified , Labeled , Owned , Flagged , Dated , Described , Realized , DeployableEx , Sampled , Cloneable , Replicable , Packageable , Cacheable

Author :

manuelam

NOTE

```
public interface Note
```

The Note object. Notes can be attached to documents or folders; the note's content is stored as file, thus being indexed in the fulltext database, if available. Depending on the parent object, notes may have a rich text format (in the case of folders)

The note identification structure :

ID component	Value	Description
Prefix	NTDF, NTDP	See com.siatel.domain.Prefix for more information on prefixes
Database ID	numeric	The database ID
Folder or process ID	numeric	The folder or process container numeric identification
Pin ID	numeric	The numeric pin ID, if the document is pinned; zero otherwise
Document ID	numeric	The numeric document ID
Note ID	numeric	The note numeric identification

All Superinterfaces :

Identified , Owned , DatedEx , Realized , Replicable , Cloneable , Deletable

Author :

stefanu

ObjectExtendedProperty

```
public interface ObjectExtendedProperty
```

This interface represents the definition for a assignment of a extended property to a object.

The structure for any extended property is as follows:

ID component	Value	Description
Prefix	ASOP	The extended property prefix
Database ID	0	The database ID that stores the extened property definition; since the properties are stored in the administration database, this is always set to zero
Extended property ID	numeric	The numeric ID of the extended property object
Assignment id	numeric	The numeric ID of the assignment
Extended property id	numeric	The numeric ID of the extended property
Object id	numeric	The numeric ID of the object
Not used	0	Always set to zero

All Superinterfaces :

Assignment

Author :

mariusz

Package

```
public interface Package
```

The Package object

The assignment identification structure :

ID component	Value	Description
Prefix	PCKG	See com.siatel.domain.Prefix for more information on prefixes
Database ID	numeric	The database ID
Package ID	numeric	The numeric package ID
Not used	0	-

Not used	0	-
Not used	0	-

All Superinterfaces :

Identified , Labeled , Described , Dated , Flagged , Owned , Cloneable , Replicable

Author :

mariusz

Planner

```
public interface Planner
```

This interface declares the structure of the planner object. A planner consists of a weekly timetable and list of exceptions from it.

In addition to the accessors defined by the inherited interfaces, this interfaces defines accessors to the weekly timetable.

`'Timetable'` is an array of 7 long values. Each element in the array corresponds to a day in the week, 0 representing monday, 6 representing sunday. For each day, the least significant 48 bits represent half-hour intervals. Bit 0 corresponds to 0:00-0:30, 1 to 0:30-1:00, 47 to 23:30-0:00. If a bit is set then it represents activity time, if it is reset then it represents inactivity time.

The planner identification structure :

ID component	Value	Description
Prefix	PLAN	See com.siatel.domain.Prefix for more information on prefixes
Not used	0	-
Planner ID	numeric	The numeric identification of the planner object
Not used	0	-
Not used	0	-
Not used	0	-

All Superinterfaces :

Identified , Labeled , Owned , Flagged , Dated , Described , Deployable , Cloneable , Replicable

Since :

27/03/2008

Author :

victorc

See also :

PlannerDayImpl

PlannerDay

```
public interface PlannerDay
```

This interface declares the structure of an exception from a planner's weekly timetable.

The numeric part of a planner day object will be represented by the numeric evaluation of the date string in YYYYMMDD format. E.g. the full planner day for the planner with numeric id 1 and date 2000/01/01 will be obtained by this code:

```
String dayId = IdFormat.toString(Prefix.PANNER_DAY, 0, 1, 20000101);
```

All Superinterfaces :

Identified , Cloneable , Replicable

Since :

27/032008

Author :

victorc

See also :

Planner

Portal

```
public interface Portal
```

The Portal object

The portal identification structure :

ID component	Value	Description
`Prefix`	`PORT`	See com.siatel.domain.Prefix for more information on prefixes
`Not used`	`0`	-
`Portal ID`	`numeric`	The portal numeric identification
`Not used`	`0`	-
`Not used`	`0`	-

Not used	0	-

All Superinterfaces :

Identified , Labeled , Owned , Described , Realized , Flagged , Dated , Cloneable , Replicable

Author :

stefanu

Process

```
public interface Process
```

The Process object

The process identification structure :

ID component	Value	Description
Prefix	PROC	See com.siatel.domain.Prefix for more information on prefixes
Database ID	numeric	The database ID
Process ID	numeric	The process numeric identification
Not used	0	-
Not used	0	-
Not used	0	-

All Superinterfaces :

Identified , Labeled , Owned , DatedEx , Flagged , Realized , Prioritized , Managed , Referenced , ViewableEx , Timed , Expirable , Replicable , Cloneable , Securable , UUIDEnabled , Modified

Author :

stefanu

ProcessDefinition

```
public interface ProcessDefinition
```

The Process definition object

The process definition identification structure :

ID component	Value	Description

Prefix	DFPR	See com.siatel.domain.Prefix for more information on prefixes
Database ID	numeric	The database ID
Definition ID	numeric	The process definition numeric identification
Not used	0	-
Not used	0	-
Not used	0	-

All Superinterfaces :

Identified , Labeled , Owned , Dated , Described , Flagged , Realized , DeployableEx , Replicable , Referenced , Cloneable , Securable , TimeboxedEx , Packageable , Cacheable

Author :

stefanu

ProcessDefinitionVersion

```
public interface ProcessDefinitionVersion
```

The Process definition version object

The process definition version identification structure :

ID component	Value	Description
Prefix	DFPR	See com.siatel.domain.Prefix for more information on prefixes
Database ID	numeric	The database ID
Definition ID	numeric	The process definition numeric identification
Definition Version ID	The process definition version numeric identification	-
Not used	0	-
Not used	0	-

All Superinterfaces :

Identified , Labeled , Owned , Dated , Described , Flagged , Realized , DeployableEx , Replicable , Referenced , Cloneable , Securable , Version , Numbered , TimeboxedEx

Author :

stefanu

RESOURCE

```
public interface Resource
```

The resource object.

All Superinterfaces :

Identified , Labeled , Described , Flagged , Owned , Dated , Realized , Cloneable , Replicable , Cacheable

See also :

Prefix#RESOURCE

ROLE

```
public interface Role
```

The role object.

All Superinterfaces :

Identified , Labeled , Owned , Flagged , Described , Dated , Replicable , Cloneable , Cacheable

See also :

Prefix#ROLE for id structure , Privilege for privilege definitions

RULE

```
public interface Rule
```

The storage rule object interface.

The object links a database to a storage area based on a list of conditions. The rule priority is used to enforce an evaluation order. Higher priority is specified as lower integer value.

All Superinterfaces :

Identified , Cloneable , Replicable , Flagged , Cacheable

See also :

Prefix#DATABASE_STORAGE_RULE

Script

```
public interface Script
```

The 'script' concept definition. This concept groups script code (stored in a file) with script meta-data (lang - the programming language, type - the execution context/layer) and Siatel-object identification naming and common metadata (dates, flags etx).

All Superinterfaces :

Identified , Labeled , Owned , Dated , Flagged , Described , Deployable , Realized , Serializable , Cloneable , Replicable

Author :

victorc

Storage

```
public interface Storage
```

The storage area object interface.

All Superinterfaces :

Identified , Labeled , Owned , Described , Flagged , Dated , Cloneable , Replicable , Cacheable

See also :

Prefix#STORAGE_AREA

Task

```
public interface Task
```

The task object.

All Superinterfaces :

Identified , Labeled , Owned , DatedEx , Described , Flagged , Timed , Expirable , Managed , Prioritized , Replicable , Cloneable

See also :

Prefix#TASK for id structure

Type

```
public interface Type
```

The type object.

All Superinterfaces :

Identified , Labeled , Owned , Flagged , Dated , Described , Replicable , Realized , Cloneable , Packageable , Cacheable

See also :

Prefix#TYPE for id structure

User

```
public interface User
```

The user interface specifies the behavior of user objects. A user is a physical person that interacts with the system; a user has a password that is used in order to authenticate within the system.

The user may belong to groups and may be assigned to certain roles by the system administrator

The user identification structure :

ID component	Value	Description
Prefix	USER	The user object prefix
Database ID	0	The database ID that stores the user definition; since the users are stored in the administration database, this is always set to zero
User ID	numeric	The numeric ID of the user object
Not used	0	Always set to zero
Not used	0	Always set to zero
Not used	0	Always set to zero
Not used	0	Always set to zero

All Superinterfaces :

Identified , Labeled , Owned , Flagged , Described , Dated , Replicable , Cloneable , Cacheable

Author :

stefanu

UserDelegation

```
public interface UserDelegation
```

The user delegation interface specifies the behavior of user delegation objects. A user can delegate his work to another user during his absence. See the specfile DELEGATE for more details.

The user delegation object identification structure :

ID component	Value	Description
Prefix	DLGT	The user delegation object prefix
Database ID	0	The database ID that stores the user delegation definition; since the delegations are stored in the administration database, this is always set to zero
Delegation ID	numeric	The numeric ID of the user delegation object
Not used	0	Always set to zero
Not used	0	Always set to zero
Not used	0	Always set to zero
Not used	0	Always set to zero

All Superinterfaces :

Identified , Owned , Flagged , Described , Dated , Timed , Replicable , Cloneable

Author :

stefanu

View

```
public interface View
```

This interface defines the 'view' concept and groups concept definition data accessors.

A view consists of a hierarchy of 'view elements'. A view WILL NOT usable until deployed. The view deployment process MUST validate the 'view element' hierarchy and compile database queries for each of the view elements it contains.

Contained view elements will start with a parent id specifying 0 in it's numeric id part. Example:

DFVE0001000000010000000000000000000000000 is the parent id for root view elements of the DFVW0001000000010000000000000000000000000 view.

All Superinterfaces :

Identified , Labeled , Owned , Flagged , Dated , Described , Deployable , Replicable , Cloneable , Packageable , Cacheable

Since :

12/07/2007

Author :

vic

See also :

com.siatel.domain.ViewElement , com.siatel.defs.Identified , com.siatel.defs.Labeled , com.siatel.defs.Owned , com.siatel.defs.Flagged , com.siatel.defs.Dated , com.siatel.defs.Described , com.siatel.defs.Deployable , com.siatel.defs.Replicable , java.lang.Cloneable

ViewElement

```
public interface ViewElement
```

This interface defines the 'view element' concept and groups concept definition data accessors.

A 'view element' consists of the following:

1. the parent id (view elements are organzied hierarchically)
2. a domain filter specification
3. a unicity rule specification
4. a list of name tokens
5. a sort order specification

The **parent id** specifies the id of another view element. All view elements MUST have a parent id set. For view elements defined directly in the view root the numeric part of the parent id is 0.

The **domain filter** specifies a list of filter criteria that restricts the values domain for the attributes participating in generating this element's classes and the children classes generated by this element's children. If more than one criterion is specified then criteria are combined using 'AND' logic operator. **No more than 3 filter criteria can be specified by a view item.**

The **unicity rule** consists of a list if unicity criteria, each of them specifying what attribute or attribute part participates in generating a class. If more than one criterion is specified then the combination of all criteria specified must render to a unique set. **No more than 3 unicity criteria can be specified by a view item.**The rule will be ignored for view elements that specify `IS_OBJECTS` in their `flags` member (see below for more flags).

The **name tokens** must be specified in the order they will appear in the generated classes name. Recognizable and renderable tokens are declared in this interface by the `TOKEN_UNIQUE` constants. The number in the name of the constant specifies which criterion from the unique criteria list to render in the class name. Additional tokens are declared in the `ValueToken` utility class. Plain text tokens can be specified and they will render identically in the generated class name.

The **sort order** refers to the order in which the generated classes are enumerated. It is specified by the

'flags' member of this class by the `Flag.ORDER` flags grouped by `Flag.CLASS_SORT_MASK`. The number of the flag name specifies to which of the unicity criteria the sort order applies.

The hierachical strucure of the view elements will result in the fact that the domain filters and unique criteria of a view element will act as additional restrictions for it's children when generating the children classes.

Additional flags MUST be specified in the `flags` member of this interface in `Flag.CLASSES_MASK` range. See below table for allowed combinations.

Combination	Significance
FOR_FOLDERS	Specifies that the classes to which this view element evaluates are based on folders attributes only.
FOR_FOLDERS | IS_OBJECTS	Specifies that the classes to which this view element evaluates are actual folders. The unique criteria and name tokens will be ignored in this case. Persistence call will be made for folders not for classes when instantiating this view element.
FOR_PROCESSES	Specifies that the classes to which this view element evaluates are based on processes attributes only.
FOR_PROCESSES | IS_OBJECTS	Specifies that the classes to which this view element evaluates are actual processes. The unique criteria and name tokens will be ignored in this case. Persistence call will be made for processes not for classes when instantiating this view element.
FOR_FOLDERS | FOR_PROCESSES	Specifies that the classes to which this view element evaluates are based on folders and processes attributes only.
FOR_FOLDERS | FOR_PROCESSES | IS_OBJECTS	Specifies that the classes to which this view element evaluates are actual folders and processes. The unique criteria and name tokens will be ignored in this case. Persistence call will be made for folders and processes not for classes when instantiating this view element.
FOR_DOCUMENTS	Specifies that the classes to which this view element evaluates are based on documents attributes only.
FOR_DOCUMENTS | IS_OBJECTS	Specifies that the classes to which this view element evaluates are actual documents. The unique criteria and name tokens will be ignored in this case. Persistence call will be made for documentts not for classes when instantiating this view element.

All Superinterfaces :

Identified , Labeled , Owned , Flagged , Dated , Described , Replicable , Cloneable

Since :

12/07/2007

See also :

com.siatel.domain.View , com.siatel.domain.FilterCriterion , com.siatel.domain.UniqueCriterion , com.siatel.domain.Flag , com.siatel.domain.ValueToken , com.siatel.defs.Identified ,

com.siatel.defs.Labeled , com.siatel.defs.Owned , com.siatel.defs.Flagged , com.siatel.defs.Dated , com.siatel.defs.Described , com.siatel.defs.Replicable , java.lang.Cloneable

Author :

vic

ViewElementVersion

```
public interface ViewElementVersion
```

This interface defines the 'view element version' concept and groups concept definition data accessors.

A 'view element' consists of the following:

1. the parent id (view elements are organized hierarchically)
2. a domain filter specification
3. a unicity rule specification
4. a list of name tokens
5. a sort order specification

The **parent id** specifies the id of another view element. All view elements MUST have a parent id set. For view elements defined directly in the view root the numeric part of the parent id is 0.

The **domain filter** specifies a list of filter criteria that restricts the values domain for the attributes participating in generating this element's classes and the children classes generated by this element's children. If more than one criterion is specified then criteria are combined using 'AND' logic operator. **No more than 3 filter criteria can be specified by a view item.**

The **unicity rule** consists of a list if unicity criteria, each of them specifying what attribute or attribute part participates in generating a class. If more than one criterion is specified then the combination of all criteria specified must render to a unique set. **No more than 3 unicity criteria can be specified by a view item.**The rule will be ignored for view elements that specify `IS_OBJECTS` in their `flags` member (see below for more flags).

The **name tokens** must be specified in the order they will appear in the generated classes name. Recognizable and renderable tokens are declared in this interface by the `TOKEN_UNIQUE` constants. The number in the name of the constant specifies which criterion from the unique criteria list to render in the class name. Additional tokens are declared in the `ValueToken` utility class. Plain text tokens can be specified and they will render identically in the generated class name.

The **sort order** refers to the order in which the generated classes are enumerated. It is specified by the 'flags' member of this class by the `Flag.ORDER` flags grouped by `Flag.CLASS_SORT_MASK`. The number of the flag name specifies to which of the unicity criteria the sort order applies.

The hierachical strucure of the view elements will result in the fact that the domain filters and unique criteria of a view element will act as additional restrictions for it's children when generating the children

classes.

Additional flags MUST be specified in the `flags` member of this interface in `Flag.CLASSES_MASK` range. See below table for allowed combinations.

Combination	Significance
FOR_FOLDERS	Specifies that the classes to which this view element evaluates are based on folders attributes only.
FOR_FOLDERS | IS_OBJECTS	Specifies that the classes to which this view element evaluates are actual folders. The unique criteria and name tokens will be ignored in this case. Persistence call will be made for folders not for classes when instantiating this view element.
FOR_PROCESSES	Specifies that the classes to which this view element evaluates are based on processes attributes only.
FOR_PROCESSES | IS_OBJECTS	Specifies that the classes to which this view element evaluates are actual processes. The unique criteria and name tokens will be ignored in this case. Persistence call will be made for processes not for classes when instantiating this view element.
FOR_FOLDERS | FOR_PROCESSES	Specifies that the classes to which this view element evaluates are based on folders and processes attributes only.
FOR_FOLDERS | FOR_PROCESSES | IS_OBJECTS	Specifies that the classes to which this view element evaluates are actual folders and processes. The unique criteria and name tokens will be ignored in this case. Persistence call will be made for folders and processes not for classes when instantiating this view element.
FOR_DOCUMENTS	Specifies that the classes to which this view element evaluates are based on documents attributes only.
FOR_DOCUMENTS | IS_OBJECTS	Specifies that the classes to which this view element evaluates are actual documents. The unique criteria and name tokens will be ignored in this case. Persistence call will be made for documentts not for classes when instantiating this view element.

All Superinterfaces :

Identified , Labeled , Owned , Flagged , Dated , Described , Replicable , Cloneable

Since :

12/07/2007

See also :

com.siatel.domain.View , com.siatel.domain.FilterCriterion , com.siatel.domain.UniqueCriterion , com.siatel.domain.Flag , com.siatel.domain.ValueToken , com.siatel.defs.Identified , com.siatel.defs.Labeled , com.siatel.defs.Owned , com.siatel.defs.Flagged , com.siatel.defs.Dated ,

com.siatel.defs.Described , com.siatel.defs.Replicable , java.lang.Cloneable , com.siatel.domain.ViewVersion

Author :

vic

ViewVersion

```
public interface ViewVersion
```

This interface defines the 'view version' concept and groups concept definition data accessors.

All Superinterfaces :

Identified , Labeled , Owned , Flagged , Dated , Described , Deployable , Replicable , Cloneable , Commented , Numbered

Since :

12/07/2007

Author :

vic

See also :

com.siatel.domain.View , com.siatel.domain.ViewElement , com.siatel.defs.Identified , com.siatel.defs.Labeled , com.siatel.defs.Owned , com.siatel.defs.Flagged , com.siatel.defs.Dated , com.siatel.defs.Described , com.siatel.defs.Deployable , com.siatel.defs.Replicable , java.lang.Cloneable

Classes

AttributeModifier

```
public class AttributeModifier
```

This utility class defines attribute modifiers used by view elements unique criteria.

Since :

12/07/2007

Author :

vic

Constructor summary :

Constructor

AttributeModifier ()

Constructor detail :

```
public AttributeModifier ()
```

AttributeProp

```
public class AttributeProp
```

Attribute properties; depending on the attribute type, several properties can be set

Author :

cristi

Constructor summary :

Constructor
AttributeProp ()

Constructor detail :

```
public AttributeProp ()
```

Attributes

```
public class Attributes
```

This utility class defines default attribute constants.

Since :

12/07/2007

Author :

stefanu

Constructor summary :

Constructor
Attributes ()

Constructor detail :

```
public Attributes ()
```

FilterCriterion

```
public class FilterCriterion
```

This class defines a filter criterion used by view elements definition.

This class consists of:

1. the **id of the attribute** used to filter a domain.
2. 4 **reference values** which specify the domain (min, max, eq, ne).

Supported combinations of the reference values are:

combination	significance
(min != null, max = null, eq = null, ne = null, vN = null, vNNull = null)	specifies attribute must have values greater than min
(min = null, max != null, eq = null, ne = null, vN = null, vNNull = null)	specifies attribute must have values lower than min
(min != null, max != null, eq = null, ne = null, vN = null, vNNull = null)	specifies attribute must have values in the range between min and max
(min = null, max = null, eq != null, ne = null, vN = null, vNNull = null)	specifies attribute values must be equal to eq
(min = null, max = null, eq = null, ne != null, vN = null, vNNull = null)	specifies attribute values must not be equal to eq
(min = null, max = null, eq = null, ne = null, vN != null, vNNull = null)	specifies attribute values must not be null
(min = null, max = null, eq = null, ne = null, vN = null, vNNull != null)	specifies attribute values must not be null

All Superinterfaces :

Cloneable

Since :

12/07/2006

Author :

vic

Constructor summary :

Constructor
`FilterCriterion ()`
`FilterCriterion (java.lang.String attrId, java.lang.Object valueMin,`

```
java.lang.Object valueMax, java.lang.Object valueEq, java.lang.Object
valueNe)
FilterCriterion (java.lang.String attrId, java.lang.Object valueMin,
java.lang.Object valueMax, java.lang.Object valueEq, java.lang.Object
valueNe, java.lang.Object valueCt, java.lang.Object valueNc)
FilterCriterion (java.lang.String attrId, java.lang.Object valueMin,
java.lang.Object valueMax, java.lang.Object valueEq, java.lang.Object
valueNe, java.lang.Object valueCt, java.lang.Object valueNc,
java.lang.Object valueIsNull, java.lang.Object valueIsNotNull)
```

Constructor detail :

```
public FilterCriterion ()
```

No arguments constructor.

```
public FilterCriterion (String attrId, Object valueMin, Object valueMax,
Object valueEq, Object valueNe)
```

Constructor with initialization parameters.

Parameters :

`attrId` - the attribute id

`valueMin` - the min value

`valueMax` - the max value

`valueEq` - the eq value

`valueNe` - the ne value

```
public FilterCriterion (String attrId, Object valueMin, Object valueMax,
Object valueEq, Object valueNe, Object valueCt, Object valueNc)
```

```
public FilterCriterion (String attrId, Object valueMin, Object valueMax,
Object valueEq, Object valueNe, Object valueCt, Object valueNc, Object
valueIsNull, Object valueIsNotNull)
```

FLAG

```
public class Flag
```

Various flags used across the Gargantua system.

Bits 0-39 match to core definition interfaces' accessors (Identified, Labeled, Owned etc.).

Bits 40-63 match to core objects specific interfaces' accessors (Folder, Process, Document etc.)

All Superinterfaces :

IEFlag , SearchFlag , AuditFlag , OtherFlag

Author :

stefanu, victorc

Field summary :

Modifier and type	Field
public static final long	ID
public static final long	LABEL
public static final long	OWNER_ID
public static final long	FLAGS
public static final long	DESCRIPTION
public static final long	DEPLOYED
public static final long	VERSION
public static final long	FORM_ID
public static final long	ATTRIBUTES
public static final long	FILE_ID
public static final long	FILE_TYPE
public static final long	FILE_SIZE
public static final long	DATE_C
public static final long	DATE_M
public static final	DATE_A

long	
public static final long	ICON
public static final long	DATE_S
public static final long	DATE_E
public static final long	DATE_D
public static final long	DATE_X
public static final long	OVERDUE
public static final long	EXPIRED
public static final long	REFCOUNT
public static final long	PRIORITY
public static final long	SECURITY_PROFILE
public static final long	RENDER_SECURITY
public static final long	UUID
public static final long	TAGS
public static final long	DATE_AR
public static final long	EMAIL
public static final long	REAL_NAME
public static final long	DIRECTORY
public static final long	PASSWORD

public static final long	PRIVILEGES
public static final long	PROFILE
public static final long	ACTIVATED_BY
public static final long	ACTIVATED_AT
public static final long	DEACTIVATED_BY
public static final long	DEACTIVATED_AT
public static final long	DATE_LI
public static final long	DATE_LO
public static final long	WORKING_PLANNER
public static final long	ACCESS_RESTRICTED_BY_WP
public static final long	PARENT_ID
public static final long	DESCRIPTION_URL
public static final long	CONN_URL
public static final long	CRON_EXPRESSION
public static final long	SCRIPT_LANG
public static final long	SCRIPT_TYPE
public static final long	SAMPLE_FILE_ID
public static final long	DISPLAYPROFILE_ID
public static final	LIFECYCLE_RULE_FILEID

long	
public static final long	ARCHIVE_STORAGE_ID
public static final long	FRAGMENT_FILE_ID
public static final long	HIERARCHY_FILE_ID
public static final long	VERS_OBJ_COMMENT
public static final long	VERS_OBJ_INDEX
public static final long	DATA
public static final long	VERSION_OF
public static final long	DELAY_X
public static final long	DELAY_D
public static final long	PROCESS_ID_PLANNER
public static final long	DEPLOYMENT_FORM_ID
public static final long	DEFINITION_ID
public static final long	RUNTIME_STATE
public static final long	RUNTIME_FILE_ID
public static final long	PROCESS_STATUS
public static final long	PROCESS_START_POINT
public static final long	DRAFT_FORM_PAGE
public static final long	PROFILE_ID

public static final long	TASK_STATUS
public static final long	TASK_ACTORID
public static final long	TASK_EXECUTORID
public static final long	TASK_LOCATION
public static final long	TASK_SUBJECT
public static final long	TASK_PROCESS_LABEL
public static final long	TASK_FORM_PAGE
public static final long	TASK_FORM_ID
public static final long	TIMETABLE
public static final long	DATABASE_ID
public static final long	DAY_TIMETABLE
public static final long	DAY_DATE
public static final long	DAY_RECURRENT
public static final long	RES_TYPE
public static final long	ENUM_MASK
public static final long	GENERIC_ASC
public static final long	GENERIC_DESC
public static final long	BY_DEFAULT_ASC
public static final	BY_DEFAULT_DESC

long	
public static final long	BY_LABEL_ASC
public static final long	BY_LABEL_DESC
public static final long	BY_DATEC_ASC
public static final long	BY_DATEC_DESC
public static final long	BY_DATEM_ASC
public static final long	BY_DATEM_DESC
public static final long	BY_DATES_ASC
public static final long	BY_DATES_DESC
public static final long	BY_DATED_ASC
public static final long	BY_DATED_DESC
public static final long	BY_PRIO_ASC
public static final long	BY_PRIO_DESC
public static final long	COMMENT_CONTENT
public static final long	OWNER_NAME
public static final long	HIGHER_ID
public static final long	USER_ALL
public static final long	USER_UPDATE_ALL
public static final long	GROUP_ALL

public static final long	GROUP_UPDATE_ALL
public static final long	ROLE_ALL
public static final long	ROLE_UPDATE_ALL
public static final long	PORTAL_ALL
public static final long	PORTAL_UPDATE_ALL
public static final long	PLANNER_ALL
public static final long	PLANNER_DAY_ALL
public static final long	DATABASE_ALL
public static final long	DATABASE_UPDATE_ALL
public static final long	STORAGE_ALL
public static final long	RULE_ALL
public static final long	RESOURCE_ALL
public static final long	TYPE_ALL
public static final long	GRID_ALL
public static final long	ATTRIBUTE_ALL
public static final long	ATTRIBUTE_UPDATE_MASK
public static final long	FORM_ALL
public static final long	FORM_UPDATE_ALL
public static final	FORM_DEPLOY

long	
public static final long	FORM_UNDEPLOY
public static final long	PROCESSDEF_ALL
public static final long	PROCESSDEF_UPDATE_ALL
public static final long	PROCESSDEF_DEPLOY
public static final long	PROCESSDEF_UNDEPLOY
public static final long	PROCESS_MANAGEMENT
public static final long	SCRIPT_ALL
public static final long	SCRIPT_VERSION_ALL
public static final long	VIEW_ALL
public static final long	VIEW_UPDATE_ALL
public static final long	VIEW_ELEMENT_ALL
public static final long	VIEW_ELEMENT_UPDATE_ALL
public static final long	VIEW_RUNTIME_TREE
public static final long	VIEW_DEPLOY
public static final long	VIEW_UNDEPLOY
public static final long	ISJOB_ALL
public static final long	MODELOCR_ALL
public static final long	SECURITY_PROFILE_ALL

public static final long	FOLDER_PROFILE_ALL
public static final long	FOLDER_TREE
public static final long	FOLDER_ALL
public static final long	FOLDER_FULL
public static final long	FOLDER_UPDATE_ALL
public static final long	FOLDER_UPDATE_FULL
public static final long	DOCUMENT_THUMB
public static final long	DOCUMENT_VC_SYSATTRS
public static final long	DOCUMENT_LIST
public static final long	DOCUMENT_ALL
public static final long	DOCUMENT_FULL
public static final long	VERSION_ALL
public static final long	TRASH_INFO
public static final long	NOTE_ALL
public static final long	SEARCHTEMPLATE_ALL
public static final long	SEARCHTEMPLATE_UPDATEALL
public static final long	PROCESS_ALL
public static final long	TASK_INBOX
public static final	TASK_ALL

long	
public static final long	USER_PROFILE_ALL
public static final long	JASY_ALL
public static final long	COMPOSITION_RULE_ALL
public static final long	COMPOSITION_RULE_VERSION_ALL
public static final long	FORM_DISPLAY_PROFILE_ALL
public static final long	FORM_VERSION_ALL
public static final long	PROCESSDEF_VERSION_ALL
public static final long	ISJOB_VERSION_ALL
public static final long	MODELOCR_VERSION_ALL
public static final long	VIEW_VERSION_ALL
public static final long	COMMENT_ALL
public static final long	COMMENT_UPDATE_ALL
public static final long	MAIL_TEMPLATE_ALL
public static final long	DEF_FILE_ID
public static final long	DOCUMENT_TEMPLATE_ALL
public static final long	FOR_FOLDERS
public static final long	FOR_PROCESSES
public static final long	FOR_DOCUMENTS

public static final long	FOR_SEARCH
public static final long	CLASSES_MASK
public static final long	IS_OBJECTS
public static final long	SHOW_SUBFOLDERS
public static final long	CLASS_SORT_MASK
public static final long	ORDER_ASC_1
public static final long	ORDER_DESC_1
public static final long	ORDER_ASC_2
public static final long	ORDER_DESC_2
public static final long	ORDER_ASC_3
public static final long	ORDER_DESC_3
public static final long	HAS_CLASSES
public static final long	HAS_DOCUMENTS
public static final long	IS_FOLDER
public static final long	IS_PROCESS
public static final long	ADD_ENABLED
public static final long	HAS_FULLTEXT
public static final long	SECURITY_MODE_MASK
public static final	SECURITY_MODE_NONE

long	
public static final long	SECURITY_MODE_SIMPLE
public static final long	SECURITY_MODE_MEDIUM
public static final long	SECURITY_MODE_COMPLEX
public static final long	IS_PUBLIC
public static final long	DATABASE_OFFLINE
public static final long	DATABASE_WORKFLOW_INACTIVE
public static final long	DATABASE_LIFECYCLE_INACTIVE
public static final long	DATABASE_LIFECYCLE_DEBUG
public static final long	DATABASE_UNAVAILABLE
public static final long	ENABLE_COMMENTS
public static final long	ENABLE_TRASH
public static final long	DATABASE_LUCENE_SIMPLESEARCH
public static final long	DATABASE_BUILD_LUCENE_SIMPLESEARCH
public static final long	DATABASE_LUCENE_CONTENTSEARCH
public static final long	DATABASE_BUILD_LUCENE_CONTENTSEARCH
public static final long	FULLTEXT_ACCENT_INSENSITIVE
public static final long	FULLTEXT_ACCENT_SENSITIVE
public static final long	FULLTEXT_ACCENT_BOTH

public static final long	IS_DEFAULT
public static final long	FOLDER_HAS_ATTRIBUTES
public static final long	FOLDER_HAS_NOTES
public static final long	FOLDER_HAS_CHILDREN
public static final long	HAS_COMMENTS
public static final long	HAS_NOTIFICATIONS
public static final long	FORM_ATTR_MANDATORY
public static final long	FORM_ATTR_LINKED
public static final long	FORM_ATTR_XNORMAL
public static final long	FORM_ATTR_XSIMPLE
public static final long	FORM_ATTR_XFULLPATH
public static final long	FORM_ATTR_XMULTPILE
public static final long	FORM_ATTR_XCATEGORY
public static final long	FORM_ATTR_XMULTILINE
public static final long	FORM_ATTR_XRADIO
public static final long	FORM_ATTR_XRADIO_INLINE
public static final long	FORM_ATTR_XMASK
public static final long	PROFILE_DEFAULT
public static final	PROFILE_ANONYMOUS

long	
public static final long	PROFILE_EXTERNAL_EMAIL
public static final long	DOCSTAT_ATTRIBUTES
public static final long	DOCSTAT_CDARCHIVED
public static final long	DOCSTAT_LOCK_OTHER
public static final long	DOCSTAT_LOCK_MYSELF
public static final long	DOCSTAT_LOCKED
public static final long	DOCSTAT_NOTES
public static final long	DOCSTAT_FULLTEXT
public static final long	DOCSTAT_HAS_VERSIONS
public static final long	DOCSTAT_SYS_ATTRIBUTES
public static final long	DOCSTAT_COPYOFDOC
public static final long	DOCSTAT_REFTODOC
public static final long	DOCSTAT_HAS_COMMENTS
public static final long	DOCSTAT_HAS_NOTIFICATIONS
public static final long	USER_IS_EXTERNAL_LDAP
public static final long	USER_IS_SERVER_ACCOUNT
public static final long	USER_LOCKED
public static final long	USER_CHANGEPASS

public static final long	USER_IS_EXTERNAL_EMAIL
public static final long	CAN_CANCEL
public static final long	OVERDUE_BY_TASK
public static final long	EXPIRING_BY_TASK
public static final long	DELETE_ON_FINISH
public static final long	DELETE_ON_CANCEL
public static final long	DELETE_ON_EXPIRE
public static final long	ATTR_ADVANCED_SEARCH
public static final long	ATTR_DATABASE_INDEX_PENDING
public static final long	ATTR_DATABASE_INDEX
public static final long	ATTR_CAPITAL_LETTERS
public static final long	VIRTUAL
public static final long	NO_CHILDREN_ALIVE
public static final long	HAS_VIRTUAL
public static final long	IS_LADRAD
public static final long	USER_PLANNER
public static final long	PROCESS_PLANNER
public static final long	VERSION_PLANNER
public static final	REORDER_LABEL_ASC

long	
public static final long	REORDER_LABEL_DESC
public static final long	EMAIL_NOTIFICATION
public static final long	CERT_ISSUED_BY
public static final long	CERT_SERVER
public static final long	CERT_FLAG_VALID_2
public static final long	CERT_IDENTITY_PREFIX
public static final long	CERT_IDENTITY_N0
public static final long	CERT_IDENTITY
public static final long	CERT_GEN_CACERT
public static final long	CERTIFICATE_ALL
public static final long	IS_JOB_SCAN
public static final long	IS_JOB_IMPORT
public static final long	IS_JOB_EMAIL
public static final long	IS_JOB_LAD_RAD
public static final long	IS_JOB_SIGN
public static final long	IS_JOB_SIGN_MANDATORY
public static final long	SIMPLE_DEBUG
public static final long	ENABLE_DOCUMENT_ACCESS

public static final long	HAS_PLUGIN_OPTIONS
public static final long	NOTIFICATION_DEFINITION_ALL
public static final long	NOTIFICATION_DEFINITION_FULL
public static final long	NOTIFICATION_ALL
public static final long	NOTIFICATION_FULL
public static final long	FAVORITE_ALL
public static final long	VC_FLAG_START
public static final long	VC_FLAG_MODIF
public static final long	VC_FLAG_AVIZ
public static final long	VC_FLAG_VALID
public static final long	VC_FLAG_PUBLIC
public static final long	VC_FLAG_GED
public static final long	VC_FLAG_BROADCAST
public static final long	PACKAGE_ALL
public static final long	SECURITY_PROFILE_FOR_WORKSPACE
public static final long	OPINION_SET_BY_SUPERVISOR
public static final long	KEEP_VALUES_ON_RESTART
public static final long	TYPE_LOCALIZED
public static final	ENABLE_TRANSLATIONS

long	
public static final long	MAIL_TEMPLATE_TYPE_MASK
public static final long	MAIL_TEMPLATE_FOR_PROCESS
public static final long	MAIL_TEMPLATE_FOR_VCYCLE
public static final long	MAIL_TEMPLATE_FOR_GENERAL_PURPOSE

Field detail :

```
public static final long ID = 1L
```

Bit 0. Specifies the 'id' field of com.siatel.defs.Identified implementing objects.

```
public static final long LABEL = 2L
```

Bit 1. Specifies the 'label' property of com.siatel.defs.Labeled implementing objects.

```
public static final long OWNER_ID = 4L
```

Bit 2. Specifies the 'owner' property of com.siatel.defs.Owned implementing objects.

```
public static final long FLAGS = 8L
```

Bit 3. Specifies the 'flags' property of com.siatel.defs.Flagged implementing objects.

```
public static final long DESCRIPTION = 16L
```

Bit 4. Specifies the 'description' property of com.siatel.defs.Described implementing objects.

```
public static final long DEPLOYED = 32L
```

Bit 5. Specifies the 'deployed' property of com.siatel.defs.Deployable implementing objects.

```
public static final long VERSION = 64L
```

Bit 6. Specifies the 'version' property of com.siatel.defs.Versioned implementing objects.

```
public static final long FORM_ID = 128L
```

Bit 7. Specifies the 'form id' property of com.siatel.defs.Viewable implementing objects.

```
public static final long ATTRIBUTES = 256L
```

Bit 8. Specifies the 'attributes' property of com.siatel.defs.Viewable implementing objects.

```
public static final long FILE_ID = 512L
```

Bit 9. Specifies the 'file id' property of com.siatel.defs.Realized implementing objects.

```
public static final long FILE_TYPE = 1024L
```

Bit 10. Specifies the 'file type' property of com.siatel.defs.RealizedEx implementing objects.

```
public static final long FILE_SIZE = 2048L
```

Bit 11. Specifies the 'file size' property of com.siatel.defs.RealizedEx implementing objects.

```
public static final long DATE_C = 4096L
```

Bit 12. Specifies the 'date created' property of com.siatel.defs.Dated implementing objects.

```
public static final long DATE_M = 8192L
```

Bit 13. Specifies the 'date modified' property of com.siatel.defs.Dated implementing objects.

```
public static final long DATE_A = 16384L
```

Bit 14. Specifies the 'date accessed' property of com.siatel.defs.DatedEx implementing objects.

```
public static final long ICON = 32768L
```

Bit 15. Specifies the 'icon' property of com.siatel.defs.Viewable implementing objects.

```
public static final long DATE_S = 65536L
```

Bit 16. Specifies the 'date started' property of com.siatel.defs.Timed implementing objects.

```
public static final long DATE_E = 131072L
```

Bit 17. Specifies the 'date ended' property of com.siatel.defs.Timed implementing objects.

```
public static final long DATE_D = 262144L
```

Bit 18. Specifies the 'date due' property of com.siatel.defs.Expirable implementing objects.

```
public static final long DATE_X = 524288L
```

Bit 19. Specifies the 'date expired' property of com.siatel.defs.Expirable implementing objects.

```
public static final long OVERDUE = 1048576L
```

Bit 20. Specifies the 'overdue' property of com.siatel.defs.Expirable implementing objects.

```
public static final long EXPIRED = 2097152L
```

Bit 21. Specifies the 'expired' property of com.siatel.defs.Expirable implementing objects.

```
public static final long REFCOUNT = 4194304L
```

Bit 22. Specifies the 'refcount' property of com.siatel.defs.Referenced implementing objects.

```
public static final long PRIORITY = 33554432L
```

Bit 25. Specifies the 'priority' property of com.siatel.defs.Prioritized implementing objects.

```
public static final long SECURITY_PROFILE = 34359738368L
```

Bit 35. Specifies the 'security profile' property of com.siatel.defs.Securable implementing objects.

```
public static final long RENDER_SECURITY = 68719476736L
```

Bit 36. Specifies the 'render security' operation which will fill the 'security mask' property of com.siatel.defs.Securable implementing objects.

```
public static final long UUID = 137438953472L
```

Bit 37. Specifies the 'uuid' property of com.siatel.defs.UUIDEnabled implementing objects.

```
public static final long TAGS = 274877906944L
```

Bit 38. Specifies the 'tags' property of com.siatel.defs.Tagged implementing objects.

```
public static final long DATE_AR = 549755813888L
```

Bit 39. Specifies the 'date archive' property of com.siatel.defs.Archive implementing objects.

```
public static final long EMAIL = 1099511627776L
```

User: Bit 40. Specifies the User#getEMail() property.

```
public static final long REAL_NAME = 2199023255552L
```

User: Bit 41. Specifies the User#getRealName() property.

```
public static final long DIRECTORY = 4398046511104L
```

User, Group, Role: Bit 42. Specifies the User#getExternalDirectoryId() (similar for groups and roles) property.

```
public static final long PASSWORD = 8796093022208L
```

User: Bit 43. Specifies the 'password' property for.

```
public static final long PRIVILEGES = 17592186044416L
```

User, Group, Role: Bit 44. Specifies the User#getPrivileges(), (similar for groups and roles) property.

```
public static final long PROFILE = 35184372088832L
```

User: Bit 45. Specifies the User#getProfileId() property.

```
public static final long ACTIVATED_BY = 4503599627370496L
```

User: Bit 52. Specifies the 'activated by' property.

```
public static final long ACTIVATED_AT = 9007199254740992L
```

User: Bit 53. Specifies the 'activation date-time ' property.

```
public static final long DEACTIVATED_BY = 18014398509481984L
```

User: Bit 54. Specifies the 'deactivated by' property.

```
public static final long DEACTIVATED_AT = 36028797018963968L
```

User: Bit 55. Specifies the 'deactivation date-time' property.

```
public static final long DATE_LI = 72057594037927936L
```

User: Bit 56. Specifies the User#getDateLogin() property.

```
public static final long DATE_LO = 144115188075855872L
```

User: Bit 57. Specifies the User#getDateLogout() property.

```
public static final long WORKING_PLANNER = 288230376151711744L
```

User: Bit 58. Specifies the 'working planner' property.

```
public static final long ACCESS_RESTRICTED_BY_WP = 576460752303423488L
```

User: Bit 59. Specifies the 'planner restricted access' property.

```
public static final long PARENT_ID = 1099511627776L
```

Folder, ViewElement, ClassificationElement, Comment: Bit 40. Specifies the 'parentId' property.

```
public static final long DESCRIPTION_URL = 1099511627776L
```

Database, View: Bit 40. Specifies the Database#getDescriptionUrl() (similar for View) property.

```
public static final long CONN_URL = 35184372088832L
```

Database: Bit 45. Specifies the Database#getConnUrl() property.

```
public static final long CRON_EXPRESSION = 70368744177664L
```

Database: Bit 46. Specifies the Database#getCronExpression() property.

```
public static final long SCRIPT_LANG = 1099511627776L
```

Script, Script2: Bit 40. Specifies the Script#getLang(), Script2#getLang() property.

```
public static final long SCRIPT_TYPE = 2199023255552L
```

Script, Script2: Bit 41. Specifies the Script#getType(), Script2#getType() property.

```
public static final long SAMPLE_FILE_ID = 1099511627776L
```

ModelOCR: Bit 40. Specifies the ModelOCR#getSampleFileId() property.

```
public static final long DISPLAYPROFILE_ID = 1099511627776L
```

Bit 40. Specifies the Form#getDisplayProfileId() property (similar for FormVersion).

```
public static final long LIFECYCLE_RULE_FILEID = 2199023255552L
```

Bit 41. Specifies the Form#getLifecycleRuleFileId() property (similar for FormVersion).

```
public static final long ARCHIVE_STORAGE_ID = 4398046511104L
```

Bit 42. Specifies the Form#getArchiveStorageId() property (similar for FormVersion).

```
public static final long FRAGMENT_FILE_ID = 8796093022208L
```

Bit 43. Specifies the Form#getFragmentFileId() property (similar for FormVersion).

```
public static final long HIERARCHY_FILE_ID = 17592186044416L
```

Bit 44. Specifies the Form#getHierarchyFileId() property (similar for FormVersion).

```
public static final long VERS_OBJ_COMMENT = 2251799813685248L
```

Commented: Bit 55. Specifies the Commented#getComment() property. Usually applies to design object versions.

```
public static final long VERS_OBJ_INDEX = 4503599627370496L
```

Numbered: Bit 56. Specifies the Numbered#getNumber() property. Usually applies to design object versions.

```
public static final long DATA = 18014398509481984L
```

Bit 54. Specifies the Attribute#getData() property.

```
public static final long VERSION_OF = 1099511627776L
```

ProcessDefinition: Bit 40: Specifies the ProcessDefinition#getVersionOfId() property

```
public static final long DELAY_X = 2199023255552L
```

ProcessDefinition: Bit 41: Specifies the ProcessDefinition#getDelayX() property

```
public static final long DELAY_D = 4398046511104L
```

ProcessDefinition: Bit 42: Specifies the ProcessDefinition#getDelayD() property

```
public static final long PROCESS_ID_PLANNER = 8796093022208L
```

ProcessDefinition: Bit 43: Specifies the ProcessDefinition#getPlannerId() property

```
public static final long DEPLOYMENT_FORM_ID = 17592186044416L
```

ProcessDefinition: Bit 44: Specifies the ProcessDefinition#getDeploymentFormId() property

```
public static final long DEFINITION_ID = 1099511627776L
```

Process: Bit 40: Specifies the Process#getDefinitionId() property

```
public static final long RUNTIME_STATE = 2199023255552L
```

Process: Bit 41: Specifies the Process#getRuntimeState() property

```
public static final long RUNTIME_FILE_ID = 4398046511104L
```

Process: Bit 42: Specifies the Process#getRuntimeFileId() property

```
public static final long PROCESS_STATUS = 8796093022208L
```

Process: Bit 43: Specifies the Process#getStatus() property

```
public static final long PROCESS_START_POINT = 17592186044416L
```

Process: Bit 44: Specifies the Process#getStartPoint() property

```
public static final long DRAFT_FORM_PAGE = 35184372088832L
```

Process: Bit 45: Specifies the Process#getDraftFormPage() property

```
public static final long PROFILE_ID = 140737488355328L
```

Folder, Process, ProcessDefinition: Bit 51: Specifies the Folder#getProfileId() property (similar for Process, ProcessDefinition)

```
public static final long TASK_STATUS = 1099511627776L
```

Task: Bit 40: Specifies the Task#getStatus() property.

```
public static final long TASK_ACTORID = 2199023255552L
```

Task: Bit 41: Specifies the Task#getActorId() property.

```
public static final long TASK_EXECUTORID = 4398046511104L
```

Task: Bit 42: Specifies the Task#getExecutorId() property.

```
public static final long TASK_LOCATION = 8796093022208L
```

Task: Bit 43: Specifies the Task#getLocation() property.

```
public static final long TASK_SUBJECT = 17592186044416L
```

Task: Bit 44: Specifies the Task#getSubject() property.

```
public static final long TASK_PROCESS_LABEL = 35184372088832L
```

Task: Bit 45: Specifies the Task#getProcessLabel() property.

```
public static final long TASK_FORM_PAGE = 70368744177664L
```

Task: Bit 46: Specifies the Task#getFormPage() property.

```
public static final long TASK_FORM_ID = 140737488355328L
```

Task: Bit 47: Specifies the Task#getFormId() property.

```
public static final long TIMETABLE = 1099511627776L
```

Planner: Bit 40: Specifies the Planner#getTimetable() property.

```
public static final long DATABASE_ID = 2199023255552L
```

Planner: Bit 41: Internal flag, it's purpose does not go beyond the persistence level. Developer note: if used in planners, it does not require explicit definition, it is added if the ID flag is present (the id of the planner depends on the database object that uses it)

```
public static final long DAY_TIMETABLE = 1099511627776L
```

PlannerDay: Bit 40: Specifies the PlannerDay#getTimetable() property.

```
public static final long DAY_DATE = 2199023255552L
```

PlannerDay: Bit 41: Specifies the PlannerDay#getDate() property.

```
public static final long DAY_RECURRENT = 4398046511104L
```

PlannerDay: Bit 42: Specifies the PlannerDay#isRecurrent() property.

```
public static final long RES_TYPE = 1099511627776L
```

Resource: Bit 40: Specifies the Resource#getType() property.

```
public static final long ENUM_MASK = 71776119061217280L
```

Enumeration sort flag. Mask used to filter out the enumeration flags.

```
public static final long GENERIC_ASC = 0L
```

Enumeration sort flag. Generic ascending. `0`.

```
public static final long GENERIC_DESC = 281474976710656L
```

Enumeration sort flag. Generic descending. `1`.

```
public static final long BY_DEFAULT_ASC = 0L
```

Enumeration sort flag. Use default criteria, ascending.

```
public static final long BY_DEFAULT_DESC = 281474976710656L
```

Enumeration sort flag. Use default criteria, descending.

```
public static final long BY_LABEL_ASC = 562949953421312L
```

Enumeration sort flag. Use 'label' criterion, ascending.

```
public static final long BY_LABEL_DESC = 844424930131968L
```

Enumeration sort flag. Use 'label' criterion, descending.

```
public static final long BY_DATEC_ASC = 1125899906842624L
```

Enumeration sort flag. Use 'dateC' criterion, ascending.

```
public static final long BY_DATEC_DESC = 1407374883553280L
```

Enumeration sort flag. Use 'dateC' criterion, descending.

```
public static final long BY_DATEM_ASC = 1688849860263936L
```

Enumeration sort flag. Use 'dateM' criterion, ascending.

```
public static final long BY_DATEM_DESC = 1970324836974592L
```

Enumeration sort flag. Use 'dateM' criterion, descending.

```
public static final long BY_DATES_ASC = 2251799813685248L
```

Task enumeration sort flag. Use 'dateS' as criterion, ascending. Developer note: The actual criteria is 'task.date_s asc, task.date_d asc, task.date_x asc, proc.priority asc'

```
public static final long BY_DATES_DESC = 2533274790395904L
```

Task enumeration sort flag. Use 'dateS' as criterion, descending. Developer note: The actual criteria is 'task.date_s desc, task.date_d asc, task.date_x asc, proc.priority asc'

```
public static final long BY_DATED_ASC = 2814749767106560L
```

Task enumeration sort flag. Use 'dateD' as criterion, ascending. Developer note: The actual criteria is 'task.date_d asc, task.date_x asc, proc.priority asc, task.date_s desc'

```
public static final long BY_DATED_DESC = 3096224743817216L
```

Task enumeration sort flag. Use 'dateD' as criterion, descending. Developer note: The actual criteria is 'task.date_d desc, task.date_x asc, proc.priority asc, task.date_s desc'

```
public static final long BY_PRIO_ASC = 3377699720527872L
```

Task enumeration sort flag. Use 'priority' as criterion, ascending. Developer note: The actual criteria is 'proc.priority desc, task.date_d asc, task.date_x asc, task.date_s desc'

```
public static final long BY_PRIO_DESC = 3659174697238528L
```

Task enumeration sort flag. Use 'priority' as criterion, ascending. Developer note: The actual criteria is 'proc.priority asc, task.date_d asc, task.date_x asc, task.date_s desc'

```
public static final long COMMENT_CONTENT = 2199023255552L
```

Comment: Bit 41: Specifies the Comment#getContent() property.

```
public static final long OWNER_NAME = 4398046511104L
```

Comment: Bit 42: Specifies the Comment#getOwnerName() property.

```
public static final long HIGHER_ID = 175921860444160L
```

Role: Bit 45 and 47: Specifies the Role#getHigherId() property.

```
public static final long USER_ALL = 10809331798014812 47L
```

ID | LABEL | OWNER_ID | FLAGS | DESCRIPTION | DATE_C | DATE_M | PASSWORD | PRIVILEGES | EMAIL | REAL_NAME | DIRECTORY | DATE_LI | DATE_LO | WORKING_PLANNER | ACCESS_RESTRICTED_BY_WP

```
public static final long USER_UPDATE_ALL = 86471641722258 2298L
```

LABEL | FLAGS | DESCRIPTION | DATE_M | PRIVILEGES | EMAIL | REAL_NAME | DIRECTORY | WORKING_PLANNER | ACCESS_RESTRICTED_BY_WP

```
public static final long GROUP_ALL = 21990232567839L
```

ID | LABEL | OWNER_ID | FLAGS | DESCRIPTION | DATE_C | DATE_M | PRIVILEGES | DIRECTORY

```
public static final long GROUP_UPDATE_ALL = 21990232563739L
```

ID | LABEL | FLAGS | DESCRIPTION | DATE_M | PRIVILEGES | DIRECTORY

```
public static final long ROLE_ALL = 197912093011999L
```

ID | LABEL | OWNER_ID | FLAGS | DESCRIPTION | DATE_C | DATE_M | PRIVILEGES | DIRECTORY | HIGHER_ID

```
public static final long ROLE_UPDATE_ALL = 197912093007899L
```

ID | LABEL | FLAGS | DESCRIPTION | DATE_M | PRIVILEGES | DIRECTORY |

HIGHER_ID

```
public static final long PORTAL_ALL = 12831L
```

ID | LABEL | OWNER_ID | FLAGS | DESCRIPTION | DATE_C | DATE_M | FILE_ID

```
public static final long PORTAL_UPDATE_ALL = 8731L
```

ID | LABEL | FLAGS | DESCRIPTION | DATE_M | FILE_ID.

```
public static final long PLANNER_ALL = 3298534895679L
```

ID | LABEL | OWNER_ID | FLAGS | DESCRIPTION | DATE_C | DATE_M | DEPLOYED | TIMETABLE | DATABASE_ID.

```
public static final long PLANNER_DAY_ALL = 7696581394441L
```

ID | FLAGS | DAY_TIMETABLE | DAY_DATE | DAY_RECURRENT.

```
public static final long DATABASE_ALL = 18154036486221855L
```

ID | LABEL | OWNER_ID | FLAGS | DESCRIPTION | DATE_C | DATE_M | ADMINISTRATOR_ID | DESCRIPTION_URL | ANONYMOUS_COMMANDS | RESTRICTED_COMMANDS | FULL_COMMANDS | CONN_URL | CRON_EXPRESSION | DATA

```
public static final long DATABASE_UPDATE_ALL = 18154036486217755L
```

ID | LABEL | FLAGS | DESCRIPTION | DATE_M | ADMINISTRATOR_ID | DESCRIPTION_URL | CONN_URL | ANONYMOUS_COMMANDS | RESTRICTED_COMMANDS | FULL_COMMANDS | CRON_EXPRESSION | DATA

```
public static final long STORAGE_ALL = 12319L
```

ID | LABEL | OWNER_ID | FLAGS | DESCRIPTION | DATE_C | DATE_M.

```
public static final long RULE_ALL = 9L
```

ID | FLAGS

```
public static final long RESOURCE_ALL = 1099511640607L
```

ID | LABEL | OWNER_ID | FLAGS | DESCRIPTION | FILE_ID | RES_TYPE | DATE_C | DATE_M

```
public static final long TYPE_ALL = 137438966303L
```

ID | LABEL | OWNER_ID | FLAGS | DESCRIPTION | FILE_ID | DATE_C | DATE_M | PACKAGE

```
public static final long GRID_ALL = 137438966303L
```

ID | LABEL | OWNER_ID | FLAGS | DESCRIPTION | FILE_ID | DATE_C | DATE_M | PACKAGE

```
public static final long ATTRIBUTE_ALL = 18014536485318687L
```

ID | LABEL | OWNER_ID | FLAGS | DESCRIPTION | TYPE_ID | DATE_C | DATE_M | DATA | PACKAGE

```
public static final long ATTRIBUTE_UPDATE_MASK = 536870938L
```

LABEL | FLAGS | DESCRIPTION | TYPE_ID

```
public static final long FORM_ALL = 342351843620 47L
```

ID | LABEL | OWNER_ID | FLAGS | DESCRIPTION | DATE_C | DATE_M | DEPLOYED | FILE_ID | DEPLOYMENT_ID | DEPLOYMENT_STAMP | ICON | DISPLAYPROFILE_ID | LIFECYCLE_RULE_FILEID | ARCHIVE_STORAGE_ID | FRAGMENT_FILE_ID | HIERARCHY_FILE_ID | PACKAGE

```
public static final long FORM_UPDATE_ALL = 45599L
```

ID | LABEL | OWNER_ID | FLAGS | DESCRIPTION | DATE_C | DATE_M | FILE_ID | ICON

```
public static final long FORM_DEPLOY = 12884901921L
```

ID | DEPLOYED | DEPLOYMENT_ID | DEPLOYMENT_STAMP

```
public static final long FORM_UNDEPLOY = 33L
```

ID | DEPLOYED

```
public static final long PROCESSDEF_ALL = 280560203215551L
```

ID | LABEL | OWNER_ID | FLAGS | FORM_ID | DESCRIPTION | DATE_C | DATE_M | FILE_ID | DEPLOYED | DEPLOYMENT_ID | DEPLOYMENT_STAMP | DELAY_D | DELAY_X | PRIORITY | REFCOUNT | VERSION_OF | MANAGER_ID | PROCESS_ID_PLANNER | DEPLOYMENT_FORM_ID | SECURITY_PROFILE | PROFILE_ID | DELAY_D_ATTR_ID | DELAY_X_ATTR_ID | PACKAGE

```
public static final long PROCESSDEF_UPDATE_ALL = 261718177526431L
```

ID | LABEL | OWNER_ID | FLAGS | DESCRIPTION | DATE_C | DATE_M | FILE_ID | MANAGER_ID | FORM_ID | PROCESS_ID_PLANNER | PRIORITY | ICON | DELAY_D | DELAY_X | SECURITY_PROFILE | PROFILE_ID

```
public static final long PROCESSDEF_DEPLOY = 17605070946337L
```

ID | DEPLOYED | DEPLOYMENT_ID | DEPLOYMENT_STAMP | DEPLOYMENT_FORM_ID

```
public static final long PROCESSDEF_UNDEPLOY = 33L
```

ID | DEPLOYED

```
public static final long PROCESS_MANAGEMENT = 9895659025055L
```

ID | LABEL | OWNER_ID | FLAGS | DESCRIPTION | DATE_C | DATE_M | FILE_ID | MANAGER_ID | FORM_ID | PRIORITY | PROCESS_ID_PLANNER | ICON | DATE_S | DATE_D | DATE_X | OVERDUE | EXPIRED | PROCESS_STATUS | DEFINITION_ID

```
public static final long SCRIPT_ALL = 3448858751551L
```

ID | LABEL | OWNER_ID | FLAGS | DESCRIPTION | DATE_C | DATE_M | FILE_ID | SCRIPT_LANG | SCRIPT_TYPE | DEPLOYED | DEPLOYMENT_ID | DEPLOYMENT_STAMP | PACKAGE

```
public static final long SCRIPT_VERSION_ALL = 6758848299807295L
```

SCRIPT_ALL | VERS_OBJ_COMMENT | VERS_OBJ_INDEX

```
public static final long VIEW_ALL = 1236950626367L
```

ID | LABEL | OWNER_ID | FLAGS | DATE_C | DATE_M | DESCRIPTION | DEPLOYED | DESCRIPTION_URL | ICON | PACKAGE

```
public static final long VIEW_UPDATE_ALL = 1099511672863L
```

ID | LABEL | OWNER_ID | FLAGS | DATE_C | DATE_M | DESCRIPTION | DESCRIPTION_URL | ICON

```
public static final long VIEW_ELEMENT_ALL = 1099511640095L
```

ID | LABEL | OWNER_ID | FLAGS | DATE_C | DATE_M | DESCRIPTION | DISPLAYPROFILE_ID

```
public static final long VIEW_ELEMENT_UPDATE_ALL = 8219L
```

ID | LABEL | FLAGS | DATE_M | DESCRIPTION

```
public static final long VIEW_RUNTIME_TREE = 11L
```

ID | LABEL | FLAGS

```
public static final long VIEW_DEPLOY = 33L
```

ID | DEPLOYED

```
public static final long VIEW_UNDEPLOY = 33L
```

ID | DEPLOYED

```
public static final long ISJOB_ALL = 141733933631L
```

ID | LABEL | OWNER_ID | FLAGS | DESCRIPTION | DATE_C | DATE_M | DEPLOYED | FILE_ID | DEPLOYMENT_ID | PACKAGE

```
public static final long MODELOCR_ALL = 141733933631L
```

ID | LABEL | OWNER_ID | FLAGS | DESCRIPTION | DATE_C | DATE_M | DEPLOYED | FILE_ID | DEPLOYMENT_ID | PACKAGE

```
public static final long SECURITY_PROFILE_ALL = 137438965823L
```

ID | LABEL | DESCRIPTION | DATE_C | DATE_M | OWNER_ID | FLAGS | CUSTOM_SECURITY_PROCEDURE | PACKAGE

```
public static final long FOLDER_PROFILE_ALL = 139775415693343L
```

ID | LABEL | DESCRIPTION | DATE_C | DATE_M | OWNER_ID | FLAGS | FP_COLUMNS | FP_OVERRIDDEN_OPTIONS | FP_CHILD_FORM_IDS | FP_CHILD_DEF_FORM_ID | FP_CHILD_DEF_PROFILE_ID | FP_DOC_FORM_IDS | FP_DOC_DEF_FORM_ID | PACKAGE

```
public static final long FOLDER_TREE = 32779L
```

ID | LABEL | FLAGS | ICON

```
public static final long FOLDER_ALL = 18155178947768463L
```

ID | LABEL | FLAGS | ICON | OWNER_ID | FORM_ID | FORM_PAGE | DATE_C | DATE_M | MODIFY_BY | DATE_A | DATE_S | DATE_E | SECURITY_PROFILE | PROFILE_ID

```
public static final long FOLDER_FULL = 18155178947768719L
```

ID | LABEL | FLAGS | ICON | OWNER_ID | FORM_ID | FORM_PAGE | DATE_C | DATE_M | MODIFY_BY | DATE_A | DATE_S | DATE_E | SECURITY_PROFILE | PROFILE_ID | ATTRIBUTES

```
public static final long FOLDER_UPDATE_ALL = 140780438253706L
```

LABEL | FLAGS | DATE_C | DATE_M | DATE_A | FORM_ID | FORM_PAGE | STATE_ID | DATE_S | DATE_E | SECURITY_PROFILE | PROFILE_ID

```
public static final long FOLDER_UPDATE_FULL = 140780438253962L
```

LABEL | FLAGS | DATE_C | DATE_M | DATE_A | FORM_ID | FORM_PAGE | ATTRIBUTES | STATE_ID | DATE_S | DATE_E | SECURITY_PROFILE | PROFILE_ID

```
public static final long DOCUMENT_THUMB = 39594266187L
```

ID | LABEL | FLAGS | VERSION | FILE_ID | FILE_TYPE | FILE_SIZE | ICON | LOCKED_BY | DATE_L | VALID_MILLIS | LOCK_TYPE | SECURITY_PROFILE

```
public static final long DOCUMENT_VC_SYSATTRS = 2053641430081732608L
```

Flag.REQUESTED_BY | Flag.VALIDATED_BY | Flag.DATE_VALIDATION | Flag.DATE_PUBLISHED | Flag.DATE_EXPIRED | Flag.VALIDITY

```
public static final long DOCUMENT_LIST = 18014712984813131L
```

ID | LABEL | FLAGS | VERSION | FILE_ID | FILE_TYPE | FILE_SIZE | ICON | LOCKED_BY | DATE_L | VALID_MILLIS | LOCK_TYPE | SECURITY_PROFILE | DATE_C | DATE_M | OCR_FILE_ID | LINKED_ID | TAGS

```
public static final long DOCUMENT_ALL = 20716561516564968 47L
```

ID | LABEL | FLAGS | VERSION | FILE_ID | FILE_TYPE | FILE_SIZE | ICON | LOCKED_BY | DATE_L | VALID_MILLIS | LOCK_TYPE | SECURITY_PROFILE | DATE_C | DATE_M | OCR_FILE_ID | LINKED_ID | TAGS | OWNER_ID | FORM_ID | FORM_PAGE | DATE_A

```
public static final long DOCUMENT_FULL = 2071656151656497103L
```

ID | LABEL | FLAGS | VERSION | FILE_ID | FILE_TYPE | FILE_SIZE | ICON | LOCKED_BY | DATE_L | VALID_MILLIS | LOCK_TYPE | SECURITY_PROFILE | DATE_C | DATE_M | OCR_FILE_ID | LINKED_ID | TAGS | OWNER_ID | FORM_ID | FORM_PAGE | DATE_A | ATTRIBUTES

```
public static final long VERSION_ALL = 2251799813717583L
```

ID | LABEL | OWNER_ID | FLAGS | DATE_C | DATE_M | DATE_A | FILE_ID | FILE_SIZE | FILE_TYPE | VERSION | VERS_OBJ_COMMENT

```
public static final long TRASH_INFO = 4573969445289984L
```

DELETED | DATE_DEL | DELETE_BY

```
public static final long NOTE_ALL = 29189L
```

ID | OWNER_ID | DATE_C | DATE_M | DATE_A | FILE_ID

```
public static final long SEARCHTEMPLATE_ALL = 67121695L
```

LABEL | FLAGS | DATE_C | DATE_M | DESCRIPTION | ID | OWNER_ID | FILE_ID | PRESENTATION

```
public static final long SEARCHTEMPLATE_UPDATEALL = 67117595L
```

LABEL | FLAGS | DATE_M | DESCRIPTION | ID | FILE_ID | PRESENTATION

```
public static final long PROCESS_ALL = 18294816982922127L
```

ID | LABEL | OWNER_ID | FLAGS | DATE_C | DATE_M | DATE_A | DATE_D | DATE_X | DATE_E | DATE_S | FILE_ID | FORM_ID | FORM_PAGE | ATTRIBUTES | PRIORITY | REFCOUNT | OVERDUE | EXPIRED | PROCESS_STATUS | RUNTIME_FILE_ID | RUNTIME_STATE | DEFINITION_ID | MANAGER_ID | PROCESS_ID_PLANNER | PROCESS_START_POINT | DRAFT_FORM_PAGE | SECURITY_PROFILE | PROFILE_ID

```
public static final long TASK_INBOX = 280375500537867L
```

ID | LABEL | FLAGS | DATE_S | DATE_D | DATE_X | PRIORITY | OVERDUE | TASK_ACTORID | TASK_EXECUTORID | TASK_STATUS | TASK_LOCATION | TASK_PROCESS_LABEL | TASK_SUBJECT | TASK_FORM_PAGE | TASK_FORM_ID

```
public static final long TASK_ALL = 280375517315083L
```

ID | LABEL | FLAGS | DATE_S | DATE_D | DATE_X | PRIORITY | OVERDUE | MANAGER_ID | TASK_ACTORID | TASK_EXECUTORID | TASK_STATUS | TASK_LOCATION | TASK_PROCESS_LABEL | TASK_SUBJECT | TASK_FORM_PAGE | TASK_FORM_ID

```
public static final long USER_PROFILE_ALL = 1099511640095L
```

ID | LABEL | OWNER_ID | FLAGS | DESCRIPTION | DATE_C | DATE_M | PARENT_ID

```
public static final long JASY_ALL = 69269232758799L
```

ID | LABEL | OWNER_ID | FLAGS | DATE_C | DATE_M | DATE_S | DATE_E | JASY_TYPE | JASY_STATUS | JASY_PROG_PERCENT | JASY_PROG_STATUS | JASY_EMAIL | JASY_CONFIG

```
public static final long COMPOSITION_RULE_ALL = 150323868223L
```

ID | LABEL | DESCRIPTION | DATE_C | DATE_M | OWNER_ID | FLAGS | DEPLOYED | FILE_ID | DEPLOYMENT_ID | DEPLOYMENT_STAMP| PACKAGE

```
public static final long
COMPOSITION_RULE_VERSION_ALL = 6755399441068575L
```

ID | LABEL | DESCRIPTION | DATE_C | DATE_M | OWNER_ID | FLAGS | FILE_ID | VERS_OBJ_COMMENT | VERS_OBJ_INDEX

```
public static final long FORM_DISPLAY_PROFILE_ALL = 140737488355457L
```

ID | FORM_ID | PROFILE_ID

```
public static final long FORM_VERSION_ALL = 6789634625417791L
```

FORM_ALL | VERS_OBJ_COMMENT | VERS_OBJ_INDEX

```
public static final long PROCESSDEF_VERSION_ALL = 7017130507645631L
```

ID | LABEL | OWNER_ID | FLAGS | FORM_ID | DESCRIPTION | DATE_C | DATE_M | FILE_ID | DEPLOYED | DEPLOYMENT_ID | DEPLOYMENT_STAMP | DELAY_D | DELAY_X | PRIORITY | REFCOUNT | MANAGER_ID | PROCESS_ID_PLANNER | SECURITY_PROFILE | PROFILE_ID | VERS_OBJ_COMMENT | VERS_OBJ_INDEX

```
public static final long ISJOB_VERSION_ALL = 6755399441068607L
```

ID | LABEL | OWNER_ID | FLAGS | DESCRIPTION | DATE_C | DATE_M | DEPLOYED | FILE_ID | VERS_OBJ_COMMENT | VERS_OBJ_INDEX

```
public static final long MODELOCR_VERSION_ALL = 6755399441068607L
```

ID | LABEL | OWNER_ID | FLAGS | DESCRIPTION | DATE_C | DATE_M | DEPLOYED | FILE_ID | VERS_OBJ_COMMENT | VERS_OBJ_INDEX

```
public static final long VIEW_VERSION_ALL = 6756498952695871L
```

ID | LABEL | OWNER_ID | FLAGS | DATE_C | DATE_M | DESCRIPTION | DEPLOYED | DESCRIPTION_URL | VERS_OBJ_COMMENT | VERS_OBJ_INDEX

```
public static final long COMMENT_ALL = 7696581406735L
```

ID | LABEL | FLAGS | DATE_C | DATE_M | OWNER_ID | OWNER_NAME | PARENT_ID | COMMENT_CONTENT

```
public static final long COMMENT_UPDATE_ALL = 6597069774858L
```

LABEL | FLAGS | DATE_M | OWNER_NAME | COMMENT_CONTENT

```
public static final long MAIL_TEMPLATE_ALL = 137438966303L
```

ID | LABEL | DESCRIPTION | DATE_C | DATE_M | OWNER_ID | FLAGS | FILE_ID

```
public static final long DEF_FILE_ID = 1099511627776L
```

Specifies the DocumentTemplate#getDefFileId() property).

```
public static final long DOCUMENT_TEMPLATE_ALL = 1236950594079L
```

ID | LABEL | DESCRIPTION | DATE_C | DATE_M | OWNER_ID | FLAGS | FILE_ID | DEF_FILE_ID | PACKAGE

```
public static final long FOR_FOLDERS = 16777216L
```

Form#getFlags(): Specifies form can be set on folders. Can be combined with #FOR_PROCESSES|#FOR_DOCUMENTS|#FOR_SEARCH in this case.

ViewElement#getFlags(): Specifies view element will render on folders. Can be combined with #FOR_PROCESSES but not #FOR_DOCUMENTS in this case.

```
public static final long FOR_PROCESSES = 33554432L
```

Form#getFlags(): Specifies form can be set on processes. Can be combined with #FOR_FOLDERS|#FOR_DOCUMENTS|#FOR_SEARCH in this case.

ViewElement#getFlags(): Specifies view element will render on processes. Can be combined with #FOR_FOLDERS but not #FOR_DOCUMENTS in this case.

```
public static final long FOR_DOCUMENTS = 67108864L
```

Form#getFlags(): Specifies form can be set on documents. Can be combined with #FOR_PROCESSES|#FOR_FOLDERS|#FOR_SEARCH in this case.

ViewElement#getFlags(): Specifies view element will render on folders. Cannot be combined with neither #FOR_PROCESSES nor #FOR_FOLDERS in this case.

```
public static final long FOR_SEARCH = 134217728L
```

Form#getFlags(): Specifies form can be set on folders. Can be combined with #FOR_PROCESSES|#FOR_DOCUMENTS|#FOR_SEARCH in this case.

```
public static final long CLASSES_MASK = -1895825408L
```

Bit-mask for ViewElement#getFlags()

```
public static final long IS_OBJECTS = 134217728L
```

ViewElement#getFlags(): specifies that view element renders to actual objects (folders, document, processes)

```
public static final long SHOW_SUBFOLDERS = -2147483648L
```

ViewElement#getFlags(): specifies that rendered class element shall display the real folders in the tree. Applies only to view elements containing 'FOR_FOLDERS', otherwise this should be signaled as error by compiler

```
public static final long CLASS_SORT_MASK = 1879048192L
```

Bit mask for criteria sorting specified in ViewElement#getFlags()

```
public static final long ORDER_ASC_1 = 0L
```

ViewElement#getFlags(): order by first unicity criterion, ascending (by default)

```
public static final long ORDER_DESC_1 = 268435456L
```

ViewElement#getFlags(): order by first unicity criterion, descending

```
public static final long ORDER_ASC_2 = 0L
```

ViewElement#getFlags(): order by second unicity criterion, ascending (by default)

```
public static final long ORDER_DESC_2 = 536870912L
```

ViewElement#getFlags(): order by second unicity criterion, descending

```
public static final long ORDER_ASC_3 = 0L
```

ViewElement#getFlags(): order by third unicity criterion, ascending (by default)

```
public static final long ORDER_DESC_3 = 1073741824L
```

ViewElement#getFlags(): order by third unicity criterion, descending (by default)

```
public static final long HAS_CLASSES = 4294967296L
```

ClassificationElement#getFlags(): Specifies item has child elements to render.

```
public static final long HAS_DOCUMENTS = 8589934592L
```

ClassificationElement#getFlags(): Specifies item has document list render.

```
public static final long IS_FOLDER = 17179869184L
```

ClassificationElement#getFlags(): Specifies item is actual folder.

```
public static final long IS_PROCESS = 34359738368L
```

ClassificationElement#getFlags(): Specifies item is actual process.

```
public static final long ADD_ENABLED = 68719476736L
```

ClassificationElement#getFlags(): Specifies documents can be added in item.

```
public static final long HAS_FULLTEXT = 1L
```

Database#getFlags(): Database has fulltext capability (bit 0)

```
public static final long SECURITY_MODE_MASK = 6L
```

Database#getFlags(): 2-bit mask for the database security support (bits 1 and 2)

```
public static final long SECURITY_MODE_NONE = 0L
```

Database#getFlags(): Security mode 0, database has no security support

```
public static final long SECURITY_MODE_SIMPLE = 2L
```

Database#getFlags(): Security mode 2 (`1 << 1`), database has simple security support

```
public static final long SECURITY_MODE_MEDIUM = 4L
```

Database#getFlags(): Security mode 4 (`2 << 1`), database has medium security support

```
public static final long SECURITY_MODE_COMPLEX = 6L
```

Database#getFlags(): Security mode 6 (`3 << 1`), database has complex security support

```
public static final long IS_PUBLIC = 8L
```

Database#getFlags(): database is public (bit 3).

View#getFlags(): view is public.

SearchTemplate#getFlags(): search template is public.

```
public static final long DATABASE_OFFLINE = 16384L
```

Database#getFlags(): Database offline (bit 13)

```
public static final long DATABASE_WORKFLOW_INACTIVE = 32768L
```

Database#getFlags(): Workflow paused on database (bit 14)

```
public static final long DATABASE_LIFECYCLE_INACTIVE = 131072L
```

Database#getFlags(): Lifecycle jobs paused on database (bit 15)

```
public static final long DATABASE_LIFECYCLE_DEBUG = 262144L
```

Database#getFlags(): Lifecycle is in debug mode, as in it does not actually archive/delete objects, only dumps rendered result to console (bit 16)

```
public static final long DATABASE_UNAVAILABLE = 4611686018427387904L
```

Database#getFlags(): Database schema is external and is unavailable (bit 17)

```
public static final long ENABLE_COMMENTS = 536870912L
```

Database#getFlags(): specifies if forum is enabled on a database

```
public static final long ENABLE_TRASH = 17179869184L
```

Database#getFlags(): specifies if recycle bin is enabled on a database

```
public static final long
DATABASE_LUCENE_SIMPLESEARCH = 72057594037927936L
```

Database#getFlags(): Database has lucene simple search capability

```
public static final long
DATABASE_BUILD_LUCENE_SIMPLESEARCH = 144115188075855872L
```

Database#getFlags(): lcn_ss reindexing database in progress...

```
public static final long
DATABASE_LUCENE_CONTENTSEARCH = 288230376151711744L
```

Database#getFlags(): Database has lucene content search capability

```
public static final long
DATABASE_BUILD_LUCENE_CONTENTSEARCH = 576460752303423488L
```

Database#getFlags(): lcn_content reindexing database in progress...

```
public static final long FULLTEXT_ACCENT_INSENSITIVE = 4503599627370496L
```

Database#getFlags(): accent insensitive solr index (index type)

```
public static final long FULLTEXT_ACCENT_SENSITIVE = 9007199254740992L
```

Database#getFlags(): accent sensitive solr index (index type)

```
public static final long FULLTEXT_ACCENT_BOTH = 13510798882111488L
```

Database#getFlags(): accent sensitive and insensitive solr index (query time selection) (index type)

```
public static final long IS_DEFAULT = 32L
```

Planner#getFlags(): specifies this planner is the server default planner.

Portal#getFlags(): specifies this portal is the server default portal.

```
public static final long FOLDER_HAS_ATTRIBUTES = 1L
```

Folder#getFlags(): specifies folder has custom attributes which are not empty

```
public static final long FOLDER_HAS_NOTES = 2L
```

Folder#getFlags(): specifies folder has notes, useful to add a 'note' symbol to the folder icon, wherever it is displayed.

```
public static final long FOLDER_HAS_CHILDREN = 4L
```

Folder#getFlags(): specifies folder has sub-folders, useful to add a '+' sign to the folder tree item.

```
public static final long HAS_COMMENTS = 32768L
```

Folder#getFlags(): specifies folder has comments, useful to add a 'comments' symbol to the folder icon, wherever it is displayed.

Document#getFlags(): specifies document has comments, useful to add a 'comments' symbol to the document icon, wherever it is displayed.

Process#getFlags(): specifies process has comments, useful to add a 'comments' symbol to the process icon, wherever it is displayed.

```
public static final long HAS_NOTIFICATIONS = 262144L
```

Folder#getFlags(): specifies folder has notifications, useful to add a 'notification' symbol to the folder icon, wherever it is displayed.

Document#getFlags(): specifies document has notifications, useful to add a 'notification' symbol to the document icon, wherever it is displayed.

Process#getFlags(): specifies process has notifications, useful to add a 'notification' symbol to the process icon, wherever it is displayed.

```
public static final long FORM_ATTR_MANDATORY = 1L
```

(Long) Assignment#getData(): specifies attribute is mandatory in context of form.

```
public static final long FORM_ATTR_LINKED = 2L
```

(Long) Assignment#getData(): specifies attribute is linked to another attribute in context of form.

```
public static final long FORM_ATTR_XNORMAL = 0L
```

(Long) Assignment#getData(): specifies the list attribute is normal, has no special behavior.

```
public static final long FORM_ATTR_XSIMPLE = 16L
```

(Long) Assignment#getData(): specifies the list attribute is plain, as in the hierarchy value is irrelevant, all items are considered on same level and are represented as a linear list.

```
public static final long FORM_ATTR_XFULLPATH = 32L
```

(Long) Assignment#getData(): specifies the list attribute is 'full path', as in items are specified with all parent items in the hierarchy

```
public static final long FORM_ATTR_XMULTPILE = 48L
```

(Long) Assignment#getData(): specifies the list attribute is 'multiple selection', as in the attribute value

can contain more than one value from the list.

```
public static final long FORM_ATTR_XCATEGORY = 64L
```

(Long) Assignment#getData(): specifies the list attribute is 'category type', as in the attribute value can contain any value from the list except first level values.

```
public static final long FORM_ATTR_XMULTILINE = 80L
```

(Long) Assignment#getData(): bit mask for the above properties, since they are exclusive.

```
public static final long FORM_ATTR_XRADIO = 96L
```

(Long) Assignment#getData(): specifies the list attribute is a list of radio buttons

```
public static final long FORM_ATTR_XRADIO_INLINE = 112L
```

(Long) Assignment#getData(): specifies the list attribute is a list of inline radio buttons

```
public static final long FORM_ATTR_XMASK = 240L
```

(Long) Assignment#getData(): bit mask for the above properties, since they are exclusive.

```
public static final long PROFILE_DEFAULT = 1L
```

UserProfile#getFlags(): specifies it is the default user profile.

```
public static final long PROFILE_ANONYMOUS = 2L
```

UserProfile#getFlags(): specifies it is the user profile for anonymous connections

```
public static final long PROFILE_EXTERNAL_EMAIL = 4L
```

UserProfile#getFlags(): specifies it is the user profile for external email users connections

```
public static final long DOCSTAT_ATTRIBUTES = 1L
```

Document#getFlags(): document has custom attributes which are not empty.

```
public static final long DOCSTAT_CDARCHIVED = 4L
```

Document#getFlags(): document file is stored on optical media.

DocumentVersion#getFlags(): document file is stored on optical media.

```
public static final long DOCSTAT_LOCK_OTHER = 8L
```

Document#getFlags(): document is locked to user other than one accessing the document. This flag is connection dependent.

```
public static final long DOCSTAT_LOCK_MYSELF = 16L
```

Document#getFlags(): document is locked to user accessing the document. This flag is connection dependent.

```
public static final long DOCSTAT_LOCKED = 32L
```

Document#getFlags(): document is locked for changes.

```
public static final long DOCSTAT_NOTES = 64L
```

Document#getFlags(): document has notes.

```
public static final long DOCSTAT_FULLTEXT = 256L
```

Document#getFlags(): document has indexed content.

```
public static final long DOCSTAT_HAS_VERSIONS = 512L
```

Document#getFlags(): document has versions.

```
public static final long DOCSTAT_SYS_ATTRIBUTES = 1024L
```

Document#getFlags(): search hit in document's system attributes. This flag is relevant in the context of search.

```
public static final long DOCSTAT_COPYOFDOC = 2048L
```

Document#getFlags(): document is copy of another document specified by Document#getLinkedId().

```
public static final long DOCSTAT_REFTODOC = 4096L
```

Document#getFlags(): document is a reference of another document specified by Document#getLinkedId().

```
public static final long DOCSTAT_HAS_COMMENTS = 32768L
```

Document#getFlags(): document has comments

```
public static final long DOCSTAT_HAS_NOTIFICATIONS = 262144L
```

Document#getFlags(): document has notifications

```
public static final long USER_IS_EXTERNAL_LDAP = 1L
```

User#getFlags(): user is imported from an external directory.

```
public static final long USER_IS_SERVER_ACCOUNT = 2L
```

User#getFlags(): this is a sever account

```
public static final long USER_LOCKED = 4294967296L
```

User#getFlags(): user's account is locked (explicitly by admin or due to successive failed login attempts)

```
public static final long USER_CHANGEPASS = 8589934592L
```

User#getFlags(): user must change pass on next login

```
public static final long USER_IS_EXTERNAL_EMAIL = 17179869184L
```

User#getFlags(): user is external (identified by email)

```
public static final long CAN_CANCEL = 256L
```

Task#getFlags(): task can be canceled by user

```
public static final long OVERDUE_BY_TASK = 512L
```

Process#getFlags(): process is marked overdue because of task who is past it's due date

```
public static final long EXPIRING_BY_TASK = 1024L
```

Process#getFlags(): process is marked as expiring because of task who is close (less than 1 day) to it's expiry date.

```
public static final long DELETE_ON_FINISH = 4096L
```

Process#getFlags(): process should be automatically deleted when it is finished.

ProcessDefinition#getFlags(): instances should be automatically deleted when they are finished.

```
public static final long DELETE_ON_CANCEL = 8192L
```

Process#getFlags(): process should be automatically deleted when it is canceled.

ProcessDefinition#getFlags(): instances should be automatically deleted when they are canceled.

```
public static final long DELETE_ON_EXPIRE = 16384L
```

Process#getFlags(): process should be automatically deleted when it expires.

ProcessDefinition#getFlags(): instances should be automatically deleted when they expire.

```
public static final long ATTR_ADVANCED_SEARCH = 268435456L
```

Attribute#getFlags(): hide the attribute in advanced search.

```
public static final long ATTR_DATABASE_INDEX_PENDING = 536870912L
```

Attribute#getFlags(): there is an index operation using this attribute currently running on the database.

```
public static final long ATTR_DATABASE_INDEX = 1073741824L
```

Attribute#getFlags(): this attribute is indexed to speed up search using it as criteria.

```
public static final long ATTR_CAPITAL_LETTERS = 2147483648L
```

Attribute#getFlags(): this attribute contains only capital letters

```
public static final long VIRTUAL = 36028797018963968L
```

ArchiveFolder#getFlags(): the folder is a virtual place-holder for it's corresponding active object.

```
public static final long NO_CHILDREN_ALIVE = 2147483648L
```

ArchiveFolder#getFlags(): the folder has only the children in the trash or is empty.

```
public static final long HAS_VIRTUAL = 72057594037927936L
```

Folder#getFlags(): the folder has a virtual place-holder corresponding in the archive.

```
public static final long IS_LADRAD = 64L
```

ModelOCR#getFlags(): the OCR model is result of a FlexiCapture import

```
public static final long USER_PLANNER = 64L
```

Planner#getFlags(): the planner is a user's own custom planner.

```
public static final long PROCESS_PLANNER = 128L
```

Planner#getFlags(): the planner is a process's own custom planner.

```
public static final long VERSION_PLANNER = 256L
```

Planner#getFlags(): the planner is a process version's own custom planner.

```
public static final long REORDER_LABEL_ASC = 0L
```

'Folder sort' operation flag. Means sort folders by label alphabetically ascending.

```
public static final long REORDER_LABEL_DESC = 1L
```

'Folder sort' operation flag. Means sort folders by label alphabetically descending.

```
public static final long EMAIL_NOTIFICATION = 1L
```

Comment#getFlags(): Notify author of reply to the comment.

```
public static final long CERT_ISSUED_BY = 1099511627776L
```

Certificate: Bit 40: Specifies the Certificate#getAuthority() property.

```
public static final long CERT_SERVER = 2199023255552L
```

Certificate: Bit 41: Specifies the Certificate#isOnServer() property.

```
public static final long CERT_FLAG_VALID_2 = 4398046511104L
```

Certificate: Bit 42: Specifies the Certificate#getValid2() property.

```
public static final long CERT_IDENTITY_PREFIX = 17592186044416L
```

Certificate: Bit 44: Specifies the 'prefix' db column of the identity field.

```
public static final long CERT_IDENTITY_N0 = 35184372088832L
```

Certificate: Bit 45: Specifies the 'numeric0' db column of the identity field.

```
public static final long CERT_IDENTITY = 52776558133248L
```

Certificate: `CERT_IDENTITY_PREFIX | CERT_IDENTITY_N0` : Specifies the Certificate#getIdentityId() property.

```
public static final long CERT_GEN_CACERT = 70368744177664L
```

Certificate: Bit 46: Specifies the Certificate#getGenCACert() property.

```
public static final long CERTIFICATE_ALL = 139637976740415L
```

Certificate: `ID | LABEL | OWNER_ID | FLAGS | DESCRIPTION | DATE_C | DATE_M | FILE_ID | DEPLOYED | CERT_ISSUED_BY | CERT_FLAG_VALID_2 | CERT_IDENTITY | PASSWORD | CERT_SERVER | CERT_GEN_CACERT`

```
public static final long IS_JOB_SCAN = 1L
```

IsJob#getFlags(): Scan IS job type.

```
public static final long IS_JOB_IMPORT = 2L
```

IsJob#getFlags(): Import IS job type.

```
public static final long IS_JOB_EMAIL = 4L
```

IsJob#getFlags(): Email IS job type.

```
public static final long IS_JOB_LAD_RAD = 256L
```

IsJob#getFlags(): IS Job with lad rad model.

```
public static final long IS_JOB_SIGN = 512L
```

IsJob#getFlags(): IS Job with sign.

```
public static final long IS_JOB_SIGN_MANDATORY = 1024L
```

IsJob#getFlags(): IS Job with mandatory sign.

```
public static final long SIMPLE_DEBUG = 8388608L
```

Process#getFlags(): process is running is simple debug mode. This means that all tasks come back to the user that started the process.

ProcessDefinition#getFlags(): instances will be run in simple debug mode.

```
public static final long ENABLE_DOCUMENT_ACCESS = 16777216L
```

Process#getFlags(): process allows external access from other instances to its documents. This means that documents are accessible through security profile rather than the restrictive derfault mode which allows only the manager and the current task executors to access the documents

ProcessDefinition#getFlags(): instances will allow access to documents based on security profile.

```
public static final long HAS_PLUGIN_OPTIONS = 33554432L
```

Process#getFlags(): specifies if a process has custom options that will be used in process review plugin

ProcessDefinition#getFlags(): instances will make use of this flag

```
public static final long NOTIFICATION_DEFINITION_ALL = 270582952073L
```

```
ID | FLAGS | DATE_C | DATE_M | NOTIF_TYPE | NOTIF_EVENT | FORM_ID |
NOTIF_MAIL_TEMPLATE_ID | NOTIF_RECIPIENT_ID | NOTIF_OBJECT_ID |
NOTIF_RECURSIVE
```

```
public static final long NOTIFICATION_DEFINITION_FULL = 8516920160393L
```

```
NOTIFICATION_ALL | NOTIF_TIMESTAMP | NOTIF_EXECUTOR_ID |
NOTIF_EXECUTOR_LABEL | NOTIF_EXECUTOR_REAL_NAME
```

```
public static final long NOTIFICATION_ALL = 52630529257609L
```

```
ID | FLAGS | DATE_C | DATE_M | NOTIF_EVENT | NOTIF_MAIL_TEMPLATE_ID |
NOTIF_RECIPIENT_ID | NOTIF_OBJECT_ID | NOTIF_OBJECT_LABEL | FORM_ID |
NOTIF_FORM_LABEL | NOTIF_TIMESTAMP | NOTIF_EXECUTOR_ID |
NOTIF_EXECUTOR_LABEL | NOTIF_EXECUTOR_REAL_NAME | NOTIF_PROCESSED
```

```
public static final long NOTIFICATION_FULL = 70222715302025L
```

```
NOTIFICATION_ALL | NOTIF_OBJECT_DATA
```

```
public static final long FAVORITE_ALL = 520205L
```

```
ID | DATE_C | DATE_M | FLAGS | OWNER_ID | FAVORITE_OBJECT_ALL |
FAVORITE_POSITION
```

```
public static final long VC_FLAG_START = 1L
```

ValidationCycle#getFlags(): specifies validationcycle has active start stage .

```
public static final long VC_FLAG_MODIF = 2L
```

ValidationCycle#getFlags(): specifies validationcycle has active modification stage .

```
public static final long VC_FLAG_AVIZ = 4L
```

ValidationCycle#getFlags(): specifies validationcycle has active avization stage .

```
public static final long VC_FLAG_VALID = 8L
```

ValidationCycle#getFlags(): specifies validationcycle has active validation stage .

```
public static final long VC_FLAG_PUBLIC = 16L
```

ValidationCycle#getFlags(): specifies validationcycle has active publication stage .

```
public static final long VC_FLAG_GED = 32L
```

ValidationCycle#getFlags(): specifies validationcycle has active ged (read) stage .

```
public static final long VC_FLAG_BROADCAST = 65536L
```

ValidationCycle#getFlags(): specifies validationcycle broadcast mode

```
public static final long PACKAGE_ALL = 4294979615L
```

`ID | LABEL | DESCRIPTION | DATE_C | DATE_M | OWNER_ID | FLAGS`

```
public static final long SECURITY_PROFILE_FOR_WORKSPACE = 1L
```

SecurityProfile#getFlags(): SecurityProfile for workspace (workspace securityprofile).

```
public static final long OPINION_SET_BY_SUPERVISOR = 1L
```

This flag indicates that the value of the 'opinion' type attribute has been set by a supervisor.

```
public static final long KEEP_VALUES_ON_RESTART = 8388608L
```

Task#getFlags(): keep values on restart option.

```
public static final long TYPE_LOCALIZED = 1L
```

This flag indicates that the given type is localized.

```
public static final long ENABLE_TRANSLATIONS = 1L
```

This flag indicates that the given object(mail template, document template, composition rules) is localized.

```
public static final long MAIL_TEMPLATE_TYPE_MASK = 14L
```

MailTemplate#getFlags(): 3-bit mask for the mail template type (bits 1 and 2 and 3)

```
public static final long MAIL_TEMPLATE_FOR_PROCESS = 2L
```

MailTemplate#getFlags(): Mail template type 2 (`1 << 1`), mail template for process

```
public static final long MAIL_TEMPLATE_FOR_VCYCLE = 4L
```

MailTemplate#getFlags(): Mail template type 4 (2 << 1), mail template for validation cycle

```
public static final long MAIL_TEMPLATE_FOR_GENERAL_PURPOSE = 8L
```

MailTemplate#getFlags(): Mail template type 8 (3 << 1), mail template for general purpose

Prefix

```
public class Prefix
```

The Prefix class provides support for object identification; it contains all prefixes used for object IDs.

Field summary :

Modifier and type	Field
public static final java.lang.String	FILE
public static final java.lang.String	USER
public static final java.lang.String	USER_DELEGATION
public static final java.lang.String	USER_PROFILE
public static final java.lang.String	GROUP
public static final java.lang.String	ROLE
public static final java.lang.String	MANAGER
public static final java.lang.String	GROUP_USER
public static final java.lang.String	USER_ROLE
public static final java.lang.String	GROUP_ROLE
public static final java.lang.String	MANAGER_USER
public static final java.lang.String	FORM_DISPLAY_PROFILE

public static final java.lang.String	DATABASE
public static final java.lang.String	GROUP_DATABASE
public static final java.lang.String	USER_DATABASE
public static final java.lang.String	ROLE_DATABASE
public static final java.lang.String	STORAGE_AREA
public static final java.lang.String	DATABASE_STORAGE_RULE
public static final java.lang.String	GROUP_PORTAL
public static final java.lang.String	USER_PORTAL
public static final java.lang.String	ROLE_PORTAL
public static final java.lang.String	PLANNER
public static final java.lang.String	PLANNER_DAY
public static final java.lang.String	RESOURCE
public static final java.lang.String	SCRIPT
public static final java.lang.String	SECURITY_PROFILE
public static final java.lang.String	MAIL_TEMPLATE
public static final java.lang.String	DOCUMENT_TEMPLATE
public static final java.lang.String	FOLDER_PROFILE
public static final java.lang.String	COMPOSITION_RULE
public static final	SEARCH_TEMPLATE

java.lang.String	
public static final java.lang.String	USER_SEARCHTEMPLATE
public static final java.lang.String	GROUP_SEARCHTEMPLATE
public static final java.lang.String	ROLE_SEARCHTEMPLATE
public static final java.lang.String	ASYNC_JOB
public static final java.lang.String	EXTENDED_PROPERTY
public static final java.lang.String	OBJECT_EXTENDED_PROPERTY
public static final java.lang.String	CERTIFICATE
public static final java.lang.String	TYPE
public static final java.lang.String	GRID
public static final java.lang.String	ATTRIBUTE
public static final java.lang.String	FORM
public static final java.lang.String	FORM_ATTRIBUTE
public static final java.lang.String	VIEW
public static final java.lang.String	VIEW_ELEMENT
public static final java.lang.String	GROUP_VIEW
public static final java.lang.String	USER_VIEW
public static final java.lang.String	ROLE_VIEW
public static final java.lang.String	PROCESS_DEF

public static final java.lang.String	SCRIPT2
public static final java.lang.String	SCRIPT2_VERSION
public static final java.lang.String	COMPOSITION_RULE_VERSION
public static final java.lang.String	FORM_VERSION
public static final java.lang.String	PROCESS_DEF_VERSION
public static final java.lang.String	IS_JOB_VERSION
public static final java.lang.String	MODEL_OCR_VERSION
public static final java.lang.String	VIEW_VERSION
public static final java.lang.String	VIEW_ELEMENT_VERSION
public static final java.lang.String	IS_JOB
public static final java.lang.String	MODEL_OCR
public static final java.lang.String	GROUP_ISJOB
public static final java.lang.String	ROLE_ISJOB
public static final java.lang.String	USER_ISJOB
public static final java.lang.String	CLASS_ELEMENT
public static final java.lang.String	PROCESS
public static final java.lang.String	TASK
public static final java.lang.String	FOLDER
public static final	FOLDER_VALIDATIONCYCLE

java.lang.String	
public static final java.lang.String	FOLDER_MYDOCS
public static final java.lang.String	FOLDER_WORKSPACE
public static final java.lang.String	FOLDER_IN_WORKSPACE
public static final java.lang.String	PIN_DOCF
public static final java.lang.String	PIN_DOCP
public static final java.lang.String	PIN_DOCW
public static final java.lang.String	FILE_DOCF
public static final java.lang.String	FILE_DOCP
public static final java.lang.String	FICHE_DOCF
public static final java.lang.String	FICHE_DOCD
public static final java.lang.String	FICHE_DOCV
public static final java.lang.String	FICHE_DOCW
public static final java.lang.String	FILE_DOCD
public static final java.lang.String	FILE_DOCV
public static final java.lang.String	FILE_DOCW
public static final java.lang.String	FICHE_DOCP
public static final java.lang.String	REF_DOCF
public static final java.lang.String	REF_DOCP

public static final java.lang.String	REF_FICHEF
public static final java.lang.String	REF_FICHEP
public static final java.lang.String	FOLDER_NOTE
public static final java.lang.String	PROCESS_NOTE
public static final java.lang.String	DOC_NOTE_BASE
public static final java.lang.String	FICHEF_NOTE
public static final java.lang.String	DOCP_NOTE
public static final java.lang.String	FICHEP_NOTE
public static final java.lang.String	DOCF_VERSION
public static final java.lang.String	DOCP_VERSION
public static final java.lang.String	TASK_DEF
public static final java.lang.String	DELEGATION
public static final java.lang.String	ARCHIVE_FOLDER
public static final java.lang.String	ARCHIVE_PIN
public static final java.lang.String	ARCHIVE_FILE
public static final java.lang.String	INTERVENTION_REPORT
public static final java.lang.String	SERVER_REPORT
public static final java.lang.String	FOLDER_COMMENT
public static final	FOLDER_COMMENT_ARCH

java.lang.String	
public static final java.lang.String	DOCF_COMMENT
public static final java.lang.String	DOCF_COMMENT_ARCH
public static final java.lang.String	PROCESS_COMMENT
public static final java.lang.String	DOCP_COMMENT
public static final java.lang.String	VALIDCYCLE_TEMPLATE
public static final java.lang.String	VALIDCYCLE
public static final java.lang.String	VALIDCYCLE_STAGE_TEMPLATE
public static final java.lang.String	VALIDCYCLE_STAGE
public static final java.lang.String	VALIDCYCLE_STAGE_DATA_TEMPLATE
public static final java.lang.String	VALIDCYCLE_STAGE_DATA
public static final java.lang.String	VALIDCYCLE_STAGE_TEMPLATE_NOTIFICATION
public static final java.lang.String	VALIDCYCLE_STAGE_NOTIFICATION
public static final java.lang.String	EXPIRABLE_DOWNLOAD_LINK
public static final java.lang.String	NOTIFICATION_DEFINITION
public static final java.lang.String	NOTIFICATION
public static final java.lang.String	FAVORITE
public static final java.lang.String	OBJECT_LINK
public static final java.lang.String	PACKAGE

public static final java.lang.String	JUKE

Field detail :

```
public static final String FILE = "FILE"
```

File id prefix.

Id structure is

FILEDDDDXXXXXXXXAAAAAAAABBBBBBBBCCCCCCCC

where

- DDDD is the hexadecimal representation of the database numeric id,
- XXXXXXXX is the hexadecimal representation of the storage area numeric id,
- AAAAAAAA the most significant id-generator counter hexadecimal value,
- BBBBBBBB the mid significant id-generator counter hexadecimal value,
- CCCCCCCC the least significant id-generator counter hexadecimal value.

```
public static final String USER = "USER"
```

User object prefix.

Id structure is

USER0000XXXXXXXX000000000000000000000000

where XXXXXXXX is the hexadecimal representation of the user numeric id.

```
public static final String USER_DELEGATION = "DLGT"
```

User object prefix.

Id structure is

DLGT0000XXXXXXXX000000000000000000000000

where XXXXXXXX is the hexadecimal representation of the user delegation numeric id.

```
public static final String USER_PROFILE = "USPR"
```

User profile object prefix.

Id structure is

USPR0000XXXXXXXX000000000000000000000000

where XXXXXXXX is the hexadecimal representation of the user profile numeric id.

```
public static final String GROUP = "GROU"
```

Group object prefix.

Id structure is

`GROU0000XXXXXXXX00000000000000000000000000`

where `XXXXXXXX` is the hexadecimal representation of the group numeric id.

```
public static final String ROLE = "ROLE"
```

Role object prefix.

Id structure is

`USER0000XXXXXXXX00000000000000000000000000`

where `XXXXXXXX` is the hexadecimal representation of the role numeric id.

```
public static final String MANAGER = "MGRR"
```

Manager object prefix.

Id structure is

`MGRR0000XXXXXXXX00000000000000000000000000`

where `XXXXXXXX` is the hexadecimal representation of the manager role numeric id.

```
public static final String GROUP_USER = "ASGU"
```

User-to-group assignment object prefix.

Id structure is

`ASGU0000XXXXXXXXYYYYYYYY0000000000000000`

where `XXXXXXXX` is the hexadecimal representation of the group numeric id and `YYYYYYYY` is the hexadecimal representation of the user numeric id.

```
public static final String USER_ROLE = "ASUR"
```

Role-to-user assignment object prefix.

Id structure is

`ASUR0000XXXXXXXXYYYYYYYY0000000000000000`

where `XXXXXXXX` is the hexadecimal representation of the user numeric id and `YYYYYYYY` is the hexadecimal representation of the role numeric id.

```
public static final String GROUP_ROLE = "ASGR"
```

Role-to-group assignment object prefix.

Id structure is

ASRU0000XXXXXXXXYYYYYYYY0000000000000000

where XXXXXXXX is the hexadecimal representation of the group numeric id and YYYYYYYY is the hexadecimal representation of the role numeric id.

```
public static final String MANAGER_USER = "ASMU"
```

User-to-manager role assignment object prefix.

Id structure is

ASMU0000XXXXXXXXYYYYYYYY0000000000000000

where XXXXXXXX is the hexadecimal representation of the manager role numeric id and YYYYYYYY is the hexadecimal representation of the user numeric id.

```
public static final String FORM_DISPLAY_PROFILE = "ASFP"
```

Form-to-display profile assignment object prefix.

Id structure is:

ASUDDDDDXXXXXXXXYYYYYYYY0000000000000000

,

where DDDD is the hexadecimal representation of the database numeric id, XXXXXXXX is the hexadecimal representation of the form id, YYYYYYYY is the hexadecimal representation of the display profile id

```
public static final String DATABASE = "DATA"
```

Database object prefix.

Id structure is

DATADDDD0000000000000000000000000000

where DDDD is the hexadecimal representation of the database numeric id.

```
public static final String GROUP_DATABASE = "ASGD"
```

Group-to-database assignment object prefix.

Id structure is

ASGDDDDDXXXXXXXX0000000000000000000000000

where XXXXXXXX is the hexadecimal representation of the group numeric id and DDDD is the hexadecimal representation of the database numeric id.

```
public static final String USER_DATABASE = "ASUD"
```

User-to-database assignment object prefix.

Id structure is

`ASUDDDDDXXXXXXXX00000000000000000000000000`

where `XXXXXXXX` is the hexadecimal representation of the user numeric id and `DDDD` is the hexadecimal representation of the database numeric id.

```
public static final String ROLE_DATABASE = "ASRD"
```

Role-to-database assignment object prefix.

Id structure is

`ASRDDDDDXXXXXXXX00000000000000000000000000`

where `XXXXXXXX` is the hexadecimal representation of the role numeric id and `DDDD` is the hexadecimal representation of the database numeric id.

```
public static final String STORAGE_AREA = "AREA"
```

Storage area object prefix.

Id structure is

`AREA0000XXXXXXXX00000000000000000000000000`

where `XXXXXXXX` is the hexadecimal representation of the storage area numeric id.

```
public static final String DATABASE_STORAGE_RULE = "RULE"
```

Storage rule object prefix.

Id structure is

`RULE0000XXXXXXXX00000000000000000000000000`

where `XXXXXXXX` is the hexadecimal representation of the storage rule numeric id.

```
public static final String GROUP_PORTAL = "ASGP"
```

Group-to-portal assignment object prefix.

Id structure is

`ASUP0000XXXXXXXXYYYYYYYY0000000000000000`

where `XXXXXXXX` is the hexadecimal representation of the group numeric id and `YYYYYYYY` is the hexadecimal representation of the portal numeric id.

```
public static final String USER_PORTAL = "ASUP"
```

User-to-portal assignment object prefix.

Id structure is

ASUP0000XXXXXXXXYYYYYYYY0000000000000000

where XXXXXXXX is the hexadecimal representation of the user numeric id and YYYYYYYY is the hexadecimal representation of the portal numeric id.

```
public static final String ROLE_PORTAL = "ASRP"
```

Role-to-portal assignment object prefix.

Id structure is

ASRP0000XXXXXXXXYYYYYYYY0000000000000000

where XXXXXXXX is the hexadecimal representation of the role numeric id and YYYYYYYY is the hexadecimal representation of the portal numeric id.

```
public static final String PLANNER = "PLAN"
```

Planner object prefix.

Id structure is

PLAN0000XXXXXXXX00000000000000000000000000

where XXXXXXXX is the hexadecimal representation of the planner numeric id.

```
public static final String PLANNER_DAY = "PLND"
```

Planner day object prefix.

Id structure is

PLND0000XXXXXXXXYYYYYYYY0000000000000000

where XXXXXXXX is the hexadecimal representation of the planner numeric id and YYYYYYYY is the 00YYYYMMDD representation of the day.

```
public static final String RESOURCE = "RSRC"
```

Resource object prefix.

Id structure is

RSRC0000XXXXXXXX00000000000000000000000000

where XXXXXXXX is the hexadecimal representation of the resource numeric id.

```
public static final String SCRIPT = "SCPT"
```

Script object prefix.

Id structure is

SCPT0000XXXXXXXX00000000000000000000000000

where XXXXXXXX is the hexadecimal representation of the script numeric id.

```
public static final String SECURITY_PROFILE = "SECP"
```

Security profile object prefix.

Id structure is

SECPDDDDXXXXXXXX0000000000000000000000000

where XXXXXXXX is the hexadecimal representation of the security profile numeric id and DDDD is the database numeric id.

```
public static final String MAIL_TEMPLATE = "MTPL"
```

Mail template object prefix.

Id structure is

MTPLDDDDXXXXXXXX0000000000000000000000000

where XXXXXXXX is the hexadecimal representation of the mail template numeric id and DDDD is the database numeric id.

```
public static final String DOCUMENT_TEMPLATE = "DTPL"
```

Document template object prefix.

Id structure is

DTPLDDDDXXXXXXXX0000000000000000000000000

where XXXXXXXX is the hexadecimal representation of the document template numeric id and DDDD is the database numeric id.

```
public static final String FOLDER_PROFILE = "FLPR"
```

Folder profile object prefix.

Id structure is

FLPRDDDDXXXXXXXX0000000000000000000000000

where XXXXXXXX is the hexadecimal representation of the folder profile numeric id and DDDD is the database numeric id.

```
public static final String COMPOSITION_RULE = "CPRU"
```

Composition rule object prefix.

Id structure is

CPRUDDDDXXXXXXXX0000000000000000000000000

where XXXXXXXX is the hexadecimal representation of the composition rule numeric id and DDDD is the database numeric id.

```
public static final String SEARCH_TEMPLATE = "SRCH"
```

Search template object prefix.

Id structure is

SRCHDDDDXXXXXXXX00000000000000000000000000

where XXXXXXXX is the hexadecimal representation of the search template numeric id and DDDD is the database numeric id.

```
public static final String USER_SEARCHTEMPLATE = "ASUT"
```

Searchtemplate-to-user assignment object prefix.

Id structure is

ASUT0000XXXXXXXXYYYYYYYY0000000000000000

where XXXXXXXX is the hexadecimal representation of the user numeric id and YYYYYYYY is the hexadecimal representation of the Searchtemplate numeric id.

```
public static final String GROUP_SEARCHTEMPLATE = "ASGT"
```

Searchtemplate-to-group assignment object prefix.

Id structure is

ASGT0000XXXXXXXXYYYYYYYY0000000000000000

where XXXXXXXX is the hexadecimal representation of the group numeric id and YYYYYYYY is the hexadecimal representation of the Searchtemplate numeric id.

```
public static final String ROLE_SEARCHTEMPLATE = "ASRT"
```

Searchtemplate-to-role assignment object prefix.

Id structure is

ASRT0000XXXXXXXXYYYYYYYY0000000000000000

where XXXXXXXX is the hexadecimal representation of the role numeric id and YYYYYYYY is the hexadecimal representation of the Searchtemplate numeric id.

```
public static final String ASYNC_JOB = "JASY"
```

Asynchronous job object prefix.

Id structure is

JASY0000XXXXXXXX00000000000000000000000000

where XXXXXXXX is the hexadecimal representation of the asynchronous job numeric id.

```
public static final String EXTENDED_PROPERTY = "EXPR"
```

Extended property object prefix.

Id structure is

EXPR0000XXXXXXXX000000000000000000000000

where XXXXXXXX is the hexadecimal representation of the extended property id.

```
public static final String OBJECT_EXTENDED_PROPERTY = "ASOP"
```

Object to extended property assignment prefix.

Id structure is

ASOP0000XXXXXXXXYYYYYYYY0000000000000000

where XXXXXXXX is the hexadecimal representation of extended property id and YYYYYYYY is the hexadecimal representation of the object numeric id.

```
public static final String CERTIFICATE = "CERT"
```

'Certificate' object prefix.

Id structure is

CERT0000 XXXXXXXX 00000000 00000000 00000000

where XXXXXXXX is the hexadecimal representation of certificate id

```
public static final String TYPE = "DFTY"
```

Type (list) object prefix.

Id structure is

DFTYDDDDXXXXXXXX000000000000000000000000

where DDDD is the database numeric id and XXXXXXXX is the hexadecimal representation of the type numeric id.

```
public static final String GRID = "GRID"
```

Grid object prefix.

Id structure is

GRIDDDDDXXXXXXXX000000000000000000000000

where DDDD is the database numeric id and XXXXXXXX is the hexadecimal representation of the type numeric id.

```
public static final String ATTRIBUTE = "DFAT"
```

Attribute object prefix.

Id structure is

`DFATDDDDXXXXXXXX00000000000000000000000000`

where `DDDD` is the database numeric id and `XXXXXXXX` is the hexadecimal representation of the attribute numeric id.

```
public static final String FORM = "DFFR"
```

Form object prefix.

Id structure is

`DFFRDDDDXXXXXXXX00000000000000000000000000`

where `DDDD` is the database numeric id and `XXXXXXXX` is the hexadecimal representation of the form numeric id.

```
public static final String FORM_ATTRIBUTE = "ASFA"
```

Attribute-to-form assignment object prefix.

Id structure is

`ASGVDDDDXXXXXXXXYYYYYYYY0000000000000000`

where `DDDD` is the hexadecimal representation of the database numeric id, `XXXXXXXX` is the hexadecimal representation of the form numeric id and `YYYYYYYY` is the hexadecimal representation of the attribute numeric id.

```
public static final String VIEW = "DFVW"
```

View object prefix.

Id structure is

`DFVWDDDDXXXXXXXX00000000000000000000000000`

where `DDDD` is the database numeric id and `XXXXXXXX` is the hexadecimal representation of the view numeric id.

```
public static final String VIEW_ELEMENT = "DFVE"
```

View element object prefix.

Id structure is

`DFVEDDDDXXXXXXXXYYYYYYYY0000000000000000`

where `DDDD` is the database numeric id, `XXXXXXXX` is the hexadecimal representation of the view

numeric id and YYYYYYYY is the hexadecimal representation of the view element numeric id

```
public static final String GROUP_VIEW = "ASGV"
```

Group-to-view assignment object prefix.

Id structure is

ASGVDDDDXXXXXXXXYYYYYYYY0000000000000000

where DDDD is the hexadecimal representation of the database numeric id, XXXXXXXX is the hexadecimal representation of the group numeric id and YYYYYYYY is the hexadecimal representation of the view numeric id.

```
public static final String USER_VIEW = "ASUV"
```

User-to-view assignment object prefix.

Id structure is

ASUVDDDDXXXXXXXXYYYYYYYY0000000000000000

where DDDD is the hexadecimal representation of the database numeric id, XXXXXXXX is the hexadecimal representation of the user numeric id and YYYYYYYY is the hexadecimal representation of the view numeric id.

```
public static final String ROLE_VIEW = "ASRV"
```

Role-to-view assignment object prefix.

Id structure is

ASRVDDDDXXXXXXXXYYYYYYYY0000000000000000

where DDDD is the hexadecimal representation of the database numeric id, XXXXXXXX is the hexadecimal representation of the role numeric id and YYYYYYYY is the hexadecimal representation of the view numeric id.

```
public static final String PROCESS_DEF = "DFPR"
```

Process definition object prefix.

Id structure is

DFPRDDDDXXXXXXXX000000000000000000000000

where DDDD is the database numeric id and XXXXXXXX is the hexadecimal representation of the process definition numeric id.

```
public static final String SCRIPT2 = "SCPD"
```

Design script object prefix.

Id structure is

SCPDDDDXXXXXXXX00000000000000000000000000

where DDDD is the hexadecimal representation of the database numeric id, XXXXXXXX is the hexadecimal representation of the script numeric id.

```
public static final String SCRIPT2_VERSION = "VDSC"
```

version of Design script object prefix.

Id structure is

VDSCDDDDXXXXXXXXYYYYYYYY0000000000000000

where DDDD is the hexadecimal representation of the database numeric id, XXXXXXXX is the hexadecimal representation of the script numeric id. YYYYYYYY is the hexadecimal representation of the script version numeric id.

```
public static final String COMPOSITION_RULE_VERSION = "VDCR"
```

version of Design composition rule object prefix.

Id structure is

VDCRDDDDXXXXXXXXYYYYYYYY0000000000000000

where DDDD is the hexadecimal representation of the database numeric id, XXXXXXXX is the hexadecimal representation of the composition rule numeric id. YYYYYYYY is the hexadecimal representation of the composition rule version numeric id.

```
public static final String FORM_VERSION = "VDFR"
```

version of Design form object prefix.

Id structure is

VDFRDDDDXXXXXXXXYYYYYYYY0000000000000000

where DDDD is the hexadecimal representation of the database numeric id, XXXXXXXX is the hexadecimal representation of the form numeric id. YYYYYYYY is the hexadecimal representation of the form version numeric id.

```
public static final String PROCESS_DEF_VERSION = "VDPR"
```

version of Design process definition object prefix.

Id structure is

VDPRDDDDXXXXXXXXYYYYYYYY0000000000000000

where DDDD is the hexadecimal representation of the database numeric id, XXXXXXXX is the hexadecimal representation of the process definition numeric id. YYYYYYYY is the hexadecimal

representation of the process definition version numeric id.

```
public static final String IS_JOB_VERSION = "VDIS"
```

version of Design isjob object prefix.

Id structure is

VDISDDDDXXXXXXXXYYYYYYYY0000000000000000

where DDDD is the hexadecimal representation of the database numeric id, XXXXXXXX is the hexadecimal representation of the isjob numeric id. YYYYYYYY is the hexadecimal representation of the isjob version numeric id.

```
public static final String MODEL_OCR_VERSION = "VDMO"
```

version of Design modelocr object prefix.

Id structure is

VDMODDDDXXXXXXXXYYYYYYYY0000000000000000

where DDDD is the hexadecimal representation of the database numeric id, XXXXXXXX is the hexadecimal representation of the modelocr numeric id. YYYYYYYY is the hexadecimal representation of the modelocr version numeric id.

```
public static final String VIEW_VERSION = "VDVW"
```

Version of View object prefix.

Id structure is

VDVWDDDDXXXXXXXXYYYYYYYY0000000000000000

where DDDD is the database numeric id , XXXXXXXX is the hexadecimal representation of the view numeric id. YYYYYYYY is the hexadecimal representation of the view version numeric id.

```
public static final String VIEW_ELEMENT_VERSION = "VDVE"
```

Version of View element object prefix.

Id structure is

VDVEDDDDXXXXXXXXYYYYYYYY0000000000000000

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the view version numeric id, YYYYYYYY is the hexadecimal representation of the view version element numeric id

```
public static final String IS_JOB = "ISJB"
```

IntelliScan job object prefix.

Id structure is

`ISJBDDDDXXXXXXXX00000000000000000000000`

where `DDDD` is the database numeric id and `XXXXXXXX` is the hexadecimal representation of the job numeric id.

```
public static final String MODEL_OCR = "MOCR"
```

ModelOCR object prefix.

Id structure is

`MOCRDDDDXXXXXXXX00000000000000000000000`

where `DDDD` is the database numeric id and `XXXXXXXX` is the hexadecimal representation of the model ocr numeric id.

```
public static final String GROUP_ISJOB = "ASGI"
```

Group-to-IsJob assignment object prefix.

Id structure is

`ASGIDDDDXXXXXXXXYYYYYYYY0000000000000000`

where `DDDD` is the hexadecimal representation of the database numeric id, `XXXXXXXX` is the hexadecimal representation of the group numeric id and `YYYYYYYY` is the hexadecimal representation of the IsJob numeric id.

```
public static final String ROLE_ISJOB = "ASRI"
```

Role-to-view assignment object prefix.

Id structure is

`ASRIDDDDXXXXXXXXYYYYYYYY0000000000000000`

where `DDDD` is the hexadecimal representation of the database numeric id, `XXXXXXXX` is the hexadecimal representation of the role numeric id and `YYYYYYYY` is the hexadecimal representation of the IsJob numeric id.

```
public static final String USER_ISJOB = "ASUI"
```

User-to-IsJob assignment object prefix.

Id structure is

`ASUIDDDDXXXXXXXXYYYYYYYY0000000000000000`

where `DDDD` is the hexadecimal representation of the database numeric id, `XXXXXXXX` is the hexadecimal representation of the user numeric id and `YYYYYYYY` is the hexadecimal representation of the IsJob numeric id.

```
public static final String CLASS_ELEMENT = "CELM"
```

Classification element object prefix.

Id structure is

`CELMDDDDXXXXXXXXYYYYYYYYZZZZZZZZ00000000`

where `DDDD` is the database numeric id, `XXXXXXXX` is the hexadecimal representation of the view numeric id, `YYYYYYYY` is the hexadecimal representation of the view element numeric id, `ZZZZZZZZ` is the hexadecimal representation of the class element numeric id.

```
public static final String PROCESS = "PROC"
```

Process object prefix.

Id structure is

`PROCDDDDXXXXXXXX00000000000000000000000000`

where `DDDD` is the database numeric id, `XXXXXXXX` is the hexadecimal representation of the process numeric id.

```
public static final String TASK = "TASK"
```

Task object prefix.

Id structure is

`TASKDDDDXXXXXXXXYYYYYYYY0000000000000000`

where `DDDD` is the database numeric id, `XXXXXXXX` is the hexadecimal representation of the process numeric id. `YYYYYYYY` is the hexadecimal representation of the task numeric id.

```
public static final String FOLDER = "FLDR"
```

Folder object prefix.

Id structure is

`FLDRDDDDXXXXXXXX00000000000000000000000000`

where `DDDD` is the database numeric id, `XXXXXXXX` is the hexadecimal representation of the folder numeric id. Numeric '0' means database root folder.

```
public static final String FOLDER_VALIDATIONCYCLE = "FLVC"
```

Validation Cycle (Virtual) Folder prefix.

Id structure is

`FLVCDDDDXXXXXXXXYYYYYYYYZZZZZZZZWWWWWWWW`

where

DDDD is the database numeric id,

XXXXXXXX is the hexadecimal representation of the folder .

YYYYYYYY is the hexadecimal representation of the document stage,

ZZZZZZZZ is the hexadecimal representation of the document stage status (normal , delayed, expired - see Constants),

WWWWWWWW is the hexadecimal representation of the actor (i as executor, i as manager ,i as owner of validation cycle - see Constants).

```
public static final String FOLDER_MYDOCS = "FLMY"
```

My documents (Virtual) Folder prefix.

Id structure is

FLMYDDDDXXXXXXXX00000000000000000000000000

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the folder numeric id. Numeric '0' means 'my documents' folder.

```
public static final String FOLDER_WORKSPACE = "FLWS"
```

Workspace (the Root) Folder prefix.

Id structure is

FLWSDDDDXXXXXXXX00000000000000000000000000

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the folder numeric id. Numeric '0' means 'workspace' folder.

```
public static final String FOLDER_IN_WORKSPACE = "FLWR"
```

Workspace (not Root) Folder prefix.

Id structure is

FLWRDDDDXXXXXXXX00000000000000000000000000

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the folder numeric id. Numeric '0' means 'workspace' folder.

```
public static final String PIN_DOCF = "PINF"
```

Pin contained in folder object prefix.

Id structure is

PINFDDDDXXXXXXXXYYYYYYYY0000000000000000

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the folder

numeric id, YYYYYYYY is the hexadecimal representation of the pin numeric id.

```
public static final String PIN_DOCP = "PINP"
```

Pin contained in process object prefix.

Id structure is

PINPDDDDXXXXXXXXYYYYYYYY0000000000000000

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the process numeric id, YYYYYYYY is the hexadecimal representation of the pin numeric id.

```
public static final String PIN_DOCW = "PINW"
```

Pin contained in a workspace

Id structure is

PINWDDDDXXXXXXXXYYYYYYYY0000000000000000

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the process numeric id, YYYYYYYY is the hexadecimal representation of the pin numeric id.

```
public static final String FILE_DOCF = "DOCF"
```

File document contained in folder object prefix.

Id structure is

DOCFDDDDXXXXXXXXYYYYYYYYZZZZZZZZ00000000

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the folder numeric id, YYYYYYYY is the hexadecimal representation of the pin numeric id, (numeric 0 means document is not in pin), ZZZZZZZZ is the hexadecimal representation of the document numeric id.

```
public static final String FILE_DOCP = "DOCP"
```

File document contained in process object prefix.

Id structure is

DOCPDDDDXXXXXXXXYYYYYYYYZZZZZZZZ00000000

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the process numeric id, YYYYYYYY is the hexadecimal representation of the pin numeric id, (numeric 0 means document is not in pin), ZZZZZZZZ is the hexadecimal representation of the document numeric id.

```
public static final String FICHE_DOCF = "FCHF"
```

File-less document (index fiche) contained in folder object prefix.

Id structure is

FCHFDDDDXXXXXXXXYYYYYYYYZZZZZZZZ00000000

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the folder numeric id, YYYYYYYY is the hexadecimal representation of the pin numeric id, (numeric 0 means document is not in pin), ZZZZZZZZ is the hexadecimal representation of the document numeric id.

```
public static final String FICHE_DOCD = "FCHD"
```

File-less document (index fiche) contained in distribution list (modified copy of original) /.

Id structure is

FCHDDDDDXXXXXXXXYYYYYYYYZZZZZZZZ00000000

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the folder numeric id, YYYYYYYY is the hexadecimal representation of the pin numeric id, (numeric 0 means document is not in pin), ZZZZZZZZ is the hexadecimal representation of the document numeric id.

```
public static final String FICHE_DOCV = "FCHV"
```

File-less document (index fiche) contained in Validation Cycle /.

Id structure is

FCHVDDDDXXXXXXXXYYYYYYYYZZZZZZZZ00000000

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the folder numeric id, YYYYYYYY is the hexadecimal representation of the pin numeric id, (numeric 0 means document is not in pin), ZZZZZZZZ is the hexadecimal representation of the document numeric id.

```
public static final String FICHE_DOCW = "FCHW"
```

File-less document (index fiche) document contained in a workspace /.

Id structure is

FCHWDDDDXXXXXXXXYYYYYYYYZZZZZZZZ00000000

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the folder numeric id, YYYYYYYY is the hexadecimal representation of the pin numeric id, (numeric 0 means document is not in pin), ZZZZZZZZ is the hexadecimal representation of the document numeric id.

```
public static final String FILE_DOCD = "DOCD"
```

File document contained in distribution list /.

Id structure is

DOCDDDDDXXXXXXXXYYYYYYYYZZZZZZZZ00000000

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the folder numeric id, YYYYYYYY is the hexadecimal representation of the pin numeric id, (numeric 0 means

document is not in pin), ZZZZZZZZ is the hexadecimal representation of the document numeric id.

```
public static final String FILE_DOCV = "DOCV"
```

File document contained in validation cycle /.

Id structure is

DOCVDDDDXXXXXXXXYYYYYYYYZZZZZZZZ00000000

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the folder numeric id, YYYYYYYY is the hexadecimal representation of the pin numeric id, (numeric 0 means document is not in pin), ZZZZZZZZ is the hexadecimal representation of the document numeric id.

```
public static final String FILE_DOCW = "DOCW"
```

File document contained in a workspace /.

Id structure is

DOCWDDDDXXXXXXXXYYYYYYYYZZZZZZZZ00000000

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the process numeric id, YYYYYYYY is the hexadecimal representation of the pin numeric id, (numeric 0 means document is not in pin), ZZZZZZZZ is the hexadecimal representation of the document numeric id.

```
public static final String FICHE_DOCP = "FCHP"
```

File document contained in workspace /.

Id structure is

DOCVDDDDXXXXXXXXYYYYYYYYZZZZZZZZ00000000

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the folder numeric id, YYYYYYYY is the hexadecimal representation of the pin numeric id, (numeric 0 means document is not in pin), ZZZZZZZZ is the hexadecimal representation of the document numeric id.

```
public static final String REF_DOCF = "RFDF"
```

Folder contained reference to a file document object prefix.

Id structure is

RFDFDDDDXXXXXXXXYYYYYYYYZZZZZZZZ00000000

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the folder numeric id, YYYYYYYY is the hexadecimal representation of the pin numeric id, (numeric 0 means document is not in pin), ZZZZZZZZ is the hexadecimal representation of the reference document numeric id.

```
public static final String REF_DOCP = "RFDP"
```

Process contained reference to a file document object prefix.

Id structure is

RFDPDDDDXXXXXXXXYYYYYYYYZZZZZZZZ00000000

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the process numeric id, YYYYYYYY is the hexadecimal representation of the pin numeric id, (numeric 0 means document is not in pin), ZZZZZZZZ is the hexadecimal representation of the reference document numeric id.

```
public static final String REF_FICHEF = "RFFF"
```

Folder contained reference to a file-less (index fiche) document object prefix.

Id structure is

RFFFDDDDXXXXXXXXYYYYYYYYZZZZZZZZ00000000

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the folder numeric id, YYYYYYYY is the hexadecimal representation of the pin numeric id, (numeric 0 means document is not in pin), ZZZZZZZZ is the hexadecimal representation of the reference document numeric id.

```
public static final String REF_FICHEP = "RFFP"
```

Process contained reference to a file-less (index fiche) document object prefix.

Id structure is

RFFPDDDDXXXXXXXXYYYYYYYYZZZZZZZZ00000000

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the process numeric id, YYYYYYYY is the hexadecimal representation of the pin numeric id, (numeric 0 means document is not in pin), ZZZZZZZZ is the hexadecimal representation of the reference document numeric id.

```
public static final String FOLDER_NOTE = "NTFL"
```

Folder note object prefix.

Id structure is

NTFLDDDDXXXXXXXX0000000000000000YYYYYYYY

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the folder numeric id, YYYYYYYY is the hexadecimal representation of the note numeric id.

```
public static final String PROCESS_NOTE = "NTPR"
```

Process note object prefix.

Id structure is

NTPRDDDDXXXXXXXX0000000000000000YYYYYYYY

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the process numeric id, YYYYYYYY is the hexadecimal representation of the note numeric id.

```
public static final String DOC_NOTE_BASE = "NTD"
```

Folder contained document note object prefix.

Id structure is

NTDFDDDDXXXXXXXXYYYYYYYYZZZZZZZZWWWWWWWW

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the folder numeric id, YYYYYYYY is the hexadecimal representation of the pin numeric id, (numeric 0 means document is not in pin), ZZZZZZZZ is the hexadecimal representation of the document numeric id, WWWWWWWW is the hexadecimal representation of the note numeric id.

```
public static final String FICHEF_NOTE = "NTFF"
```

Folder contained card (fiche) note object prefix.

Id structure is

NTDFDDDDXXXXXXXXYYYYYYYYZZZZZZZZWWWWWWWW

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the folder numeric id, YYYYYYYY is the hexadecimal representation of the pin numeric id, (numeric 0 means card (fiche) is not in pin), ZZZZZZZZ is the hexadecimal representation of the card numeric id, WWWWWWWW is the hexadecimal representation of the note numeric id.

```
public static final String DOCP_NOTE = "NTDP"
```

Process contained document note object prefix.

Id structure is

NTDFDDDDXXXXXXXXYYYYYYYYZZZZZZZZWWWWWWWW

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the process numeric id, YYYYYYYY is the hexadecimal representation of the pin numeric id, (numeric 0 means document is not in pin), ZZZZZZZZ is the hexadecimal representation of the document numeric id, WWWWWWWW is the hexadecimal representation of the note numeric id.

```
public static final String FICHEP_NOTE = "NTFP"
```

Process contained document note object prefix.

Id structure is

NTDFDDDDXXXXXXXXYYYYYYYYZZZZZZZZWWWWWWWW

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the process numeric id, YYYYYYYY is the hexadecimal representation of the pin numeric id, (numeric 0 means card (fiche) is not in pin), ZZZZZZZZ is the hexadecimal representation of the card (fiche) numeric id, WWWWWWWW is the hexadecimal representation of the note numeric id.

```
public static final String DOCF_VERSION = "VNDF"
```

Folder contained document version object prefix.

Id structure is

VNDFDDDDXXXXXXXXYYYYYYYYZZZZZZZZWWWWWWWW

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the folder numeric id, YYYYYYYY is the hexadecimal representation of the pin numeric id, (numeric 0 means document is not in pin), ZZZZZZZZ is the hexadecimal representation of the document numeric id, WWWWWWWW is the hexadecimal representation of the version numeric id.

```
public static final String DOCP_VERSION = "VNDP"
```

Process contained document version object prefix.

Id structure is

VNDFDDDDXXXXXXXXYYYYYYYYZZZZZZZZWWWWWWWW

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the process numeric id, YYYYYYYY is the hexadecimal representation of the pin numeric id, (numeric 0 means document is not in pin), ZZZZZZZZ is the hexadecimal representation of the document numeric id, WWWWWWWW is the hexadecimal representation of the version numeric id.

```
public static final String TASK_DEF = "TKDF"
```

Task definition object prefix.

Id structure is

TKDFDDDDXXXXXXXXYYYYYYYY0000000000000000

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the process definition numeric id and YYYYYYYY is the hexadecimal representation of the task definition numeric id.

```
public static final String DELEGATION = "DLGN"
```

User object prefix.

Id structure is

USER0000XXXXXXXX00000000000000000000000000

where XXXXXXXX is the hexadecimal representation of the user numeric id.

```
public static final String ARCHIVE_FOLDER = "ARFL"
```

Archive folder object prefix.

Id structure is

ARFLDDDDXXXXXXXX00000000000000000000000000

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the folder numeric id. Numeric '0' means database root folder.

```
public static final String ARCHIVE_PIN = "APIN"
```

Pin contained in folder object prefix.

Id structure is

APINDDDDXXXXXXXXYYYYYYYY0000000000000000

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the folder numeric id, YYYYYYYY is the hexadecimal representation of the pin numeric id.

```
public static final String ARCHIVE_FILE = "ADOC"
```

File document contained in folder object prefix.

Id structure is

ADOCDDDDXXXXXXXXYYYYYYYYZZZZZZZZ00000000

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the folder numeric id, YYYYYYYY is the hexadecimal representation of the pin numeric id, (numeric 0 means document is not in pin), ZZZZZZZZ is the hexadecimal representation of the document numeric id.

```
public static final String INTERVENTION_REPORT = "INRE"
```

Intervention report.

Id structure is

INRE0000XXXXXXXX00000000000000000000000000

where XXXXXXXX is the hexadecimal representation of the intervention report numeric id.

```
public static final String SERVER_REPORT = "SVRE"
```

Server report.

Id structure is

SVRE0000XXXXXXXX00000000000000000000000000

where XXXXXXXX is the hexadecimal representation of the server report numeric id.

```
public static final String FOLDER_COMMENT = "CMTF"
```

A folder comment.

CMTFDDDDXXXXXXXX0000000000000000CCCCCCCC

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the folder numeric id, CCCCCCCC is the hexadecimal representation of the comment numeric id.

```
public static final String FOLDER_COMMENT_ARCH = "ACMF"
```

An archived folder's comment.

ACMFDDDDXXXXXXXX0000000000000000CCCCCCCC

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the folder numeric id, CCCCCCCC is the hexadecimal representation of the comment numeric id.

```
public static final String DOCF_COMMENT = "CMDF"
```

A document-in-a-folder's comment.

CMDFDDDDXXXXXXXXYYYYYYYYZZZZZZZZCCCCCCCC

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the folder numeric id, YYYYYYYY is the hexadecimal representation of the pin numeric id, (numeric 0 means document is not in pin), ZZZZZZZZ is the hexadecimal representation of the document numeric id, CCCCCCCC is the hexadecimal representation of the comment numeric id.

```
public static final String DOCF_COMMENT_ARCH = "ACDF"
```

An archived document-in-a-folder's comment.

ACDFDDDDXXXXXXXXYYYYYYYYZZZZZZZZCCCCCCCC

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the folder numeric id, YYYYYYYY is the hexadecimal representation of the pin numeric id, (numeric 0 means document is not in pin), ZZZZZZZZ is the hexadecimal representation of the document numeric id, CCCCCCCC is the hexadecimal representation of the comment numeric id.

```
public static final String PROCESS_COMMENT = "CMTP"
```

A processes comment.

CMTPDDDDXXXXXXXX0000000000000000CCCCCCCC

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the process numeric id, CCCCCCCC is the hexadecimal representation of the comment numeric id.

```
public static final String DOCP_COMMENT = "CMDP"
```

A document-in-a-process's comment.

CMDPDDDDXXXXXXXXYYYYYYYYZZZZZZZZCCCCCCCC

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the process numeric id, YYYYYYYY is the hexadecimal representation of the pin numeric id, (numeric 0 means document is not in pin), ZZZZZZZZ is the hexadecimal representation of the document numeric id, CCCCCCCC is the hexadecimal representation of the comment numeric id.

```
public static final String VALIDCYCLE_TEMPLATE = "VCCT"
```

validation cycle template .

Id structure is

VCCTDDDDXXXXXXXX00000000000000000000000000

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the validation cycle template id.

```
public static final String VALIDCYCLE = "VCCR"
```

validation cycle (runtime).

Id structure is

VCCRDDDDXXXXXXXX00000000000000000000000000

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the validation cycle id.

```
public static final String VALIDCYCLE_STAGE_TEMPLATE = "VCST"
```

validation cycle template stage .

Id structure is

VCSTDDDDXXXXXXXXYYYYYYYY0000000000000000

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the validation cycle template id, YYYYYYYY is the hexadecimal representation of the validation cycle template stage id.

```
public static final String VALIDCYCLE_STAGE = "VCSR"
```

validation cycle stage .

Id structure is

VCSRDDDDXXXXXXXXYYYYYYYY0000000000000000

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the validation cycle id, YYYYYYYY is the hexadecimal representation of the validation cycle stage id

```
public static final String VALIDCYCLE_STAGE_DATA_TEMPLATE = "VCDT"
```

validation cycle template stage data.

Id structure is

VCDTDDDDXXXXXXXXYYYYYYYYZZZZZZZZCCCCCCCC

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the validation cycle template id, YYYYYYYY is the hexadecimal representation of the validation cycle template stage id, ZZZZZZZZ is the hexadecimal representation of the actor id, CCCCCCCC is the hexadecimal representation of the actor type.

```
public static final String VALIDCYCLE_STAGE_DATA = "VCDR"
```

validation cycle stage data.

Id structure is

VCDRDDDDXXXXXXXXYYYYYYYYZZZZZZZZCCCCCCCC

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the validation cycle template id, YYYYYYYY is the hexadecimal representation of the validation cycle template stage id, ZZZZZZZZ is the hexadecimal representation of the actor id, CCCCCCCC is the hexadecimal representation of the actor type.

```
public static final String
VALIDCYCLE_STAGE_TEMPLATE_NOTIFICATION = "VCNT"
```

validation cycle template stage notification.

Id structure is

VCNTDDDDXXXXXXXXYYYYYYYYZZZZZZZZCCCCCCCC

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the validation cycle template id, YYYYYYYY is the hexadecimal representation of the validation cycle template stage id, ZZZZZZZZ is the hexadecimal representation of the event, CCCCCCCC is the hexadecimal representation of the destination type .

```
public static final String VALIDCYCLE_STAGE_NOTIFICATION = "VCNR"
```

validation cycle stage notification.

Id structure is

VCNRDDDDXXXXXXXXYYYYYYYYZZZZZZZZCCCCCCCC

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the validation cycle id, YYYYYYYY is the hexadecimal representation of the validation cycle stage id, ZZZZZZZZ is the hexadecimal representation of the event, CCCCCCCC is the hexadecimal representation of the

destination type .

```
public static final String EXPIRABLE_DOWNLOAD_LINK = "EXDL"
```

Expirable download link.

Id structure is

`EXDLDDDDXXXXXXXX00000000000000000000000000`

where `DDDD` is the database numeric id, `XXXXXXXX` is the hexadecimal representation of the expirable download link id.

```
public static final String NOTIFICATION_DEFINITION = "NODF"
```

Notification definition

Id structure is

`NODFDDDDXXXXXXXX00000000000000000000000000`

where `DDDD` is the database numeric id, `XXXXXXXX` is the hexadecimal representation of the notification definition id.

```
public static final String NOTIFICATION = "NOTF"
```

Notification event

Id structure is

`NOTFDDDDXXXXXXXX00000000000000000000000000`

where `DDDD` is the database numeric id, `XXXXXXXX` is the hexadecimal representation of the notification event id.

```
public static final String FAVORITE = "FAVO"
```

Favorite

Id structure is

`FAVODDDDXXXXXXXX00000000000000000000000000`

where `DDDD` is the database numeric id, `XXXXXXXX` is the hexadecimal representation of the notification id.

```
public static final String OBJECT_LINK = "OLNK"
```

Object link

Id structure is

`OLNKDDDDXXXXXXXX00000000000000000000000000`

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the link id.

```
public static final String PACKAGE = "PCKG"
```

Package link

Id structure is

PCKGDDDDXXXXXXXX00000000000000000000000000

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the package id.

```
public static final String JUKE = "JUKE"
```

Juke link

Id structure is

JUKEDDDDXXXXXXXX00000000000000000000000000

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the jukeid id.

SIATEL_0

```
public abstract class Siatel_0
```

Inheritance root abstrract class for all Identified implementations.

All Superinterfaces :

Serializable , Cloneable , Replicable , Identified

See also :

Identified , Prefix , IdFormat

Constructor summary :

Constructor
Siatel_0 ()
Siatel_0 (com.siatel.domain.Siatel_0 source)
Siatel_0 (java.lang.String id)

Constructor detail :

```
public Siatel_0 ()
```

Default constructor.

```
public Siatel_0 (Siatel_0 source)
```

Copy constructor.

Parameters :

`source` - the copy source

```
public Siatel_0 (String id)
```

Id initializer constructor.

Parameters :

`id` - the object id

SIATEL_1

```
public abstract class Siatel_1
```

All Superinterfaces :

Labeled , Owned , Flagged , Dated

Author :

stefanu

Constructor summary :

Constructor
`Siatel_1 ()`
`Siatel_1 (java.lang.String id)`
`Siatel_1 (java.lang.String id, java.lang.String label, java.lang.String ownerId, long flags, java.util.Date dateC, java.util.Date dateM)`

Constructor detail :

```
public Siatel_1 ()
```

```
public Siatel_1 (String id)
```

```
public Siatel_1 (String id, String label, String ownerId, long flags,
Date dateC, Date dateM)
```

SIATEL_1_1

```
public abstract class Siatel_1_1
```

All Superinterfaces :

Described

Author :

stefanu

Constructor summary :

Constructor
Siatel_1_1 ()
Siatel_1_1 (java.lang.String id)
Siatel_1_1 (java.lang.String id, java.lang.String label, java.lang.String ownerId, long flags, java.util.Date dateC, java.util.Date dateM, java.lang.String description)

Constructor detail :

```
public Siatel_1_1 ()
```

```
public Siatel_1_1 (String id)
```

```
public Siatel_1_1 (String id, String label, String ownerId, long flags, Date dateC, Date dateM, String description)
```

SIATEL_1_2

```
public abstract class Siatel_1_2
```

All Superinterfaces :

Deployable

Author :

stefanu

Constructor summary :

Constructor
Siatel_1_2 ()
Siatel_1_2 (java.lang.String id)
Siatel_1_2 (java.lang.String id, java.lang.String label, java.lang.String ownerId, long flags, java.util.Date dateC, java.util.Date dateM, java.lang.String description, boolean deployed)

Constructor detail :

```
public Siatel_1_2 ()
```

```
public Siatel_1_2 (String id)
```

```
public Siatel_1_2 (String id, String label, String ownerId, long flags,
Date dateC, Date dateM, String description, boolean deployed)
```

SIATEL_1_3

```
public abstract class Siatel_1_3
```

All Superinterfaces :

Realized

Author :

stefanu

Constructor summary :

Constructor
Siatel_1_3 ()
Siatel_1_3 (java.lang.String id)
Siatel_1_3 (java.lang.String id, java.lang.String label, java.lang.String ownerId, long flags, java.util.Date dateC, java.util.Date dateM, java.lang.String description, boolean deployed, java.lang.String fileId)

Constructor detail :

```
public Siatel_1_3 ()
```

```
public Siatel_1_3 (String id)
```

```
public Siatel_1_3 (String id, String label, String ownerId, long flags,
Date dateC, Date dateM, String description, boolean deployed, String
fileId)
```

SIATEL_1_4

```
public abstract class Siatel_1_4
```

All Superinterfaces :

DeployableEx

Since :

30/11/2007

Author :

Victor Citiriga

Constructor summary :

Constructor
Siatel_1_4 ()
Siatel_1_4 (java.lang.String id)
Siatel_1_4 (java.lang.String id, java.lang.String label, java.lang.String ownerId, long flags, java.util.Date dateC, java.util.Date dateM, java.lang.String description, boolean deployed, java.lang.String fileId, java.lang.String deploymentId, long deploymentTimestamp)

Constructor detail :

```
public Siatel_1_4 ()
```

```
public Siatel_1_4 (String id)
```

```
public Siatel_1_4 (String id, String label, String ownerId, long flags, Date dateC, Date dateM, String description, boolean deployed, String fileId, String deploymentId, long deploymentTimestamp)
```

SIATEL_2

```
public abstract class Siatel_2
```

New in 7.8: added support for UUIDEnabled. The uuid is not (de)serialized in this class, but by descending classes, since it was later added and descendants' token indexes have evolved differently

All Superinterfaces :

DatedEx , Viewable , Timed , UUIDEnabled

Author :

victorc

null :

7.8

Constructor summary :

Constructor
Siatel_2 ()
Siatel_2 (java.lang.String id)

Siatel_2 (java.lang.String id, java.lang.String label, java.lang.String ownerId, long flags, java.util.Date dateC, java.util.Date dateM, java.util.Date dateA, java.lang.String formId, java.util.List attributes, int icon, java.util.Date dateS, java.util.Date dateE)

Constructor detail :

```
public Siatel_2 ()
```

```
public Siatel_2 (String id)
```

```
public Siatel_2 (String id, String label, String ownerId, long flags, Date dateC, Date dateM, Date dateA, String formId, List attributes, int icon, Date dateS, Date dateE)
```

SIATEL_3

```
public abstract class Siatel_3
```

All Superinterfaces :

Lifecycled , Securable , Deletable

Author :

stefanu

Constructor summary :

Constructor
Siatel_3 ()
Siatel_3 (java.lang.String id)
Siatel_3 (java.lang.String id, java.lang.String label, java.lang.String ownerId, long flags, java.util.Date dateC, java.util.Date dateM, java.util.Date dateA, java.lang.String formId, java.util.List attributes, int icon, java.lang.String stateId, java.util.Date dateS, java.util.Date dateE, boolean deleted, java.util.Date dateDel, int deleteBy)

Constructor detail :

```
public Siatel_3 ()
```

```
public Siatel_3 (String id)
```

```
public Siatel_3 (String id, String label, String ownerId, long flags, Date dateC, Date dateM, Date dateA, String formId, List attributes, int
```

```
icon, String stateId, Date dateS, Date dateE, boolean deleted, Date
dateDel, int deleteBy)
```

TAG

```
public class Tag
```

Object tags.

Bits 0-63 match to different object tags

Author :

stefanu

Field summary :

Modifier and type	Field
public static final long	TAG_TEXT_1
public static final long	TAG_TEXT_2
public static final long	TAG_TEXT_3
public static final long	TAG_TEXT_4
public static final long	TAG_TEXT_5
public static final long	TAG_BACKGROUND_1
public static final long	TAG_BACKGROUND_2
public static final long	TAG_BACKGROUND_3
public static final long	TAG_BACKGROUND_4
public static final long	TAG_BACKGROUND_5

Field detail :

```
public static final long TAG_TEXT_1 = 1L
```

Bit 1. Tag #1.

```
public static final long TAG_TEXT_2 = 2L
```

Bit 2. Tag #2.

```
public static final long TAG_TEXT_3 = 4L
```

Bit 3. Tag #3.

```
public static final long TAG_TEXT_4 = 8L
```

Bit 4. Tag #1.

```
public static final long TAG_TEXT_5 = 16L
```

Bit 5. Tag #2.

```
public static final long TAG_BACKGROUND_1 = 4294967296L
```

Bit 32. Tag #1.

```
public static final long TAG_BACKGROUND_2 = 8589934592L
```

Bit 33. Tag #2.

```
public static final long TAG_BACKGROUND_3 = 17179869184L
```

Bit 34. Tag #1.

```
public static final long TAG_BACKGROUND_4 = 34359738368L
```

Bit 35. Tag #2.

```
public static final long TAG_BACKGROUND_5 = 68719476736L
```

Bit 36. Tag #2.

UniqueCriterion

```
public class UniqueCriterion
```

This class defines a unqiue criterion used by view elements design and runtime.

This class consists of:

1. the **id of the attribute** on which this criterion is evaluated.
2. a **modifier** to apply the attribute values when evaluating.
3. the **value** to which this criterion evaluates at view runtime. This member is ignored when managing view elements (view design).

All Superinterfaces :

Cloneable

Since :

12/07/2006

Author :

vic

See also :

ViewElement , View

Constructor summary :

Constructor
`UniqueCriterion ()`
`UniqueCriterion (java.lang.String attrId, int attrModifier)`
`UniqueCriterion (java.lang.String attrId, int attrModifier, java.lang.Object value)`
`UniqueCriterion (java.lang.String attrId, int attrModifier, java.lang.Object value, boolean ascending)`

Constructor detail :

```
public UniqueCriterion ()
```

No arguments constructor.

```
public UniqueCriterion (String attrId, int attrModifier)
```

Constructor with design time initialization parameters.

Parameters :

`attrId` - the attribute id

`attrModifier` - the attribute modifier

```
public UniqueCriterion (String attrId, int attrModifier, Object value)
```

Constructor with runtime initialization parameters.

Parameters :

`attrId` - the attribute id

`attrModifier` - the attribute modifier

`value` - the criterion evaluation

```
public UniqueCriterion (String attrId, int attrModifier, Object value,
boolean ascending)
```

Constructor with runtime initialization parameters.

Parameters :

`attrId` - the attribute id

`attrModifier` - the attribute modifier

`value` - the criterion evaluation

`ascending` - the sort order

ValueToken

```
public class ValueToken
```

String tokens used for various purposes.

Since :

12/07/2007

Author :

vic

Constructor summary :

Constructor
`ValueToken ()`

Constructor detail :

```
public ValueToken ()
```

Package com.siatel.framework.service.admin

This package groups administration related service definitions and utilities.

Classes

AdministrationLibrary

```
public class AdministrationLibrary
```

This class defines the id's for the administration service commands.

Since :

01/03/2007

Author :

Victor Citiriga

Field summary :

Modifier and type	Field
public static final int	CMD_CREATE_DATABASE
public static final int	CMD_READ_DATABASE
public static final int	CMD_ENUMERATE_DATABASES
public static final int	CMD_CREATE_GROUP
public static final int	CMD_READ_GROUP
public static final int	CMD_UPDATE_GROUP
public static final int	CMD_DELETE_GROUP
public static final int	CMD_ENUMERATE_GROUPS
public static final int	CMD_CREATE_USER
public static final int	CMD_READ_USER
public static final int	CMD_UPDATE_USER
public static final int	CMD_DELETE_USER
public static final int	CMD_ENUMERATE_USERS
public static final int	CMD_UPDATE_USER_PASS
public static final	CMD_CREATE_ROLE

int	
public static final int	CMD_READ_ROLE
public static final int	CMD_UPDATE_ROLE
public static final int	CMD_DELETE_ROLE
public static final int	CMD_ENUMERATE_ROLES
public static final int	CMD_CREATE_GROUP_USER
public static final int	CMD_DELETE_GROUP_USER
public static final int	CMD_ENUMERATE_GROUP_USER
public static final int	CMD_CREATE_USER_ROLE
public static final int	CMD_DELETE_USER_ROLE
public static final int	CMD_ENUMERATE_USER_ROLE
public static final int	CMD_CREATE_GROUP_ROLE
public static final int	CMD_DELETE_GROUP_ROLE
public static final int	CMD_ENUMERATE_GROUP_ROLE
public static final int	CMD_READ_STORAGE
public static final int	CMD_ENUMERATE_STORAGES
public static final int	CMD_READ_PORTAL
public static final int	CMD_ENUMERATE_PORTALS
public static final int	CMD_DELEGATE_OWNERSHIP

public static final int	CMD_READ_RESOURCE
public static final int	CMD_ENUMERATE_RESOURCES

Constructor summary :

Constructor
AdministrationLibrary ()

Field detail :

```
public static final int CMD_CREATE_DATABASE = 33556225
```

Create a database.

DTO structure on input

Key	Type	Significance
object	Database	The database object to create.

DTO structure on output

Key	Type	Significance
object	Database	The same object supplied on input but with id, dateC and other initialization-time relevant fields set.

Validations to perform

- value of label field of input object is not-null and unique in value throughout the domain of the server. Failure will result in a InvalidInputBusinessException

Specific responsabilities

- persist and initialize the administration Database object
- create the corresponding production database schema
- create the fulltext database schema, if flags field of supplied object has Flag.HAS_FULLTEXT bit set
- create the default form for the created database

```
public static final int CMD_READ_DATABASE = 33556226
```

Read a database administration object.

DTO structure on input

Key	Type	Significance
`object`	`Database`	The database object to initialize with persisted data.
`flags`	`long`	Flags specifying what fields of the database object to initialize.

DTO structure on output

Key	Type	Significance
`object`	`Database`	The same object supplied on input but with the fields specified by `flags` parameter set.

Validations to perform

- value of either `id` or `label` fields of input `object` are not-null. Failure will result in a `InvalidInputBusinessException`
- object specified by `id` or `label` exists in the database. Failure will result in a `NotFoundPersistenceException`

Specific responsabilities

- invoke persistence to initialize the `object`

```
public static final int CMD_ENUMERATE_DATABASES = 33556229
```

Enumerate/list databases.

DTO structure on input

Key	Type	Significance
`flags`	`long`	the fields of the Database object to initialize with server data. A combination of Flag defined constants, relevant for the Database class objects (see Flag#DATABASE_ALL)
`offset`	`Integer`	offset from which to start enumeration
`count`	`Integer`	maximum number of objects to return.

`byConnection`	`Boolean`	Optional.If specified and set to `true` then only databases in which the connected user can access content are listed (the ones the user can access when he logs in to the web application). Otherwise, all existing databases are listed

DTO structure on output

Key	Type	Significance
`list`	`List<Database>`	the list of databases

Errors

None

Notes

None

```
public static final int CMD_CREATE_GROUP = 33554689
```

Create a group.

DTO structure on input

Key	Type	Significance
`object`	Group	the data of the group to create.Must have the `label` property set.

DTO structure on output

Key	Type	Significance
`object`	Group	the same object as on input but with creation (essential for further use would be the `id`) data initialized

Errors

- InvalidInputBusinessException - invalid `label` (null or empty)
- AlreadyExistsPersistenceException - an object with the same name already exists
- DeniedSecurityException - security context does not allow this operation for connected user

Notes

- group names are case insensitive (e.g. creating the group "dummy" when a group "Dummy"

exists will generate an error)
- this operation is internally audited

```
public static final int CMD_READ_GROUP = 33554690
```

Read a group's data.

DTO structure on input

Key	Type	Significance
`object`	Group	the group object to initialize with persisted data.Must have either `id` or `label` properties set
`flags`	`long`	Flags specifying what fields of the group object to initialize (see Flag#GROUP_ALL).

DTO structure on output

Key	Type	Significance
`object`	`Group`	the same object supplied on input but with the fields specified by `flags` parameter set.

Errors

- InvalidInputBusinessException - neither `id` nor `label` are specified in input `object`
- NotFoundPersistenceException - object with specified `id` or `label` not found
- DeniedSecurityException - security context does not allow this operation for connected user

Notes

- group names are case insensitive (e.g. creating the group "dummy" when a group "Dummy" exists will generate an error)
- this operation is internally audited

```
public static final int CMD_UPDATE_GROUP = 33554691
```

Update a group's data.

DTO structure on input

Key	Type	Significance
`object`	Group	the group object to initialize with persisted data.Must have either `id` or `label` properties set
`flags`	`long`	flags specifying what fields of the group object to update (see Flag#GROUP_ALL).

DTO structure on output

Key	Type	Significance
`object`	`Group`	the same object supplied on input

Errors

- InvalidInputBusinessException - invalid `id` or `label` is `null` and `flags` contains Flag#LABEL
- NotFoundPersistenceException - object with specified `id` not found
- AlreadyExistsPersistenceException - a different object with specified `label` already exists when updating the group name
- DeniedSecurityException - security context does not allow this operation for connected user

Notes

- group names are case insensitive (e.g. creating the group "dummy" when a group "Dummy" exists will generate an error)
- this operation is internally audited

```
public static final int CMD_DELETE_GROUP = 33554692
```

Delete a group.

DTO structure on input

Key	Type	Significance
`objectId`	`String`	the id of the group object to delete

DTO structure on output

empty, `null` will be returned

Errors

- InvalidInputBusinessException - invalid `objectId`
- DeniedSecurityException - security context does not allow this operation for connected user

Notes

- this operation is internally audited

```
public static final int CMD_ENUMERATE_GROUPS = 33554693
```

Enumerate/list/search groups.

DTO structure on input

Key	Type	Significance
`criteriaIds`	`List<String>`	a list of criteria used to enumerate the groups

`flags`	`long`	flags specifying what fields of the group objects returned to initialize (see Flag#GROUP_ALL)
`offset`	`Integer`	offset from which to start enumeration
`count`	`Integer`	maximum number of objects to return.

DTO structure on output

Key	Type	Significance
`list`	`List<Group>`	the list of groups

Errors

- InvalidInputBusinessException - invalid `criteriaIds` (see Notes for supported criteria combinations)
- DeniedSecurityException - security context does not allow this operation for connected user

Notes

- group names are case insensitive (e.g. creating the group "dummy" when a group "Dummy" exists will generate an error)
- The following table lists the supported criteria combinations

Content of `criteriaIds`	Significance
`[]` - empty	all groups are enumerated
`["LABEL_SW", <expr>]`	groups with `label` starting with <expr>
`["LABEL", <expr>]`	groups with `label` containg <expr>.A minimum of 2 characters requirement for <expr> in this case
`["DESCRIPTION", <expr>]`	groups with `description` containg <expr>.A minimum of 2 characters requirement for <expr> in this case
`["REALNAME", <expr>]`	groups with `realName` containg <expr>.A minimum of 2 characters requirement for <expr> in this case

- in the table above, `"LABEL"`, `"DESCRIPTION"` and `"REALNAME"` can be combined in the same criterion.E.g. specifying `["LABEL REALNAME", <expr>]` will look for <expr> in both `label` and `realName` properties of group objects.`"LABEL_SW"` cannot be part of these combinations
- this operation is internally audited

```
public static final int CMD_CREATE_USER = 33554945
```

Create a user (account).

DTO structure on input

Key	Type	Significance
`object`	User	the data of the user to create.Must have the `label` property set.

DTO structure on output

Key	Type	Significance
`object`	User	the same object as on input but with creation (essential for further use would be the `id`) data initialized

Errors

- InvalidInputBusinessException - invalid `label` (null or empty)
- AlreadyExistsPersistenceException - an object with the same name already exists
- DeniedSecurityException - security context does not allow this operation for connected user

Notes

- user names are case insensitive (e.g. creating the user "dummy" when a user "Dummy" exists will generate an error)
- this operation is internally audited

```
public static final int CMD_READ_USER = 33554946
```

Read a user's data.

DTO structure on input

Key	Type	Significance
`object`	User	the user object to initialize with persisted data.Must have either `id` or `label` properties set
`flags`	`long`	Flags specifying what fields of the user object to initialize (see Flag#USER_ALL).

DTO structure on output

Key	Type	Significance
`object`	User	the same object supplied on input but with the fields specified by `flags` parameter set.

Errors

- InvalidInputBusinessException - neither `id` nor `label` are specified in input `object`
- NotFoundPersistenceException - object with specified `id` or `label` not found
- DeniedSecurityException - security context does not allow this operation for connected user

Notes

- user names are case insensitive (e.g. creating the user "dummy" when a user "Dummy" exists will generate an error)
- trying to get the user password using this command will disappoint you
- this operation is internally audited

```
public static final int CMD_UPDATE_USER = 33554947
```

Update a user's data.

DTO structure on input

Key	Type	Significance
`object`	User	the user object to initialize with persisted data.Must have either `id` or `label` properties set
`flags`	`long`	flags specifying what fields of the user object to update (see Flag#USER_ALL).

DTO structure on output

Key	Type	Significance
`object`	`User`	the same object supplied on input.

Errors

- InvalidInputBusinessException - invalid `id` or `label` is `null` and `flags` contains Flag#LABEL
- NotFoundPersistenceException - object with specified `id` not found
- AlreadyExistsPersistenceException - a different object with specified `label` already exists when updating the user name
- DeniedSecurityException - security context does not allow this operation for connected user

Notes

- user names are case insensitive (e.g. creating the user "dummy" when a user "Dummy" exists will generate an error)
- this command will not change a user's password.Use #CMD_UPDATE_USER_PASS to achieve this
- this operation is internally audited

```
public static final int CMD_DELETE_USER = 33554948
```

Delete a user (account).

DTO structure on input

Key	Type	Significance
`objectId`	`String`	the id of the user object to delete

DTO structure on output

empty, `null` will be returned

Errors

- InvalidInputBusinessException - invalid `objectId`
- DeniedSecurityException - security context does not allow this operation for connected user

Notes

- this operation is internally audited

```
public static final int CMD_ENUMERATE_USERS = 33554949
```

Enumerate/list/search users.

DTO structure on input

Key	Type	Significance
`criteriaIds`	`List<String>`	a list of criteria used to enumerate the users
`flags`	`long`	flags specifying what fields of the user objects returned to initialize (see Flag#USER_ALL)
`offset`	`Integer`	offset from which to start enumeration
`count`	`Integer`	maximum number of objects to return.

DTO structure on output

Key	Type	Significance
`list`	`List<User>`	the list of users

Errors

- InvalidInputBusinessException - invalid `criteriaIds` (see Notes for supported criteria combinations)
- DeniedSecurityException - security context does not allow this operation for connected user

Notes

- user names are case insensitive (e.g. creating the user "dummy" when a user "Dummy" exists will generate an error)
- trying to get the users' passwords using this command will disappoint you
- The following table lists the supported criteria combinations

Content of `criteriaIds`	Significance
`[]` - empty	all users are enumerated
`["LABEL_SW", <expr>]`	users with `label` starting with <expr>
`["LABEL", <expr>]`	users with `label` containg <expr>.A minimum of 2 characters requirement for <expr> in this case
`["DESCRIPTION", <expr>]`	users with `description` containg <expr>.A minimum of 2 characters requirement for <expr> in this case
`["REALNAME", <expr>]`	users with `realName` containg <expr>.A minimum of 2 characters requirement for <expr> in this case

- in the table above, `"LABEL"`, `"DESCRIPTION"` and `"REALNAME"` can be combined in the same criterion.E.g. specifying `["LABEL REALNAME", <expr>]` will look for <expr> in both `label` and `realName` properties of user objects.`"LABEL_SW"` cannot be part of these combinations
- this operation is internally audited

```
public static final int CMD_UPDATE_USER_PASS = 33555072
```

Change a user's password.

DTO structure on input

Key	Type	Significance
`userIds`	`String`	the id of the user whose password to change
`passwrod`	`String`	the new password
`expired`	`Boolean`	Optional.`true` to force intercative password change on next user login

DTO structure on output

Key	Type	Significance
`object`	`User`	the user object containing the updated timestamp

Errors

- InvalidInputBusinessException - invalid `userId`
- PasswordValidationBusinessException - specified password does not conform to password policy requirements
- DeniedSecurityException - security context does not allow this operation for connected user

Notes

- this operation is internally audited

```
public static final int CMD_CREATE_ROLE = 33555201
```

Create a role.

DTO structure on input

Key	Type	Significance
`object`	Role	the data of the role to create.Must have the `label` property set.

DTO structure on output

Key	Type	Significance
`object`	Role	the same object as on input but with creation (essential for further use would be the `id`) data initialized

Errors

- InvalidInputBusinessException - invalid `label` (null or empty)
- AlreadyExistsPersistenceException - an object with the same name already exists
- DeniedSecurityException - security context does not allow this operation for connected user

Notes

- role names are case insensitive (e.g. creating the role "dummy" when a role "Dummy" exists will generate an error)
- this operation is internally audited

```
public static final int CMD_READ_ROLE = 33555202
```

Read a role's data.

DTO structure on input

Key	Type	Significance

`object`	Role	the role object to initialize with persisted data.Must have either `id` or `label` properties set
`flags`	`long`	Flags specifying what fields of the role object to initialize (see Flag#ROLE_ALL).

DTO structure on output

Key	Type	Significance
`object`	Role	the same object supplied on input but with the fields specified by `flags` parameter set.

Errors

- InvalidInputBusinessException - neither `id` nor `label` are specified in input `object`
- NotFoundPersistenceException - object with specified `id` or `label` not found
- DeniedSecurityException - security context does not allow this operation for connected user

Notes

- role names are case insensitive (e.g. creating the role "dummy" when a role "Dummy" exists will generate an error)
- this operation is internally audited

```
public static final int CMD_UPDATE_ROLE = 33555203
```

Update a role's data.

DTO structure on input

Key	Type	Significance
`object`	Role	the role object to initialize with persisted data.Must have either `id` or `label` properties set
`flags`	`long`	flags specifying what fields of the role object to update (see Flag#ROLE_ALL).

DTO structure on output

Key	Type	Significance
`object`	Role	the same object supplied on input

Errors

- InvalidInputBusinessException - invalid `id` or `label` is `null` and `flags` contains Flag#LABEL
- NotFoundPersistenceException - object with specified `id` not found
- AlreadyExistsPersistenceException - a different object with specified `label` already exists

when updating the role name
- DeniedSecurityException - security context does not allow this operation for connected user

Notes

- role names are case insensitive (e.g. creating the role "dummy" when a role "Dummy" exists will generate an error)
- this operation is internally audited

```
public static final int CMD_DELETE_ROLE = 33555204
```

Delete a role.

DTO structure on input

Key	Type	Significance
`objectId`	`String`	the id of the role object to delete

DTO structure on output

empty, `null` will be returned

Errors

- InvalidInputBusinessException - invalid `objectId`
- DeniedSecurityException - security context does not allow this operation for connected user

Notes

- this operation is internally audited

```
public static final int CMD_ENUMERATE_ROLES = 33555205
```

Enumerate/list/search roles.

DTO structure on input

Key	Type	Significance
`criteriaIds`	`List<String>`	a list of criteria used to enumerate the roles
`flags`	`long`	flags specifying what fields of the role objects returned to initialize (see Flag#ROLE_ALL)
`offset`	`Integer`	offset from which to start enumeration
`count`	`Integer`	maximum number of objects to return.

DTO structure on output

Key	Type	Significance

`list`	`List<Role>`	the list of roles

Errors

- InvalidInputBusinessException - invalid `criteriaIds` (see Notes for supported criteria combinations)
- DeniedSecurityException - security context does not allow this operation for connected user

Notes

- role names are case insensitive (e.g. creating the role "dummy" when a role "Dummy" exists will generate an error)
- The following table lists the supported criteria combinations

Content of `criteriaIds`	Significance
`[]` - empty	all roles are enumerated
`["LABEL_SW", <expr>]`	roles with `label` starting with <expr>
`["LABEL", <expr>]`	roles with `label` containg <expr>.A minimum of 2 characters requirement for <expr> in this case
`["DESCRIPTION", <expr>]`	roles with `description` containg <expr>.A minimum of 2 characters requirement for <expr> in this case
`["REALNAME", <expr>]`	roles with `realName` containg <expr>.A minimum of 2 characters requirement for <expr> in this case

- in the table above, `"LABEL"`, `"DESCRIPTION"` and `"REALNAME"` can be combined in the same criterion.E.g. specifying `["LABEL REALNAME", <expr>]` will look for <expr> in both `label` and `realName` properties of role objects.`"LABEL_SW"` cannot be part of these combinations
- this operation is internally audited

```
public static final int CMD_CREATE_GROUP_USER = 33555457
```

Create an assignment relationship of a user to a group.

DTO structure on input

Key	Type	Significance
`object`	Assignment	the relationship object.See Prefix#GROUP_USER to

		understand how to set the id of this object.The id of this object must be set on input

DTO structure on output

Key	Type	Significance
`object`	Assignment	the same object as on input

Errors

- InvalidInputBusinessException - invalid `id` (null or empty)
- AlreadyExistsPersistenceException - the relationship already exists
- DeniedSecurityException - security context does not allow this operation for connected user

Notes

- the id of this object must be set on input
- this operation is internally audited

```
public static final int CMD_DELETE_GROUP_USER = 33555460
```

Delete an assignment relationship of a user to a group.

DTO structure on input

Key	Type	Significance
`objectId`	String	the id of the relationship object.See Prefix#GROUP_USER to understand how to set the id of this object

Errors

- InvalidInputBusinessException - invalid `id` (null or empty)
- DeniedSecurityException - security context does not allow this operation for connected user

Notes

- this operation is internally audited

```
public static final int CMD_ENUMERATE_GROUP_USER = 33555461
```

Enumerate/list the assignments between users and groups.

DTO structure on input

Key	Type	Significance
`criteriaIds`	`List<String>`	a list of criteria used to enumerate the

		assignments
offset	Integer	offset from which to start enumeration
count	Integer	maximum number of objects to return.

DTO structure on output

Key	Type	Significance
list	List<Assignment>	the list of assignments

Errors

- InvalidInputBusinessException - invalid `criteriaIds` (see Notes for supported criteria combinations)
- DeniedSecurityException - security context does not allow this operation for connected user

Notes

- The following table lists the supported criteria combinations

Content of `criteriaIds`	Significance
[] - empty	all group-user assignments are listed
[<groupId>]	user assignments to the specified group
[<userId>]	group assignments of the specified user
[<assigId>]	the specified assignment if it exists. can be used to verify if an assignment relationship exists

```
public static final int CMD_CREATE_USER_ROLE = 33555713
```

Create an assignment relationship of a role to a user.

DTO structure on input

Key	Type	Significance
object	Assignment	the relationship object.See Prefix#USER_ROLE to understand how to set the id of this object.The id of this object must be set on input

DTO structure on output

Key	Type	Significance
object	Assignment	the same object as on input

Errors

- InvalidInputBusinessException - invalid `id` (null or empty)
- AlreadyExistsPersistenceException - the relationship already exists
- DeniedSecurityException - security context does not allow this operation for connected user

Notes

- the id of this object must be set on input
- this operation is internally audited

```
public static final int CMD_DELETE_USER_ROLE = 33555716
```

Delete an assignment relationship of a role to a user.

DTO structure on input

Key	Type	Significance
`objectId`	String	the id of the relationship object.See Prefix#USER_ROLE to understand how to set the id of this object

Errors

- InvalidInputBusinessException - invalid `id` (null or empty)
- DeniedSecurityException - security context does not allow this operation for connected user

Notes

- this operation is internally audited

```
public static final int CMD_ENUMERATE_USER_ROLE = 33555717
```

Enumerate/list the assignments between users and roles.

DTO structure on input

Key	Type	Significance
`criteriaIds`	`List<String>`	a list of criteria used to enumerate the assignments
`offset`	`Integer`	offset from which to start enumeration
`count`	`Integer`	maximum number of objects to return.

DTO structure on output

Key	Type	Significance
`list`	`List<Assignment>`	the list of assignments

Errors

- InvalidInputBusinessException - invalid `criteriaIds` (see Notes for supported criteria combinations)
- DeniedSecurityException - security context does not allow this operation for connected user

Notes

- The following table lists the supported criteria combinations

Content of `criteriaIds`	Significance
`[]` - empty	all user-role assignments are listed
`[<roleId>]`	user assignments to the specified role
`[<userId>]`	role assignments of the specified user
`[<assigId>]`	the specified assignment if it exists. can be used to verify if an assignment relationship exists

```
public static final int CMD_CREATE_GROUP_ROLE = 33555969
```

Create an assignment relationship of a role to a group.

DTO structure on input

Key	Type	Significance
`object`	Assignment	the relationship object.See Prefix#GROUP_ROLE to understand how to set the id of this object.The id of this object must be set on input

DTO structure on output

Key	Type	Significance
`object`	Assignment	the same object as on input

Errors

- InvalidInputBusinessException - invalid `id` (null or empty)
- AlreadyExistsPersistenceException - the relationship already exists
- DeniedSecurityException - security context does not allow this operation for connected user

Notes

- the id of this object must be set on input
- this operation is internally audited

```
public static final int CMD_DELETE_GROUP_ROLE = 33555972
```

Delete an assignment relationship of a role to a group.

DTO structure on input

Key	Type	Significance
`objectId`	String	the id of the relationship object.See Prefix#GROUP_ROLE to understand how to set the id of this object

Errors

- InvalidInputBusinessException - invalid `id` (null or empty)
- DeniedSecurityException - security context does not allow this operation for connected user

Notes

- this operation is internally audited

```
public static final int CMD_ENUMERATE_GROUP_ROLE = 33555973
```

Enumerate/list the assignments between groups and roles.

DTO structure on input

Key	Type	Significance
`criteriaIds`	`List<String>`	a list of criteria used to enumerate the assignments
`offset`	`Integer`	offset from which to start enumeration
`count`	`Integer`	maximum number of objects to return.

DTO structure on output

Key	Type	Significance
`list`	`List<Assignment>`	the list of assignments

Errors

- InvalidInputBusinessException - invalid `criteriaIds` (see Notes for supported criteria combinations)
- DeniedSecurityException - security context does not allow this operation for connected user

Notes

- The following table lists the supported criteria combinations

Content of `criteriaIds`	Significance

`[]` - empty	all group-roleassignments are listed
`[<roleId>]`	group assignments of the specified role
`[<groupId>]`	role assignments to the specified group
`[<assigId>]`	the specified assignment if it exists. can be used to verify if an assignment relationship exists

```
public static final int CMD_READ_STORAGE = 33557506
```

Read a storage area's data.

DTO structure on input

Key	Type	Significance
`object`	Storage	the storage object to initialize with persisted data.Must have either `id` property set
`flags`	`long`	Flags specifying what fields of the role object to initialize (see Flag#STORAGE_ALL).

DTO structure on output

Key	Type	Significance
`object`	`Storage`	the same object supplied on input but with the fields specified by `flags` parameter set.

Errors

- InvalidInputBusinessException - invalid `id` specified in input `object`
- NotFoundPersistenceException - object with specified `id` not found
- DeniedSecurityException - security context does not allow this operation for connected user

Notes

- this operation is internally audited

```
public static final int CMD_ENUMERATE_STORAGES = 33557509
```

Enumerate/list storage areas.

DTO structure on input

Key	Type	Significance
`criteriaIds`	`List<String>`	a list of criteria used to enumerate the storage areas

`flags`	`long`	flags specifying what fields of the storage area objects returned to initialize (see Flag#STORAGE_ALL)
`offset`	`Integer`	offset from which to start enumeration
`count`	`Integer`	maximum number of objects to return.

DTO structure on output

Key	Type	Significance
`list`	`List<Storage>`	the list of storages

Errors

- InvalidInputBusinessException - invalid `criteriaIds` (see Notes for supported criteria combinations)
- DeniedSecurityException - security context does not allow this operation for connected user

Notes

- The following table lists the supported criteria combinations

Content of `criteriaIds`	Significance
`[]` - empty	all storages are enumerated

```
public static final int CMD_READ_PORTAL = 33558018
```

Read a portal's data.

DTO structure on input

Key	Type	Significance
`object`	Portal	the portal object to initialize with persisted data.Must have `id` property set
`flags`	`long`	Flags specifying what fields of the role object to initialize (see Flag#PORTAL_ALL).

DTO structure on output

Key	Type	Significance
`object`	`Portal`	the same object supplied on input but with the fields specified by `flags` parameter set.

Errors

- InvalidInputBusinessException - invalid `id` specified in input `object`
- NotFoundPersistenceException - object with specified `id` not found
- DeniedSecurityException - security context does not allow this operation for connected user

Notes

- this operation is internally audited

```
public static final int CMD_ENUMERATE_PORTALS = 33558021
```

Enumerate/list portals.

DTO structure on input

Key	Type	Significance
`criteriaIds`	`List<String>`	a list of criteria used to enumerate the portals
`flags`	`long`	flags specifying what fields of the storage area objects returned to initialize (see Flag#PORTAL_ALL)
`offset`	`Integer`	offset from which to start enumeration
`count`	`Integer`	maximum number of objects to return.

DTO structure on output

Key	Type	Significance
`list`	`List<Portal>`	the list of portals

Errors

- InvalidInputBusinessException - invalid `criteriaIds` (see Notes for supported criteria combinations)
- DeniedSecurityException - security context does not allow this operation for connected user

Notes

- The following table lists the supported criteria combinations

Content of `criteriaIds`	Significance
`[]` - empty	all portals are enumerated
`[<userId>]` - empty	all portals which user specified by `<userId>` can access through direct, group- and role-inherited assignments

```
public static final int CMD_DELEGATE_OWNERSHIP = 33554562
```

Delegate to a user the ownership of all objects in a database beloging to another user.

DTO structure on input

Key	Type	Significance
`fromRoleId`	String	**Required**. The id of the user from whom object ownership is removed
`toUserId`	String	**Required**. The id of the user to whom object ownership is delegated
`databaseId`	String	**Required**. The id of the database containing the objects whose ownership is delegated

DTO structure on output

empty, can be `null`

Validations to perform

- `fromUserId` specifies a valid user id. Failure will result in a com.siatel.error.InvalidInputBusinessException `("fromUserId")`
- `toUserId` specifies a valid user id. Failure will result in a com.siatel.error.InvalidInputBusinessException `("toUserId")`
- user specified by `toUserId` exists. Failure will result in a com.siatel.error.NotFoundPersistenceException `("toUserId")`
- `databaseId` specifies a valid database id. Failure will result in a com.siatel.error.InvalidInputBusinessException `("databaseId")`
- database specified by `databaseId` exists. Failure will result in a com.siatel.error.NotFoundPersistenceException `("databaseId")`

Specific responsabilities

- invoke persistence call that performs the operation

```
public static final int CMD_READ_RESOURCE = 33559554
```

Read a resource's data.

DTO structure on input

Key	Type	Significance
`object`	com.siatel.domain.Resource	the resource object to initialize

		with persisted data.Must have `id` property set
`flags`	`long`	Flags specifying what fields of the role object to initialize (see Flag#RESOURCE_ALL).

DTO structure on output

Key	Type	Significance
`object`	`com.siatel.domain.Resource`	the same object supplied on input but with the fields specified by `flags` parameter set.

Errors

- InvalidInputBusinessException - invalid `id` specified in input `object`
- NotFoundPersistenceException - object with specified `id` not found
- DeniedSecurityException - security context does not allow this operation for connected user

Notes

- this operation is internally audited

```
public static final int CMD_ENUMERATE_RESOURCES = 33559557
```

Enumerate/list resources.

DTO structure on input

Key	Type	Significance
`criteriaIds`	`List<String>`	a list of criteria used to enumerate the resources
`flags`	`long`	flags specifying what fields of the resource objects returned to initialize (see Flag#RESOURCE_ALL)
`offset`	`Integer`	offset from which to start enumeration
`count`	`Integer`	maximum number of objects to return.

DTO structure on output

Key	Type	Significance
`list`	`List<com.siatel.domain.Resource>`	the list of portals

Errors

- InvalidInputBusinessException - invalid `criteriaIds` (see Notes for supported criteria combinations)
- DeniedSecurityException - security context does not allow this operation for connected user

Notes

- The following table lists the supported criteria combinations

Content of `criteriaIds`	Significance
`[]` - empty	all resources are enumerated

Constructor detail :

```
public AdministrationLibrary ()
```

Package com.siatel.framework.service.audit

This package groups audit related service definitions and utilities.

Classes

AuditLibrary

```
public class AuditLibrary
```

Author :

victorc

Field summary :

Modifier and type	Field
`public static final int`	CMD_QUERY
`public static final int`	CMD_COUNT

Constructor summary :

Constructor
AuditLibrary ()

Field detail :

```
public static final int CMD_QUERY = 1342226458
```

Query audit log.

DTO structure on input

Key	Type	Significance
criteriaIds	List<String>	criteria ids to be used for the query.Normally some resources ids to filter the domain
offset	Integer	the offset of the first entry to return. If not specified, default value used is 0.
count	Integer	the maximum number of entries to return. If not specified, then no limit is set for the number of entries to return
dateMin	Date	lower boundary of the time domain to seek. This parameter is optional
dateMax	Date	upper boundary of the time domain to seek. This parameter is optional
flags	long	a group of flags to further customize the query. No flags are defined at this moment

DTO structure on output

Key	Type	Significance
list	List<AuditEntry>	the list of entries returned by the query. Can be empty but not null

Errors

- InvalidInputBusinessException - invalid criteriaIds.See Notes section for supported criteria combinations

Notes

- The following table lists the supported criteria combinations

Content of criteriaIds	Significance
[Constants#ADMIN_OBJS,	<resourceId> is an id of an

<resourceId>]	administration object (group, role, database, portal, storage, resource etc) whose log is requested.
[Constants#DESIGN_OBJS, <resourceId>]	<resourceId> is an id of a design object (form, view, process definition, script etc) whose log is requested.
[Constants#DESIGN_OBJS, <databaseId>, <userId>]	<userId> is an id of a user whose design activity log (what did he do) is requested for <databaseId>-specified database.
[Constants#PRODUCTION_OBJS, <resourceId>]	<resourceId> is an id of a production object (folder, process, document etc) whose log is requested.Can also be a database id, in which case the audit of all objects in specified database is queried
[Constants#PRODUCTION_OBJS, <resourceId>, <userId>]	same as above but filtered down to the activities performed by user specified by <userId>

```
public static final int CMD_COUNT = 1342226445
```

Count the audit entries given specified criteria. This call is used in before a #CMD_QUERY to enable page-by-page display.

DTO structure on input

Key	Type	Significance
criteriaIds	List<String>	criteria ids to be used for the query.Normally some resources ids to filter the domain
dateMin	Date	lower boundary of the time domain to seek. This

		parameter is optional
`dateMax`	`Date`	upper boundary of the time domain to seek. This parameter is optional

DTO structure on output

Key	Type	Significance
`count`	`Integer`	the count of entries matching specified criteria

Errors

- `InvalidInputBusinessException` - invalid `criteriaIds`.See Notes and #CMD_QUERY for supported criteria combinations

Notes

- the list of supported criteria is the same as for the #CMD_QUERY command since this call is used to know in advance the count of (and most likely paginate) the results of a the #CMD_QUERY

Constructor detail :

```
public AuditLibrary ()
```

Package com.siatel.framework.service.auth

This package groups authentication related service definitions and utilities.

Classes

AuthenticationLibrary

`public class AuthenticationLibrary`

This class defines the id's for the authentication service commands.

Since :

01/03/2007

Author :

Victor Citiriga

Field summary :

Modifier and type	Field
public static final int	CMD_PING
public static final int	CMD_AUTHENTICATE
public static final int	CMD_LOGIN
public static final int	CMD_LOGOUT

Constructor summary :

Constructor
AuthenticationLibrary ()

Field detail :

```
public static final int CMD_PING = 0
```

Ping. Useful to keep a user connection alive.

DTO structure on input (to Command#invoke)

empty, `null` can be passed

DTO structure on output (from Command#invoke)

empty, `null` will be returned

Errors

- InvalidConnectionBusinessException - invalid `conn` (null or expired connection)

```
public static final int CMD_AUTHENTICATE = 640
```

Authenticate a set of credentials.

DTO structure on input (to Command#invoke)

Key	Type	Significance
user	String	The user login name
pass	String	The password

DTO structure on output (from Command#invoke)

empty, `null` will be returned

Specific responsabilities

- validate and authenticate the supplied user credentials

Errors

- InvalidCredentialsAuthenticationException - user and password are not authenticated. The message of the exception is an integer number specifying the number of attempts left. If null or '-1' then unlimited number of attempts is left.
- DisabledAccountAuthenticationException - account specified by user name is locked.

Notes

- the `conn` parameter of the Command#invoke is ignored by this command and should be `null`.

```
public static final int CMD_LOGIN = 641
```

Login using a set of credentials.

DTO structure on input (to Command#invoke)

Key	Type	Significance
`user`	`String`	The user login name
`pass`	`String`	The password
`context`	`String`	One of the constants in AuthContext representing the usage context user connection. This is normally associated with a client functionality which is license controlled

DTO structure on output (from Command#invoke)

Key	Type	Significance
`conn`	`String`	The connection token, to be used when calling functionality that requires authenticated access.

Specific responsabilities

- authenticate the supplied user credentials. In case supplied `user` is `null` or empty string then anonymous login is assumed.
- generate and return the connection string.

Errors

- InvalidCredentialsAuthenticationException - user and password are not authenticated
- UserAlreadyConnectedException - user and password are authenticated but an active connection for the same credentials exists from another location.
- FrameworkStateException - the framework state does not allow this connection (only administrator web client connections are allowed when system state requires and update).
- DisabledAccountBusinessException - the account user for login is disabled.
- AccountAccessForbiddenByPlannerException - the user schedule does not allow the connection if such restriction is enabled.
- InvalidInputBusinessException - value of `context` is not specified as one of the constants define in AuthContext.
- ConnectionsLimitExceededException - the server license does not allow this connection.
- NotLicensedException - the server license does not allow this connection.

Notes

- the `conn` parameter of the Command#invoke is ignored by this command and should be `null`.

```
public static final int CMD_LOGOUT = 642
```

Logout.

DTO structure on input (to Command#invoke)

empty, `null` can be passed

DTO structure on output (from Command#invoke)

empty, `null` will be returned

Notes

- release the connection context associated with the `conn` parameter.

Constructor detail :

```
public AuthenticationLibrary ()
```

Package com.siatel.framework.service.design

This package groups design related service definitions and utilities.

Classes

DesignLibrary

```
public final class DesignLibrary
```

This class defines the design library commands and their structure

Field summary :

Modifier and type	Field
public static final int	CMD_ENUMERATE_TYPES
public static final int	CMD_ENUMERATE_ATTRIBUTES
public static final int	CMD_ENUMERATE_FORMS
public static final int	CMD_ENUMERATE_GRIDS

Constructor summary :

Constructor
DesignLibrary ()

Field detail :

```
public static final int CMD_ENUMERATE_TYPES = 50340613
```

Enumerate user defined types.

DTO structure on input

Key	Type	Significance
criteriaIds	List<String>	the enumeration criteria
flags	long	the flags used to read each type definition
offset	Integer	an offset from where to start then enumeration
count	Integer	a maximum number of returned results

DTO structure on output

Key	Type	Significance
list	List<Type>	the list of types

Errors

- InvalidInputBusinessException - invalid `criteriaIds` or `flags`

Notes

- The following table lists the supported criteria combinations

Content of `criteriaIds`	Significance
[<databaseId>]	the types in the database specified by `databaseId` will be listed

```
public static final int CMD_ENUMERATE_ATTRIBUTES = 50340869
```

Enumerate user defined attributes.

DTO structure on input

Key	Type	Significance
`criteriaIds`	`List<String>`	the enumeration criteria
`flags`	`long`	the flags used to read each attribute definition
`offset`	`Integer`	the offset from where to start then enumeration
`count`	`Integer`	the maximum number of returned results
`localized`	`String`	the java.util.Locale 2-letter language specifier

DTO structure on output

Key	Type	Significance
`list`	`List<Attribute>`	the list of attributes

Errors

- InvalidInputBusinessException - invalid `criteriaIds` or `flags`

Notes

- The following table lists the supported criteria combinations

Content of `criteriaIds`	Significance
[<databaseId>]	the attributes in the database specified by `databaseId` will be listed

```
public static final int CMD_ENUMERATE_FORMS = 50341125
```

Enumerate forms.

DTO structure on input

Key	Type	Significance
`criteriaIds`	`List<String>`	enumeration criteria
`flags`	`long`	flags used to read each form
`offset`	`Integer`	offset from where to start then enumeration
`count`	`Integer`	maximum number of returned results

DTO structure on output

Key	Type	Significance
`list`	`List<Form>`	the list of forms

Errors

- InvalidInputBusinessException - invalid `criteriaIds`

Notes

- The following table lists the supported criteria combinations

Content of `criteriaIds`	Significance
[<databaseId>]	the forms in the database specified by `databaseId` will be listed

```
public static final int CMD_ENUMERATE_GRIDS = 50355461
```

Enumerate user defined grids.

DTO structure on input

Key	Type	Significance
`criteriaIds`	`List<String>`	the enumeration criteria
`flags`	`long`	the flags used to read each grid definition
`offset`	`Integer`	an offset from where to start the enumeration
`count`	`Integer`	a maximum number of returned results

DTO structure on output

Key	Type	Significance
`list`	`List<Grid>`	the list of grids

Errors

- InvalidInputBusinessException - invalid `criteriaIds` or `flags`

Notes

- The following table lists the supported criteria combinations

Content of `criteriaIds`	Significance
[<databaseId>]	the grids in the database specified by `databaseId` will be listed

Constructor detail :

```
public DesignLibrary ()
```

Package com.siatel.framework.service.flow

This package groups flow related service definitions and utilities.

Classes

FlowLibrary

```
public class FlowLibrary
```

Library of command id's for the 'workflow' service access point.

Author :

victorc

Field summary :

Modifier and type	Field
public static final int	CMD_CREATE_PROCESS
public static final int	CMD_READ_PROCESS
public static final int	CMD_UPDATE_PROCESS
public static final int	CMD_DELETE_PROCESS
public static final int	CMD_START_PROCESS
public static final int	CMD_GET_INBOX_SUMMARY

Constructor summary :

Constructor
FlowLibrary ()

Field detail :

```
public static final int CMD_CREATE_PROCESS = 167792129
```

Create a process. This functionality is more like creating a new "draft" message before sending it.

DTO structure on input

Key	Type	Significance
object	com.siatel.domain.Process	the object to create

DTO structure on output

Key	Type	Significance
object	com.siatel.domain.Process	the same object passed in the input dto but initialized with id and other first time initialization relevant members set

Errors

- InvalidInputBusinessException - invalid `object.definitionId` or `object.formId`
- NotFoundPersistenceException - one of the objects specified by `object.definitionId` or `object.formId` do not exist
- ObjectNotReadyBusinessException - one of the objects specified by `object.definitionId` or `object.formId` are not deployed

Notes

- the operation is internally audited

```
public static final int CMD_READ_PROCESS = 167792130
```

Read a process.

DTO structure on input

Key	Type	Significance
`object`	`com.siatel.domain.Process`	the object to populate with data, must have `id` field set
`flags`	`com.siatel.domain.Flag`	combination of process object field flags specifying object properties to load

DTO structure on output

Key	Type	Significance
`object`	`com.siatel.domain.Process`	the same object passed in the input dto but initialized with all the fields specified in the `flags` field

Errors

- InvalidInputBusinessException - input `object.id` is invalid
- NotFoundPersistenceException - object with requested `object.id` does not exist
- DeniedSecurityException - security context does not allow operation for caller

Notes

- the operation is internally audited

```
public static final int CMD_UPDATE_PROCESS = 167792131
```

Update a process.

DTO structure on input (to Command#invoke)

Key	Type	Significance

object	com.siatel.domain.Process	the object to update, must have the id field set, along with the fields to update
flags	com.siatel.domain.Flag	combination of Process object field flags
formPage	String	the name of the form page used for custom layout forms when the update command was invoked. If not specified then the default page it is assumed in case form is still custom layout

DTO structure on output (from Command#invoke)

Key	Type	Significance
object	com.siatel.domain.Process	the same object passed in the input dto

Errors

- InvalidInputBusinessException - input object.id is invalid
- DeniedSecurityException - security context does not allow operation for caller

Notes

- the operation is internally audited

```
public static final int CMD_DELETE_PROCESS = 167792132
```

Delete a process.

DTO structure on input

Key	Type	Significance
objectId	String	the id of the object to delete

DTO structure on output

null

Errors

- InvalidInputBusinessException - invalid objectId
- DeniedSecurityException - security context does not allow operation for caller

Notes

- the operation is internally audited

```
public static final int CMD_START_PROCESS = 167792159
```

Start a process. The process must have been previously created (like a draft).

DTO structure on input

Key	Type	Significance
`objectId`	`String`	the id of the process to start
`startPoint`	`String`	the process start point

DTO structure on output (from Command#invoke)

`null`

Errors

- InvalidInputBusinessException - invalid `objectId`
- InvalidInputBusinessException - process specified by `objectId` does not exist
- DeniedSecurityException - security context does not allow operation for caller

Notes

- the operation is internally audited

```
public static final int CMD_GET_INBOX_SUMMARY = 167792768
```

Query inbox summary.

DTO structure on input

Empty, can be `null`

DTO structure on output

Key	Type	Significance
`"INBOX"`	`Integer`	Count of tasks in inbox of connected user
`"POSTPONED"`	`Integer`	Count of tasks in waiting box
`"OVERDUE"`	`Integer`	Count of tasks overdue
`"EXPIRING"`	`Integer`	Count of tasks in inbox and waiting tray that are expiring today

Errors

None.

Notes

None.

Constructor detail :

```
public FlowLibrary ()
```

Package com.siatel.framework.service.ged

This package groups EDM related service definitions and utilities.

Classes

GedLibrary

```
public class GedLibrary
```

Library of command id's for the 'document management' service access point.

Author :

victorc

Field summary :

Modifier and type	Field
public static final int	CMD_CREATE_FOLDER
public static final int	CMD_READ_FOLDER
public static final int	CMD_UPDATE_FOLDER
public static final int	CMD_DELETE_FOLDER
public static final int	CMD_ENUMERATE_FOLDERS
public static final int	CMD_COUNT_FOLDER
public static final int	CMD_MOVE_FOLDER
public static final int	CMD_GET_INFO_FOLDER
public static final int	CMD_POSITION_FOLDER
public static final int	CMD_REORDER_FOLDER_CONTENT
public static final	CMD_CREATE_DOCUMENT

int	
public static final int	CMD_READ_DOCUMENT
public static final int	CMD_UPDATE_DOCUMENT
public static final int	CMD_DELETE_DOCUMENT
public static final int	CMD_ENUMERATE_DOCUMENTS
public static final int	CMD_COPY_DOCUMENT
public static final int	CMD_ROLLBACK_DOCUMENT
public static final int	CMD_MOVE_DOCUMENT
public static final int	CMD_POSITION_DOCUMENT
public static final int	CMD_ADD_DOCUMENT_IN_FULLTEXT
public static final int	CMD_REMOVE_DOCUMENT_FROM_FULLTEXT
public static final int	CMD_REINDEX_DOCUMENT_IN_FULLTEXT
public static final int	CMD_REINDEX_NOTE_IN_FULLTEXT
public static final int	CMD_ENUMERATE_CLASSELEMENTS
public static final int	CMD_RENDER_CLASSELEMENT_PARENT
public static final int	CMD_PIN_DOCUMENTS
public static final int	CMD_UNPIN_DOCUMENTS
public static final int	CMD_MOVE_PIN
public static final int	CMD_CREATE_NOTE

public static final int	CMD_READ_NOTE
public static final int	CMD_UPDATE_NOTE
public static final int	CMD_DELETE_NOTE
public static final int	CMD_ENUMERATE_NOTES
public static final int	CMD_CREATE_VERSION
public static final int	CMD_READ_VERSION
public static final int	CMD_DELETE_VERSION
public static final int	CMD_ENUMERATE_VERSIONS
public static final int	CMD_COUNT_VERSION
public static final int	CMD_CHECK_OBJECT
public static final int	CMD_LOCK_DOCUMENT
public static final int	CMD_UNLOCK_DOCUMENT
public static final int	CMD_CHECKOUT_DOCUMENT
public static final int	CMD_CHECKIN_DOCUMENT
public static final int	CMD_MOVE_ACTIVE_ARCHIVE
public static final int	CMD_MOVE_ARCHIVE_ACTIVE
public static final int	CMD_COUNT_DOCUMENT
public static final int	CMD_CREATE_COMMENT
public static final	CMD_UPDATE_COMMENT

int	
public static final int	CMD_DELETE_COMMENT
public static final int	CMD_ENUMERATE_COMMENTS
public static final int	CMD_READ_COMMENT
public static final int	CMD_COUNT_COMMENTS
public static final int	CMD_READ_DOCUMENT_BY_UUID

Constructor summary :

Constructor
GedLibrary ()

Field detail :

```
public static final int CMD_CREATE_FOLDER = 67126273
```

Create a folder.

DTO structure on input

Key	Type	Significance
object	Folder	the folder to create
parentId	String	id of the parent folder, unless specified by the object parameter's parentId field

DTO structure on output

Key	Type	Significance
object	Folder	the object supplied on input with the initialization-relevant fields set

Errors

- InvalidInputBusinessException - neither parentId nor object.parentId are valid
- InvalidInputBusinessException - object has null, empty or blank label
- DeniedSecurityException - security context does not allow operation for caller

Notes

- the operation is internally audited

- security is also initialized for the created folder object depending on the database security context and security rules that apply

```
public static final int CMD_READ_FOLDER = 67126274
```

Read a folder.

DTO structure on input

Key	Type	Significance
`object`	Folder	the folder object to initialize with data from persistence; object needs to have the `id` property set
`flags`	`long`	a combination of Flag constants specifying what object fields to initialize

DTO structure on output

Key	Type	Significance
`object`	Folder	the same object supplied on input but with `flags` requested fields set

Errors

- InvalidInputBusinessException - input `object.id` is invalid
- NotFoundPersistenceException - object with requested `object.id` does not exist
- DeniedSecurityException - security context does not allow operation for caller

Notes

- the operation is internally audited

```
public static final int CMD_UPDATE_FOLDER = 67126275
```

Update a folder.

DTO structure on input

Key	Type	Significance
`object`	Folder	the folder object to update; the `id` property must be set along with the other fields to update
`flags`	`long`	a combination of Flag constants specifying what object fields to update
`formPage`	`String`	the name of the form page used for custom layout forms when the update command was invoked. If

		not specified then the default page is assumed in case form is still custom layout

DTO structure on output

Key	Type	Significance
`object`	Folder	the same object supplied on input

Errors

- InvalidInputBusinessException - input `object.id` is invalid
- DeniedSecurityException - security context does not allow operation for caller

Notes

- the operation is internally audited

```
public static final int CMD_DELETE_FOLDER = 67126276
```

Delete a folder.

DTO structure on input

Key	Type	Significance
`objectId`	`String`	the id of the folder object to delete

DTO structure on output

empty, `null` will be returned

Errors

- InvalidInputBusinessException - supplied `objectId` is invalid
- DeniedSecurityException - security context does not allow operation for caller

Notes

- the operation is internally audited
- the folder and it's contents are deleted recursively

```
public static final int CMD_ENUMERATE_FOLDERS = 67126277
```

Enumerate folders.

DTO structure on input

Key	Type	Significance
`criteriaIds`	`List<String>`	a list of criteria to use as filters for enumeration

offset	int	offset for enumeration start
count	int	the maximum number of elements to return
flags	long	a combination of Flag constants specifying what object fields to initialize for returned objects

DTO structure on output

Key	Type	Significance
list	List<Folder>	the list of folders matching enumeration criteria

Errors

- InvalidInputBusinessException - criteriaIds does not conform to one of the supported criteria combinations (see notes)
- DeniedSecurityException - security context does not allow operation for caller

Notes

- The following table lists the supported criteria combinations

Content of criteriaIds	Significance
[<folderId>]	the children folders of folder specified by folderId will be enumerated

```
public static final int CMD_COUNT_FOLDER = 67126285
```

Count folders in a container (folder including archive supported versions).

DTO structure on input

Key	Type	Significance
criteriaIds	List<String>	criteria to use for counting.Currently supported criteria are same as for folder enumeration.

DTO structure on output

Key	Type	Significance
count	Long	the folders count

Errors

- InvalidInputBusinessException - invalid criteriaIds

Notes

- this command is usually used when a page-by-page display of a large list of folders is desired
- see GedLibrary#CMD_ENUMERATE_FOLDERS for supported `criteriaIds` combinations

```
public static final int CMD_MOVE_FOLDER = 67126287
```

Move a folder to another location (folder/database root).

DTO structure on input

Key	Type	Significance
`objectId`	`String`	id of folder object to move
`targetId`	`String`	id of new location. Database root can be specified as id of folder with numeric id 0 from said database.

DTO structure on output

empty, `null` will be returned

Errors

- InvalidInputBusinessException - `objectId` or `targetId` are invalid
- DeniedSecurityException - security context does not allow operation for caller

Notes

- the operation is internally audited
- object and target must be in the same database, otherwise operation will fail

```
public static final int CMD_GET_INFO_FOLDER = 67126290
```

Get statistics about a folder's contents.

DTO structure on input

Key	Type	Significance
`objectId`	`String`	the id of the folder whose statistics are queried
`childrenFlags`	`long`	statistics flags (recursivity, what to count etc). See com.siatel.domain.IEFlag

DTO structure on output

Key	Type	Significance
`FLDR`	`Long`	the count of subfolders
`FLDR_objsize`	`Long`	the cumulated disk-size of subfolders
`DOCF`	`Long`	the count of documents

DOCF_objsize	Long	the cumulated disk-size of documents
DOCF_FILE	Long	the cumulated disk-size of the document files
FCHF	Long	the count of index cards
FCHF_objsize	Long	the cumulated disk-size of index cards
PINF	Long	the count of pins
PINF_objsize	Long	the cumulated disk-size of pins
NTDF	Long	the count of notes
NTDF_objsize	Long	the cumulated disk-size of notes
NTDF_FILE	Long	the cumulated disk-size of note files
VNDF	Long	the count of dcoument versions
VNDF_objsize	Long	the cumulated disk-size of document versions
VNDF_FILE	Long	the cumulated disk-size of document version files

Errors

- InvalidInputBusinessException - `objectId` or `targetId` are invalid
- DeniedSecurityException - security context does not allow operation for caller
- `objectId` is specified and indicates a folder other than root.Failure will result in a InvalidInputBusinessException.
- DeniedSecurityException - security context does not allow operation for caller

Notes

- this command will navigate folder contents and collect data so be weary when calling this command on a large folder as it may take quite a lot of time and may trigger a response timeout

```
public static final int CMD_POSITION_FOLDER = 67126288
```

Set a folder's position within it's parent.

DTO structure on input

Key	Type	Significance
objectId	String	id of folder whose position is set
afterId	String	id of folder after which the folder specified by `objectId` will be positioned. The 'first' position will be specified as as id with numeric 0.The 'last' position within it's parent will be specified as id with numerics 0x7FFFFFFF. The database numeric id will be ignored if 'first', or 'last' are requested

DTO structure on output

empty, `null` will be returned

Errors

- InvalidInputBusinessException - `objectId` or `afterId` are invalid
- DeniedSecurityException - security context does not allow operation for caller

Notes

- `objectId` cannot specify the database root folder
- `afterId` must specify a folder within parent of `objectId`
- this operation is internally audited

```
public static final int CMD_REORDER_FOLDER_CONTENT = 67126404
```

Reorder folder content alphabetically.

DTO structure on input

Key	Type	Significance
`folderId`	`String`	id of folder which content will be ordered alphabetically ascending or descending
`subFolders`	`Long`	`0` means only the immediate children of the `folderId` will be reordered, `1` means all sub-tree will be reordered
`flags`	`Long`	Flag#GENERIC_ASC means ascending, Flag#GENERIC_DESC means descending

DTO structure on output

empty, `null` will be returned

Errors

- InvalidInputBusinessException - `folderId`is invalid
- DeniedSecurityException - security context does not allow operation for caller

Notes

- this operation is internally audited

```
public static final int CMD_CREATE_DOCUMENT = 67126529
```

Create a document.

DTO structure on input

Key	Type	Significance
`object`	Document	The document object to create
`parentId`	`String`	id of folder/process/pin where to create the document
`add2ft`	`Boolean`	`true` if document content should be indexed (text documents only)

DTO structure on output

Key	Type	Significance
`object`	Document	The same object supplied on input but with `id`, `dateC` and other initialization-time relevant fields set

Errors

- InvalidInputBusinessException - `object.label` not supplied or invalid `parentId`
- DeniedSecurityException - security context does not allow operation for caller

Notes

- this operation is internally audited

```
public static final int CMD_READ_DOCUMENT = 67126530
```

Read a document object.

DTO structure on input

Key	Type	Significance
`object`	Document	the document object to initialize with persisted data.
`flags`	`long`	the fields of the Document object to initialize with server data. A combination of Flag defined constants, relevant for the Document class objects

DTO structure on output

Key	Type	Significance
`object`	Document	The same object supplied on input but with the fields specified by `flags` parameter set.

Errors

- InvalidInputBusinessException - `object.id` is invalid

- NotFoundPersistenceException - object with requested `object.id` does not exist
- DeniedSecurityException - security context does not allow operation for caller

Notes

- this operation is audited

```
public static final int CMD_UPDATE_DOCUMENT = 67126531
```

Update a document.

DTO structure on input

Key	Type	Significance
`object`	Document	the document object to update
`flags`	`long`	the fields of the Document object to update. A combination of Flag defined constants, relevant for the Document class objects
`formPage`	`String`	the name of the form page used for custom layout forms when the update command was invoked. If not specified then the default page it is assumed in case form is still custom layout

DTO structure on output

Key	Type	Significance
`object`	Document	The same object supplied on input.

Errors

- InvalidInputBusinessException - input `object.id` is invalid
- DeniedSecurityException - security context does not allow operation for caller

Notes

- the operation is internally audited

```
public static final int CMD_DELETE_DOCUMENT = 67126532
```

Delete a document.

DTO structure on input

Key	Type	Significance
`objectId`	`String`	the id of the document object to delete

DTO structure on output

Key	Type	Significance
ficheId	String	id of resulting index card (fiche) document if document to delete was last component of a pin. This parameter will be set if such pin to index card conversion occurs

Errors

- InvalidInputBusinessException - supplied objectId is invalid
- DeniedSecurityException - security context does not allow operation for caller

Notes

- the operation is internally audited
- the document file is also removed from storage

```
public static final int CMD_ENUMERATE_DOCUMENTS = 67126533
```

Enumerate documents.

DTO structure on input

Key	Type	Significance
criteriaIds	List<String>	a list of criteria to use as filters for enumeration
flags	long	the fields of the Document object to initialize with server data. A combination of Flag defined constants, relevant for the Document class objects. Also, a list of order options defined bythe Flag class can be passed through this parameter
offset	Integer	offset from which to start enumeration. E.g. if offset is 2, then the enumeration will return elements starting with the third document.
count	Integer	number of maximum objects to return.

DTO structure on output

Key	Type	Significance
list	List<Document>	the list of documents matching enumeration criteria. See the Notes section for supported criteria combinations

Errors

- InvalidInputBusinessException - criteriaIds does not conform to one of the supported

criteria combinations (see notes)

- DeniedSecurityException - security context does not allow operation for caller

Notes

- The following table lists the supported criteria combinations

Content of `criteriaIds`	Significance
[<parentId>]	the documents in folder/process/pin specified by `parentId` will be enumerated

```
public static final int CMD_COPY_DOCUMENT = 67126542
```

Copy a document to another location (folder/pin/process).

DTO structure on input

Key	Type	Significance
`objectId`	`String`	the id of the document to copy
`formId`	`String`	the id of the form of the resulting document
`targetId`	`String`	the id of the target parent of the document.Can be a folder, process, pin or a classification element that renders to a folder or process id
`startOption`	`String`	**Optional**. Can be `"NEW"`, `"COPY"` or `"REFERENCE"`. `"NEW"` means the new document has no link to source document, `"COPY"` means the new document will contain a link to the source document allowing for later content/meta-data synchronization, `"REFERENCE"`means the new document is a place-holder for the content and meta-data of the source document.If not specified implementation defaults to `"NEW"`

DTO structure on output

Key	Type	Significance
`objectId`	`String`	id of the resulting document

Errors

- InvalidInputBusinessException - `objectId` is neither document nor pin id
- InvalidInputBusinessException - `targetId` is neither folder/process/pin id nor the id of a

classification element that renders to a folder/process id

- InvalidInputBusinessException - if specified `startOption` is valid
- DeniedSecurityException - security context does not allow operation for caller

Notes

- the operation is internally audited
- the document file is duplicated at storage level in case of "NEW" and "COPY"

```
public static final int CMD_ROLLBACK_DOCUMENT = 67126537
```

Rollback document.

DTO structure on input

Key	Type	Significance
`versionId`	`String`	the id of the version to rollback to.It is self-sufficient as it can be used to retrieve the identification of the versioned document

or

Key	Type	Significance
`objectId`	`String`	the id of the versioned document; this parameter is always used with the `version` parameter
`version`	`long`	the version number of the version to rollback to; this parameter is always used with the `objectId` parameter

DTO structure on output

Key	Type	Significance
`object`	Document	the updated document object (id, label etc. but no meta-data/attributes)

Errors

- InvalidInputBusinessException - if none of the two DTO structures is supplied on input
- InvalidInputBusinessException - if specified `version` indicates an invalid version
- NotFoundPersistenceException - if ids in `objectId` or `versionId` indicate objects which do not exist (have been deleted for example)
- DeniedSecurityException - security context does not allow operation for caller

Notes

- the operation is internally audited

- the high int and low int of the `long version` parameter are major and minor vesion designators
- the intermediate versions are deleted along with their files
- document content is reindexed if database is fulltext enabled and document is of text content type
- the containing pin's size is updated if document is pin-contained

```
public static final int CMD_MOVE_DOCUMENT = 67126543
```

Move a document to a specific location (folder/process/pin).

DTO structure on input

Key	Type	Significance
`objectId`	`String`	the id of the document object to move
`targetId`	`String`	id of folder, process or pin where to move the document

DTO structure on output

Key	Type	Significance
`objectId`	`String`	the new id of object (document id changes reflecting the numeric id of the new immediate parent)
`ficheId`	`String`	id of resulting index card (fiche) document if document to delete was last component of a pin. This parameter will be set if such pin to index card conversion occurs (e.g. athe last document in a pin is moved to anotehr location)

Errors

- InvalidInputBusinessException - invalid `objectId` or `targetId`
- DeniedSecurityException - security context does not allow operation for caller

Notes

- the operation is internally audited
- pin to index card (fiche) cnversion occurs when the moved document was the last component of a pinned document

```
public static final int CMD_POSITION_DOCUMENT = 67126544
```

Set a documents's position within it's container (folder, process, pin).

DTO structure on input

Key	Type	Significance
`objectId`	`String`	id of document whose position is set
`afterId`	`String`	id of document after which the document specified by `objectId` will be positioned. The 'first' position will be specified as document id with numeric 0.The 'last' position within it's parent will be specified as id with numeric 0x7FFFFFFF.The database numeric id will be ignored if 'first', or 'last' are requested

DTO structure on output

empty, `null` will be returned

Errors

- InvalidInputBusinessException - invalid `objectId` or `afterId`
- InvalidInputBusinessException - `afterId` specifies id within same document container as `objectId`
- NotFoundpersistenceException - `objectId` or `afterId` specify inexisnte objects
- DeniedSecurityException - security context does not allow operation for caller

Notes

- the operation is internally audited

```
public static final int CMD_ADD_DOCUMENT_IN_FULLTEXT = 67126661
```

Index the content of a text document in the fulltext database.

DTO structure on input

Key	Type	Significance
`objectId`	`String`	id of document whose content to index

DTO structure on output

Key	Type	Significance
`object`	Document	the object with `flags` parameter set.

Errors

- UnsupportedCommandBusinessException - database containing document is not fulltext index enabled
- InvalidInputBusinessException - invalid `objectId`
- NotFoundPersistenceException - `objectId` specifies an existing document

- DeniedSecurityException - security context does not allow operation for caller

Notes

- the document status flags are updated to reflect the new index status
- the operation is internally audited

```
public static final int CMD_REMOVE_DOCUMENT_FROM_FULLTEXT = 67126664
```

Delete the content index of a text document from the fulltext database.

DTO structure on input

Key	Type	Significance
`objectId`	`String`	id of document whose content index to delete

DTO structure on output

Key	Type	Significance
`object`	Document	the object with `flags` parameter set.

Errors

- InvalidInputBusinessException - invalid `objectId`
- DeniedSecurityException - security context does not allow operation for caller

Notes

- the document status flags are updated to reflect the new index status
- the operation is internally audited

```
public static final int CMD_REINDEX_DOCUMENT_IN_FULLTEXT = 67126667
```

Re-index the content of a text document.

DTO structure on input

Key	Type	Significance
`objectId`	`String`	id of document whose content to index

DTO structure on output

Key	Type	Significance
`object`	Document	the object with `flags` parameter set.

Errors

- InvalidInputBusinessException - invalid `objectId`

- DeniedSecurityException - security context does not allow operation for caller

Notes

- the document status flags are updated to reflect the resulting index status (in case of failure, the corresponding status flag is removed)
- the operation is internally audited

```
public static final int CMD_REINDEX_NOTE_IN_FULLTEXT = 67127435
```

Re-index the content of a note.

DTO structure on input

Key	Type	Significance
`objectId`	`String`	id of note whose content to index

DTO structure on output

`null`

Errors

- InvalidInputBusinessException - invalid `objectId`
- DeniedSecurityException - security context does not allow operation for caller

Notes

- the operation is internally audited

```
public static final int CMD_ENUMERATE_CLASSELEMENTS = 67127813
```

Enumerate classification elements (view navigation).

DTO structure on input

Key	Type	Significance
`criteriaIds`	`List<String>`	a list of criteria to use as filters for enumeration
`offset`	`int`	offset for enumeration start
`count`	`int`	the maximum number of elements to return
`flags`	`long`	a combination of Flag constants specifying what object fields to initialize for returned objects

DTO structure on output

Key	Type	Significance
`list`	`List<ClassificationElement>`	the list of classification elements matching

		enumeration criteria.

Errors

- InvalidInputBusinessException - `criteriaIds` does not conform to one of the supported criteria combinations (see notes)

Notes

- The following table lists the supported criteria combinations

Content of `criteriaIds`	Significance
[<classEmenentId>]	the children class elements of element specified by `classEmenentId` will be enumerated
[<databaseId>]	the top level class elements database specified by `databaseId` will be enumerated

```
public static final int CMD_RENDER_CLASSELEMENT_PARENT = 67127858
```

Render a classification element's realized parent object id (folder/process/database root).

DTO structure on input

Key	Type	Significance
`classId`	`String`	the classification element object whose parent id to render

DTO structure on output

Key	Type	Significance
`objectId`	`String`	the id of the realized parent

Errors

- InvalidInputBusinessException - invalid `classId`

Notes

- this command will navigate classification elements upwards to the database root and return the first folder or process id associated with a parent class. If database root is reached with no folder or process encountered, then folder with numeric id 0 of containing database is returned (i.e the database root)

```
public static final int CMD_PIN_DOCUMENTS = 67126656
```

Pin a group of documents.

DTO structure on input

Key	Type	Significance
`object`	Document	the 'pin' document to initialize. This will be the resulting pin. On return it will be initialized with fields relevant to a 'create' operation
`docsIds`	`List<String>`	the list of documents to pin specified as id-s
`parentId`	`String`	the id of the folder or process where the 'pin' document will be created

DTO structure on output

Key	Type	Significance
`object`	Document	the resulting pin, with `id` initialized

Errors

- InvalidInputBusinessException - invalid `object.label`
- InvalidInputBusinessException - invalid `docsIds` (null, empty, or ids of non-document class objects)
- InvalidInputBusinessException - invalid `parentId`
- DeniedSecurityException - security context does not allow operation for caller

Notes

- the operation is internally audited

```
public static final int CMD_UNPIN_DOCUMENTS = 67126657
```

Unpin pin/pinned documents.

DTO structure on input

Key	Type	Significance
`docsIds`	`List<String>`	a list of 'pin'/'pinned' documents to 'explode'/'extract'
`parentId`	`String`	the id of the folder or process where the 'pin' document will be created

DTO structure on output

as hash-table will contain the <original id>-to-<new id> correspondence

Errors

- InvalidInputBusinessException - invalid `docsIds`
- InvalidInputBusinessException - invalid `parentId`
- DeniedSecurityException - security context does not allow operation for caller

Notes

- the operation is internally audited

```
public static final int CMD_MOVE_PIN = 67127055
```

Move a pin document to a specific location (folder/process).

DTO structure on input

Key	Type	Significance
`objectId`	`String`	the id of the pin object to move
`targetId`	`String`	id of folder or process where to move the pin

DTO structure on output

Key	Type	Significance
`objectId`	`String`	the new id of object (it reflects the new immediate parent at the new location)

Errors

- InvalidInputBusinessException - invalid `objectId` or `targetId`
- DeniedSecurityException - security context does not allow operation for caller

Notes

- the operation is internally audited

```
public static final int CMD_CREATE_NOTE = 67127297
```

Create a note.

DTO structure on input

Key	Type	Significance
`object`	Note	the note object to create
`parentId`	`String`	id of folder or document where to create the note

DTO structure on output

Key	Type	Significance
`object`	Note	the same object supplied on input but with `id`, `dateC` and other initialization-time relevant fields set

Errors

- InvalidInputBusinessException invalid `parentId` or `object.fileId`
- DeniedSecurityException - security context does not allow operation for caller

Notes

- the note content will be indexed if containing database is fulltext enabled
- the content of the note is actually in the file specified by `object.fileId` property
- the operation is internally audited

```
public static final int CMD_READ_NOTE = 67127298
```

Read a note.

DTO structure on input

Key	Type	Significance
`object`	Note	The note object to initialize.
`flags`	`long`	a combination of Flag constants specifying what object fields to initialize

DTO structure on output

Key	Type	Significance
`object`	Note	the same object supplied on input but with the `flags` specified fields set

Errors

- InvalidInputBusinessException - invalid `object.id`
- DeniedSecurityException - security context does not allow operation for caller

Notes

- the operation is internally audited
- the content of the note is actually in the file specified by `object.fileId` property

```
public static final int CMD_UPDATE_NOTE = 67127299
```

Update a note.

DTO structure on input

Key	Type	Significance
`object`	Note	the note object to update
`flags`	`long`	a combination of Flag constants specifying what object fields to update

DTO structure on output

`null`

Errors

- InvalidInputBusinessException - invalid `object.id`
- DeniedSecurityException - security context does not allow operation for caller

Notes

- the note content is re-indexed if containing database is fulltext enabled and content field (`fileId`) is among the update-requested fields (Flag#FILE_ID)
- the operation is internally audited

```
public static final int CMD_DELETE_NOTE = 67127300
```

Delete a note.

DTO structure on input

Key	Type	Significance
`objectId`	`String`	the id of the note object to delete

DTO structure on output

`null`

Errors

- InvalidInputBusinessException - invalid `objectId`
- DeniedSecurityException - security context does not allow operation for caller

Notes

- the note content is un-indexed first if containing database is fulltext enabled
- the operation is internally audited

```
public static final int CMD_ENUMERATE_NOTES = 67127301
```

Enumerate notes.

DTO structure on input

Key	Type	Significance
`criteriaIds`	`List<String>`	a list of criteria to use as filters for enumeration
`flags`	`long`	the fields of the Note object to initialize with server data. A combination of Flag defined constants, relevant for the Note class objects. Also, a list of order options defined by the Flag class can be passed through this parameter
`offset`	`Integer`	offset from which to start enumeration. E.g. if offset is 2, then the enumeration will return elements starting with the third note.
`count`	`Integer`	number of maximum objects to return.

DTO structure on output

Key	Type	Significance
`list`	`List<Note>`	the list of notes matching enumeration criteria.See Notes section for supported criteria combinations

Errors

- InvalidInputBusinessException - invalid `criteria.See Notes for supported criteria combinations`
- DeniedSecurityException - security context does not allow operation for caller

Notes

- the following table lists the supported criteria combinations

Content of **`criteriaIds`**	Significance
[`<objectId>`]	the notes of folder/process/pin/document specified by `objectId` will be enumerated

```
public static final int CMD_CREATE_VERSION = 67127553
```

Create a document version.

DTO structure on input

Key	Type	Significance
`object`	`DocumentVersion`	the version to create, specifying a fileId and optionally a label

documentId	String	the id of the document to whom this version is created
versionType	String	can be 'major' or 'minor' specifying the type of the new version of the document

DTO structure on output

Key	Type	Significance
object	DocumentVersion	the version object received on input as persisted in the database (id, label etc.)
document	Document	the updated document object as persisted in the database (id, label etc. but no attributes)

Errors

- InvalidInputBusinessException - invalid documentId or object.fileId or versionType
- NotFoundPersistenceException - document specified by documentId does not exist
- LockedContentBusinessException - document specified by documentId is locked or checked-out by some other user
- DeniedSecurityException - security context does not allow operation for caller

Notes

- load the document and get current version number.
- the new version number is generated by the application. Requests for major versions will generate a MAJ.0 version number where MAJ is the increment of the current major version. Requests for minor versions will generate a MAJ.MIN version number where MAJ is the current major version and MIN is the increment of the current minor version.
- if object.label is specified then this will be the new version label, otherwise the version will take from the name of the document
- the fileId, fileType, fileSize of the document will be swapped with the versions similar fields so that object supplied fields will be the new document properties.
- document content will be reindexed if database is fulltext enabled and document content is of text type
- the operation is internally audited

```
public static final int CMD_READ_VERSION = 67127554
```

Read a document version object.

DTO structure on input

Key	Type	Significance

`object`	DocumentVersion	the document version object to initialize
`flags`	`long`	a combination of Flag constants specifying what object fields to initialize

DTO structure on output

Key	Type	Significance
`object`	DocumentVersion	the same object supplied on input but with the `flags`-specified fields set

Errors

- InvalidInputBusinessException - invalid `object.id`
- DeniedSecurityException - security context does not allow operation for caller

Notes

- the operation is internally audited

```
public static final int CMD_DELETE_VERSION = 67127556
```

Delete a document version.

DTO structure on input

Key	Type	Significance
`objectId`	`String`	the id of the documentversion object to delete

DTO structure on output

`null`

Errors

- InvalidInputBusinessException - invalid `objectId`
- DeniedSecurityException - security context does not allow operation for caller

Notes

- the fulltext/content database index is updated to reflect the version removal
- the operation is internally audited

```
public static final int CMD_ENUMERATE_VERSIONS = 67127557
```

Enumerate document versions.

DTO structure on input

Key	Type	Significance
`objectId`	`String`	the id of the document whose versions to enumerate
`offset`	`Integer`	offset of the first version to return (optional)
`count`	`Integer`	count of versions to return (optional)
`flags`	`Long`	a combination of flags specifying what fields of the version objects to fill on return

DTO structure on output

Key	Type	Significance
`list`	`List<Identified>`	the of version objects

Errors

- InvalidInputBusinessException - invalid `objectId`
- NotFoundPersistenceException - document specified by `objectId` does not exist
- DeniedSecurityException - security context does not allow operation for caller

Notes

- unless `offset` and `count` specified all document versions are listed

```
public static final int CMD_COUNT_VERSION = 67127565
```

Count documents versions in a container (including archive).

DTO structure on input

Key	Type	Significance
`criteriaIds`	`List<String>`	criteria to use for counting.Currently supported criteria are same as for version enumeration.

DTO structure on output

Key	Type	Significance
`count`	`Long`	the versions count

Errors

- InvalidInputBusinessException - invalid `criteriaIds`

Notes

- this command is usually used when a page-by-page display of a large list of versions is desired

- see GedLibrary#CMD_ENUMERATE_VERSIONS for supported `criteriaIds` combinations

```
public static final int CMD_CHECK_OBJECT = 67128331
```

Check a folder/process for the existence of a child element specified by label, form and type.

DTO structure on input

Key	Type	Significance
`parentId`	`String`	the id of the container to search (folder/process)
`objectLabel`	`String`	the name of the obejct to look for (document/pin/subfolder)
`objectType`	`String`	the type of the obejct to look for (document/pin/subfolder) specified as Prefix of an id
`formId`	`String`	**Optional**. The id of the form of the object to look for.If not specified then the form if will not apply as lookup criterion
`flags`	`Long`	**Optional**. `Flag.SEARCH_CASE_SENSITIVE` if case sensitive lookup required.Default is `0L`
`modeSearch`	`Long`	**Optional**. `0L` for immediate children enumeration lookup, `1L` for sub-tree search

DTO structure on output

Key	Type	Significance
`objectId`	`String`	the id of the first matching object, `null` otherwise.

Errors

- InvalidInputBusinessException - invalid `parentId` or `formId` or they do not indicate same database or parent and children type are not compatible

```
public static final int CMD_LOCK_DOCUMENT = 67126555
```

Lock a document. This command will render the document un-modifiable (regarding content) and un-movable (including un-pinnable) to all users until the document is unlocked or the lock validity expires.

DTO structure on input

Key	Type	Significance
`objectId`	`String`	the id of the document to lock.
`validityMillis`	`long`	**Optional**. The lock validity period expressed in milliseconds.If neither `validityMillis` nor

		`validityDate` then unlimited validity is assumed.
`validityDate`	`java.util.Date`	**Optional**. The lock lock validity period expressed using the date when validity expires. If neither `validityMillis` nor `validityDate` then unlimited validity is assumed.

DTO structure on output

`null`

Errors

- InvalidInputBusinessException - invalid `objectId`
- NotFoundPersistenceException - document specified by `objectId` does not exist
- OutOfDateBusinessException - document specified by `objectId` is already locked or checked-out
- DeniedSecurityException - security context does not allow operation for caller

Notes

- this command will update the related document properties if successful
- the operation is internally audited

```
public static final int CMD_UNLOCK_DOCUMENT = 67126556
```

Unlock a document. This command is the pair of the 'lock' command and is also the equivalent of 'undo checkout'.

DTO structure on input

Key	Type	Significance
`objectId`	`String`	the id of the document to unlock.

DTO structure on output

`null`

Errors

- InvalidInputBusinessException - invalid `objectId`
- NotFoundPersistenceException - document specified by `objectId` does not exist
- OutOfDateBusinessException - document specified by `objectId` is not locked or checked-out by the calling user
- DeniedSecurityException - security context does not allow operation for caller

Notes

- this command will update the related document properties if successful
- the operation is internally audited

```
public static final int CMD_CHECKOUT_DOCUMENT = 67126557
```

Lock a document exclusively for content modifications. This command will render the document un-modifiable (regarding content) to all users other then the calling connected user and un-movable (including un-pinnable) to all users until the document is unlocked or the lock validity expires.

DTO structure on input

Key	Type	Significance
`objectId`	`String`	the id of the document to lock
`validityMillis`	`long`	**Optional**. The lock validity period expressed in milliseconds. If neither `validityMillis` nor `validityDate` then unlimited validity is assumed.
`validityDate`	`java.util.Date`	**Optional**. The lock lock validity period expressed using the date when validity expires. If neither `validityMillis` nor `validityDate` then unlimited validity is assumed.

DTO structure on output

`null`

Errors

- InvalidInputBusinessException - invalid `objectId`
- NotFoundPersistenceException - document specified by `objectId` does not exist
- OutOfDateBusinessException - document specified by `objectId` is already locked or checked-out by another user
- DeniedSecurityException - security context does not allow operation for caller

Notes

- this command will update the related document properties if successful
- the operation is internally audited

```
public static final int CMD_CHECKIN_DOCUMENT = 67126558
```

Check in the modified content of a document.

DTO structure on input

Key	Type	Significance

`objectId`	`String`	the id of the document to unlock.
`fileId`	`String`	**Optional**. The stored file's Id.
`version`	`String`	**Optional**. Specify if the document content is to be versioned through this operation. Can be one of `MAJOR` or `MINOR`.

DTO structure on output

`null`

Validation to perform

- InvalidInputBusinessException - invalid `objectId`
- NotFoundPersistenceException - document specified by `objectId` does not exist
- OutOfDateBusinessException - document specified by `objectId` is already locked or checked-out by another user
- DeniedSecurityException - security context does not allow operation for caller

Notes

- this command will update the related document properties if successful
- the operation is internally audited

```
public static final int CMD_MOVE_ACTIVE_ARCHIVE = 67158031
```

Move a list of objects specified by id from the active part to the archive part of the database.

DTO structure on input

Key	Type	Significance
`list`	`List<String>`	list containing objects id-s

DTO structure on output

`null`

Errors

- InvalidInputBusinessException - invalid `list` or one of the elements in it.The database root id is considered invalid
- DeniedSecurityException - security context does not allow operation for caller

Notes

- container objects (folders, pins) are moved recursively
- the operation is internally audited for each object
- the operation will handle only document database objects.Workflow objects are not supported

```
public static final int CMD_MOVE_ARCHIVE_ACTIVE = 67158287
```

Move a list of objects specified by id from the archive part to the active part of the database.

DTO structure on input

Key	Type	Significance
`list`	`List<String>`	list containing objects id-s

DTO structure on output

`null`

Validation to perform

- InvalidInputBusinessException - invalid `list` or one of the elements in it.The database root id is considered invalid
- DeniedSecurityException - security context does not allow operation for caller

Notes

- container objects (folders, pins) are moved recursively
- the operation is internally audited for each object
- the operation will handle only document database objects.Workflow objects are not supported

```
public static final int CMD_COUNT_DOCUMENT = 67126541
```

Count documents in a container (folder/process/pin including archive supported versions).

DTO structure on input

Key	Type	Significance
`criteriaIds`	`List<String>`	criteria to use for counting.Currently supported criteria are same as for document enumeration.

DTO structure on output

Key	Type	Significance
`count`	`Long`	the documents count

Errors

- InvalidInputBusinessException - invalid `criteriaIds`

Notes

- this command is usually used when a page-by-page display of a large list of documents is desired

- see GedLibrary#CMD_ENUMERATE_DOCUMENTS for supported `criteriaIds` combinations

```
public static final int CMD_CREATE_COMMENT = 67130369
```

Create a comment (on a comment-enabled object or a reply to another comment).

DTO structure on input

Key	Type	Significance
`object`	Comment	the object to create. The `parentId` property must specify either a comment-enabled object (folder/document/process - no pins) or another comment to which this comment is a reply

DTO structure on output

Key	Type	Significance
`object`	Comment	the same object supplied on input but with `id`, `dateC` and other initialization-time relevant fields set

Errors

- InvalidInputBusinessException - invalid `object` (no label, no content or no parent set)
- DeniedSecurityException - security context does not allow operation for caller

Notes

- an email alert to the owner (in case of object comment) or to the replied comment author (in case of reply)
- the operation is internally audited

```
public static final int CMD_UPDATE_COMMENT = 67130371
```

Update a comment.

DTO structure on input

Key	Type	Significance
`object`	Comment	the object to update
`flags`	`Long`	a combination of comment-supported flags specifying which fields to update.

DTO structure on output

Key	Type	Significance

`object`	Comment	the same object supplied on input

Errors

- InvalidInputBusinessException - invalid `object`
- DeniedSecurityException - security context does not allow operation for caller

Notes

- the operation is internally audited

```
public static final int CMD_DELETE_COMMENT = 67130372
```

Delete a comment.

DTO structure on input

Key	Type	Significance
`objectId`	`String`	the id of the comment object to delete

DTO structure on output

`null`

Errors

- InvalidInputBusinessException - invalid `objectId`
- DeniedSecurityException - neither connected user is DBA so that he can change any commend nor the user changes one of his own unreplied comments
- DeniedSecurityException - security context (comment-enabled object security) does not allow operation for caller

Notes

- the operation is internally audited

```
public static final int CMD_ENUMERATE_COMMENTS = 67130373
```

Enumerate comments given a set of criteria.

DTO structure on input

Key	Type	Significance
`criteriaIds`	`List<String>`	a list of criteria to use as filters for enumeration
`flags`	`long`	the fields of the Comment object to initialize with server data. A combination of Flag defined constants, relevant for the Comment class objects.

		Also, a list of order options defined by the Flag class can be passed through this parameter
`offset`	`Integer`	offset from which to start enumeration
`count`	`Integer`	number of maximum objects to return.

DTO structure on output

Key	Type	Significance
`list`	`List<Comment>`	the list of comments matching enumeration criteria

Errors

- InvalidInputBusinessException - invalid `criteriaIds`
- DeniedSecurityException - security context does not allow operation for caller

Notes

- the following table lists the supported criteria combinations

Content of `criteriaIds`	Significance
[`<objectId>`]	the comments of folder/process/document specified by `objectId` will be enumerated

```
public static final int CMD_READ_COMMENT = 67130370
```

Read a comment.

DTO structure on input

Key	Type	Significance
`object`	Comment	the comment object to initialize.
`flags`	`long`	a combination of Flag constants specifying what object fields to initialize

DTO structure on output

Key	Type	Significance
`object`	Comment	the same object supplied on input but with the `flags`-specified fields set

Errors

- InvalidInputBusinessException - invalid `object.id`

- DeniedSecurityException - security context does not allow operation for caller

Notes

- the operation is internally audited

```
public static final int CMD_COUNT_COMMENTS = 67130381
```

Count comments on an object or replies of another comment.

DTO structure on input

Key	Type	Significance
`criteriaIds`	`List<String>`	criteria to use for counting. Currently supported criteria are one comment-enabeld object id or a comment id

DTO structure on output

Key	Type	Significance
`count`	`Long`	the comments count

Validation to perform

- InvalidInputBusinessException - invalid `criteriaIds`

Notes

- the following table lists the supported criteria combinations

Content of `criteriaIds`	Significance
[`<objectId>`]	the comments of folder/process/document specified by `objectId` will be enumerated
[`<commentId>`]	the reply comments of a comment specified by `objectId` will be enumerated

```
public static final int CMD_READ_DOCUMENT_BY_UUID = 67126672
```

Read a document object by uuid.

DTO structure on input

Key	Type	Significance
`object`	Document	the document object to initialize with persisted data

`flags`	`long`	the fields of the Document object to initialize with server data. A combination of Flag defined constants, relevant for the Document class objects.

DTO structure on output

Key	Type	Significance
`object`	Document	the same object supplied on input but with the fields specified by `flags` parameter set.

Errors

- InvalidInputBusinessException - invalid `object.uuid`
- NotFoundPersistenceException - object specified by `object.uuid` does not exist

Notes

Constructor detail :

```
public GedLibrary ()
```

Package com.siatel.framework.service.misc

This package groups miscellaneous (scripting, document conversion, thumbnails etc.) service definitions and utilities.

Classes

MiscLibrary

```
public class MiscLibrary
```

Author :

victorc

Field summary :

Modifier and type	Field
`public static final int`	`CMD_CONVERT_TO_HTML`
`public static final int`	`CMD_CONVERT_TO_PDF`
`public static final`	`CMD_EVALUATE_WATERMARK`

int	
public static final int	CMD_CREATE_THUMB
public static final int	CMD_SEND_MAIL
public static final int	CMD_COUNT_PAGES
public static final int	CMD_CREATE_PDF
public static final int	CMD_STORE_PDF
public static final int	CMD_RETRIEVE_PDF
public static final int	CMD_DELETE_PDF
public static final int	CMD_CHECK_PDF

Constructor summary :

Constructor
MiscLibrary ()

Field detail :

```
public static final int CMD_CONVERT_TO_HTML = 1073741955
```

Convert a file to HTML so that it can be displayed in the browser.

DTO structure on input

Key	Type	Significance
objectId	String	the id of an object which will be used to audit the operation, if specified (this parameter is optional)
fileId	String	the id of a storage-located file to convert.Either fileId or the fileStream-fileName pair must be specified
fileStream	InputStream	the stream of the content of the file to convert.Either fileId or the fileStream-fileName pair must be specified
fileName	String	the name of the file to convert.Either fileId or

		the `fileStream-fileName` pair must be specified

DTO structure on output

Key	Type	Significance
`fileOut`	`String`	the full path of the resulting conversion file as built by the conversion support.This path is on the server and is consequently usable only to server-side entities
`success`	`Boolean`	`true` if HTML conversion was successful

Errors

- InvalidInputBusinessException - neither `fileId` nor the `fileStream-fileName` were specified
- InvalidInputBusinessException - if `fileId` specified then it is subsequent validation does not succeed (e.g. file not found)

Notes

- the `fileStream-fileName` pair take precedence over `fileId` if both specified.
- the INSO HTML conversion engine must be installed and properly configured in order for this to work correctly.

```
public static final int CMD_CONVERT_TO_PDF = 1073741958
```

Convert a file to PDF so that it can be displayed in the browser.

DTO structure on input

Key	Type	Significance
`objectId`	`String`	the id of an object which will be used to audit the operation, if specified (this parameter is optional)
`fileId`	`String`	the id of a storage-located file to convert.Either `fileId` or the `fileStream-fileName` pair must be specified
`fileStream`	`InputStream`	the stream of the content of the file to convert.Either `fileId` or the `fileStream-fileName` pair must be specified
`fileName`	`String`	the name of the file to convert.Either `fileId` or the `fileStream-fileName` pair must be specified

DTO structure on output

Key	Type	Significance
`fileOut`	`String`	the full path of the resulting conversion file as built by the conversion support.This path is on the server and is consequently usable only to server-side entities
`success`	`Boolean`	`true` if PDF conversion was successful

Errors

- InvalidInputBusinessException - neither `fileId` nor the `fileStream-fileName` were specified
- InvalidInputBusinessException - if `fileId` specified then it is subsequent validation does not succeed (e.g. file not found)

Notes

- the `fileStream-fileName` pair take precedence over `fileId` if both specified.
- the INSO PDF conversion engine must be installed and properly configured in order for this to work correctly.

```
public static final int CMD_EVALUATE_WATERMARK = 1073793171
```

Evaluates the watermark needed to print this document.

DTO structure on input

Key	Type	Significance	
`objectId`	`String`	the id of a document to evaluate watermark for	
`object`	`Document`	The object itself; Either `objectId` or the `object`	must be specified; if both are, the `object` takes precedence

DTO structure on output

Key	Type	Significance
`objectId`	`String`	The id of the resource to be used as watermark for the given document

Validations

- InvalidInputBusinessException - if either the given `objectId` or the `object` are not valid

Notes

None.

```
public static final int CMD_CREATE_THUMB = 1073791745
```

Generate a thumbnail so that it can be displayed in the browser in thumbnail-mode.

DTO structure on input

Key	Type	Significance
`fileId`	`String`	the id of a storage-located file to generate thumbnail.Either `fileId` or the `fileStream-fileName` pair must be specified
`fileStream`	`InputStream`	the stream of the content of the file to generate thumbnail.Either `fileId` or the `fileStream-fileName` pair must be specified
`fileName`	`String`	the name of the file to generate thumbnail.Either `fileId` or the `fileStream-fileName` pair must be specified
`pageNumber`	`Integer`	the page of the document file to use as thumbnail source
`sizeX`	`Integer`	width of thumbnail to generate
`sizeY`	`Integer`	height of thumbnail to generate

DTO structure on output

Key	Type	Significance
`fileStream`	`InputStream`	the stream of the generated thumbnail

Validations

- InvalidInputBusinessException - neither `fileId` nor the `fileStream-fileName` were specified
- InvalidInputBusinessException - if `fileId` specified then it is subsequent validation does not succeed (e.g. file not found)

Notes

- the `fileStream-fileName` pair take precedence over `fileId` if both specified.
- the INSO image conversion engine must be installed and properly configured in order for this to work correctly.
- the thumb file is stored alongside the source file using the same file id.

```
public static final int CMD_SEND_MAIL = 1073741956
```

Send documents by mail.

DTO structure on input

Key	Type	Significance
`action`	`String`	"0" means use "`object`" key to get a com.siatel.domain.mail.MailMessage to use. "1" means all the message parameters and attachments are supplied in **this** DTO.
`object`	com.siatel.domain.mail.MailMessage	the e-mail message already wrapped as an instance of this class. Relevant when `action` is "0"
`to`	`String`	the e-mail addresses to use in the "to" field of the e-mail message, separated by " " (blank) "," (comma) or ";" (semicolon)
`cc`	`String`	the e-mail addresses to use in the "cc" field of the e-mail message, separated by " " (blank) "," (comma) or ";" (semicolon)
`subject`	`String`	the subject field of the e-mail message
`message`	`String`	the body of the e-mail message
`readReceipt`	`String`	"1" to request read receipts
`objectList`	`List<String>`	the list of objects (normally documents) to attach to email or link in the email body

DTO structure on output

Key	Type	Significance

Errors

- InvalidInputBusinessException - neither the whole message is supplied in the dto "object" field nor the message fields are specified in the dto
- InvalidInputBusinessException - the email addresses field is not valid

Notes

- the sender of the email is the server, but the identity of the connected user is also visible in "sender" field of the sent message
- security may render inaccessible some of the objects to be sent as attachments. These will be sent as links in the message. Clicking these links will redirect the user to the database location of the linked object

```
public static final int CMD_COUNT_PAGES = 1073741969
```

Counts the number of pages in a document

DTO structure on input

Key	Type	Significance
`objectId`	`String`	the id of a object-to count the pages for

DTO structure on output

Key	Type	Significance
`count`	`String`	The number of pages detected in the document

Errors

- InvalidInputBusinessException - if `objectId` was not specified, is invalid or not available anymore
- IOException - a disk error has occurred, mainly caused by unable to read/write to the file
- PersistenceException - unable to retrieve the file content
- DiskAccessBusinessException - the content of the file could not have been cached

Notes

- there is no guarantee about the access to a page in an interval

```
public static final int CMD_CREATE_PDF = 1073792513
```

Generate a PDF file so that it can be displayed in the browser instead of the original document file.

DTO structure on input

Key	Type	Significance
`fileId`	`String`	the id of a storage-located file to generate the PDF file.

DTO structure on output

Key	Type	Significance
`fileStream`	`InputStream`	the stream of the generated PDF file

Validations

- InvalidInputBusinessException - neither `fileId` nor the `fileStream-fileName` were specified
- InvalidInputBusinessException - if `fileId` specified then its subsequent validation does not succeed (e.g. file not found)

Notes

- the INSO image conversion engine must be installed and properly configured in order for this to work correctly.
- the PDF file is stored alongside the source file using the same file id but differs by extension.

```
public static final int CMD_STORE_PDF = 1073792537
```

Store the PDF file version of a document file.

DTO structure on input

Key	Type	Significance
`fileId`	`String`	the id of the document file
`fileStream`	`InputStream`	the stream of PDF file

DTO structure on output

Key	Type	Significance
`fileId`	`String`	the PDF file id

Errors

- InvalidInputBusinessException - invalid `fileId`

Notes

- the id of the PDF file differs from the document file id only by extension
- only one PDF file can be stored for a given file

```
public static final int CMD_RETRIEVE_PDF = 1073792514
```

Retrieve the PDF file associated with a document file.

DTO structure on input

Key	Type	Significance
`fileId`	`String`	The id of the document file or of the PDF file itself

DTO structure on output

Key	Type	Significance
`fileStream`	`InputStream`	the PDF file stream

Errors

- InvalidInputBusinessException - invalid `fileId`

Notes

None.

```
public static final int CMD_DELETE_PDF = 1073792516
```

Delete a previously stored PDF file.

DTO structure on input

Key	Type	Significance
`fileId`	`String`	The id of the document file or of the PDF file itself

DTO structure on output

empty, `null` will be returned

Errors

- InvalidInputBusinessException - invalid `fileId`

Notes

None.

```
public static final int CMD_CHECK_PDF = 1073792523
```

Check if there is a PDF file associated with a previously stored file.

DTO structure on input

Key	Type	Significance
`fileId`	`String`	Id of document file to check for associated PDF

DTO structure on output

Key	Type	Significance
`hasPdf`	`boolean`	`true` if the PDF file exists, `false` otherwise

Errors

- InvalidInputBusinessException - invalid `fileId`

Notes

Constructor detail :

```
public MiscLibrary ()
```

Package com.siatel.framework.service.plugins

Classes

PluginsLibrary

```
public class PluginsLibrary
```

Library of command id's for the 'plugins' service access point.

Since :

7.4

Author :

victorc

Field summary :

Modifier and type	Field
`public static final int`	`CMD_INVOKE_PLUGIN`

Constructor summary :

Constructor
`PluginsLibrary ()`

Field detail :

```
public static final int CMD_INVOKE_PLUGIN = 184614913
```

Invoke a plugin functionality.

DTO structure on input (to Command#invoke)

Key	Type	Significance
`pluginId`	`String`	the id of the plugin to invoke. This is a free form string and should match the id of the plugin to invoke.
`cmdId`	`String`	Ithe id of the command to invoke. This is a free form string and should match an id of a command supported by the invoked plugin.

DTO structure on output (from Command#invoke)

Key	Type	Significance
`modelAndView`	`org.springframework.web.servlet.ModelAndView`	**Optional**. The UI

		response to be sent by application server to browser if invoked command delivers such response

Errors

- PluginInvocationException - invalid `pluginId`

Notes

- the input DTO can contain additional command parameters which are specific to the command and plugin.
- the output DTO can contain additional results which are specific to the command and plugin, which are relevant to the caller.

Constructor detail :

```
public PluginsLibrary ()
```

Package com.siatel.framework.service.search

This package groups object search related service definitions and utilities.

Classes

SearchLibrary

```
public class SearchLibrary
```

Author :

victorc

Field summary :

Modifier and type	Field
`public static final int`	`CMD_SEARCH_SIMPLE`
`public static final int`	`CMD_SEARCH_ADVANCED`

public static final int	CMD_SEARCH_FULLTEXT
public static final int	CMD_SEARCH_MORE
public static final int	CMD_SEARCH_BACK
public static final int	CMD_CLOSE_SEARCH
public static final int	CMD_SEARCH_ENUMERATE
public static final int	CMD_SEARCH_REMOVE_OBJECTS
public static final int	CMD_SEARCH_IS
public static final int	CMD_SORT_RESULTS
public static final int	CMD_BASIC_SORT_RESULTS
public static final int	CMD_PERSIST_RESULTSET
public static final int	CMD_CLOSE_RESULTSET
public static final int	CMD_PURGE_RESULTSETS
public static final int	CMD_SEARCHTEMPLATE_READ
public static final int	CMD_SEARCHTEMPLATE_LISTALL
public static final int	CMD_SEARCHTEMPLATE_SAVE
public static final int	CMD_SEARCHTEMPLATE_DELETE
public static final int	CMD_SEARCHTEMPLATE_UPDATE
public static final int	CMD_ENUMERATE_USER_SEARCHTEMPLATE
public static final	CMD_ENUMERATE_GROUP_SEARCHTEMPLATE

int	
public static final int	CMD_ENUMERATE_ROLE_SEARCHTEMPLATE

Constructor summary :

Constructor
SearchLibrary ()

Field detail :

```
public static final int CMD_SEARCH_SIMPLE = 134217729
```

This class handles the SearchLibrary.CMD_SEARCH_SIMPLE command.

DTO structure on input

Key	Type	Significance
expression	String	the expression to search for
date	Date	the date to search for in case date fields or date attributes should be searched
number	Number	the number to search for in case numeric attributes should be searched
databaseIds	List<String>	the databases to search for
flags	long	a group of flags for the search operation
count	int	the number of search results to retrieve on this call
datec_min	Date	additional search criteria: the minimal creation date (optional)
datem_min	Date	additional search criteria: the minimal modification date (optional)
datec_max	Date	additional search criteria: the maximal creation date (optional)
datem_max	Date	additional search criteria: the maximal modification date (optional)

DTO structure on output

Key	Type	Significance
results	List<SearchResult>	the list of results
context	String	a key identifying the search context to be used for SearchLibrary.CMD_SEARCH_MORE and SearchLibrary.CMD_CLOSE_SEARCH

statistics	Map<String , Integer>	Statistics of results

```
public static final int CMD_SEARCH_ADVANCED = 134217730
```

This class handles the SearchLibrary.CMD_SEARCH_ADVANCED command.

DTO structure on input

Key	Type	Significance
expression	Expression	the expression to search for
flags	long	a group of flags for the search operation
count	int	the number of search results to retrieve on this call

DTO structure on output

Key	Type	Significance
results	List<SearchResult>	the list of results
context	String	a key identifying the search context to be used for SearchLibrary.CMD_SEARCH_MORE and SearchLibrary.CMD_CLOSE_SEARCH
statistics	Map<String , Integer>	Statistics of results

```
public static final int CMD_SEARCH_FULLTEXT = 134217731
```

This class handles the SearchLibrary.CMD_SEARCH_FULLTEXT command.

DTO structure on input

Key	Type	Significance
expression	String	the expression to search for
databaseIds	List<String>	the databases to search for
flags	long	a group of flags for the search operation
count	int	the number of search results to retrieve on this call
datec_min	Date	additional search criteria: minimal creation date
datem_min	Date	additional search criteria: minimal modification date
datec_max	Date	additional search criteria: maximal creation date
datem_max	Date	additional search criteria: maximal modification date

DTO structure on output

Key	Type	Significance
results	List<SearchResult>	the list of results
context	String	a key identifying the search context to be used for SearchLibrary.CMD_SEARCH_MORE and SearchLibrary.CMD_CLOSE_SEARCH
statistics	Map<String , Integer>	Statistics of results
statsByDatabase	Map<String , Map<String , Integer>>	Statistics of results, by database Id
statsByDatabase	Map<String , Map<String , Integer>>	Statistics of results, by database Id

```
public static final int CMD_SEARCH_MORE = 134217732
```

This class handles the SearchLibrary.CMD_SEARCH_MORE command.

DTO structure on input

Key	Type	Significance
context	String	a key identifying the search context, returned by a previous SearchLibrary.CMD_SEARCH_SIMPLE, SearchLibrary.CMD_SEARCH_ADVANCED or SearchLibrary.CMD_SEARCH_FULLTEXT command
count	int	the number of search results to retrieve on this call

DTO structure on output

Key	Type	Significance
results	List<SearchResult>	the list of results

```
public static final int CMD_SEARCH_BACK = 134217735
```

This class handles the SearchLibrary.CMD_SEARCH_BACK command.

DTO structure on input

Key	Type	Significance
context	String	a key identifying the search context, returned by a previous SearchLibrary.CMD_SEARCH_SIMPLE, SearchLibrary.CMD_SEARCH_ADVANCED or SearchLibrary.CMD_SEARCH_FULLTEXT command

count	int	the number of search results to retrieve on this call

DTO structure on output

Key	Type	Significance
results	List<SearchResult>	the list of results

```
public static final int CMD_CLOSE_SEARCH = 134217733
```

This class handles the SearchLibrary.CMD_CLOSE_SEARCH command.

DTO structure on input

Key	Type	Significance
context	String	a key identifying the search context, returned by a previous SearchLibrary.CMD_SEARCH_SIMPLE, SearchLibrary.CMD_SEARCH_ADVANCED or SearchLibrary.CMD_SEARCH_FULLTEXT command

DTO structure on output

null

```
public static final int CMD_SEARCH_ENUMERATE = 134217736
```

This class handles the SearchLibrary.CMD_SEARCH_ENUMERATE command.

DTO structure on input

Key	Type	Significance
context	String	a key identifying the search context, returned by a previous SearchLibrary.CMD_SEARCH_SIMPLE, SearchLibrary.CMD_SEARCH_ADVANCED or SearchLibrary.CMD_SEARCH_FULLTEXT command
offset	int	the offset of search results to retrieve on this call
count	int	the number of search results to retrieve on this call

DTO structure on output

Key	Type	Significance
results	List<SearchResult>	the list of results

```
public static final int CMD_SEARCH_REMOVE_OBJECTS = 134217737
```

This class handles the SearchLibrary.CMD_SEARCH_REMOVE_OBJECTS command.

- It removes objects from the result cache.

- It does not delete the objects from the persistence.
- It refreshes the statistics.

DTO structure on input

Key	Type	Significance
`context`	`String`	a key identifying the search context, returned by a previous SearchLibrary.CMD_SEARCH_SIMPLE, SearchLibrary.CMD_SEARCH_ADVANCED or SearchLibrary.CMD_SEARCH_FULLTEXT command
`idList`	`List<String>`	the list of object IDs to delete

DTO structure on output

Key	Type	Significance
`context`	String	a key identifying the updated search context to be used for `SearchLibrary.CMD_SEARCH_ENUMERATE`
`statistics`	`Map<String , Integer>`	Statistics of results (updated)

```
public static final int CMD_SEARCH_IS = 134217734
```

This class handles the `SearchLibrary.CMD_SEARCH_IS` command.

DTO structure on input

Key	Type	Significance
`attributes`	`DTO` or `String`-encoded DTO	the map of attribute-value entries to search for
`flags`	`long`	a group of flags for the search operation
`count`	`int`	the number of search results to retrieve on this call

DTO structure on output

Key	Type	Significance
`results`	`List<SearchResult>`	the list of results
`context`	String	a key identifying the search context to be used for `SearchLibrary.CMD_SEARCH_MORE` and `SearchLibrary.CMD_CLOSE_SEARCH`
`statistics`	`Map<String , Integer>`	Statistics of results

```
public static final int CMD_SORT_RESULTS = 134217738
```

This class handles the `SearchLibrary.CMD_SORT_RESULTS` command.

DTO structure on input

Key	Type	Significance
`context`	`SearchContext`	the search context returned by a previous SearchLibrary.CMD_SEARCH_ADVANCED command
`attribute`	`String`	the attribute id for the sorting
`order`	`int`	0: ASCENDING; 1: DESCENDING

DTO structure on output

Key	Type	Significance
`context`	`SearchContext`	the search context having the result list sorted

```
public static final int CMD_BASIC_SORT_RESULTS = 134217739
```

This class handles the `SearchLibrary.CMD_BASIC_SORT_RESULTS` command.

It sorts a list of search results, from various databases. Only system attributes can be used: DATEC, DATEM, LABEL, VERSION, FILE_SIZE

DTO structure on input

Key	Type	Significance
`context`	`SearchContext`	the search context returned by a previous SearchLibrary.CMD_SEARCH_SIMPLE command
`attribute`	`Integer`	the attribute id for the sorting
`order`	`int`	0: ASCENDING; 1: DESCENDING

DTO structure on output

Key	Type	Significance
`context`	`SearchContext`	the search context having the result list sorted

```
public static final int CMD_PERSIST_RESULTSET = 134217760
```

This class handles the `SearchLibrary.CMD_PERSIST_RESULTSET` command.

DTO structure on input

Key	Type	Significance
`dbId`	`String`	The database where the results will be persisted
`searchResults`	`List<SearchResult>`	The result list to be persisted

DTO structure on output

Key	Type	Significance
lastSearchResults	SearchResultSetContext	the search result set context

```
public static final int CMD_CLOSE_RESULTSET = 134217761
```

This class handles the SearchLibrary.CMD_CLOSE_RESULTSET command.

DTO structure on input

Key	Type	Significance
searchResultSetContext	String	a key identifying the search result set context, returned by a previous SearchLibrary.CMD_SEARCH_ADVANCED command

DTO structure on output

null

```
public static final int CMD_PURGE_RESULTSETS = 134217762
```

This class handles the SearchLibrary.CMD_PURGE_RESULTSETS command.

DTO structure on input

dbId	String	The database where the results will be persisted
dateMax	Date	All entries older than this date will be deleted

DTO structure on output

null

```
public static final int CMD_SEARCHTEMPLATE_READ = 134217744
```

This class handles the SearchLibrary.CMD_SEARCHTEMPLATE_READ command.

DTO structure on input

Key	Type	Significance
object	SearchTemplate	the SearchTemplate to be read
flags	Long	the flags

DTO structure on output

Key	Type	Significance
object	SearchTemplate	the SearchTemplate read

```
public static final int CMD_SEARCHTEMPLATE_LISTALL = 134217745
```

This class handles the SearchLibrary.CMD_SEARCHTEMPLATE_LISTALL command.

DTO structure on input

Key	Type	Significance
criteriaIds	List<String>	ID of database
flags	Long	the flags
userId	String	the user ID (optional, default: current user)

DTO structure on output

Key	Type	Significance
results	List<SearchTemplate>	the SearchTemplate lists

```
public static final int CMD_SEARCHTEMPLATE_SAVE = 134217746
```

This class handles the SearchLibrary.CMD_SEARCHTEMPLATE_CREATE command.

DTO structure on input

Key	Type	Significance
object	SearchTemplate	the SearchTemplate to be saved
*		
databaseId	String	the id of the database where the searchtemplate will be stored

DTO structure on output

Key	Type	Significance

```
public static final int CMD_SEARCHTEMPLATE_DELETE = 134217747
```

This class handles the SearchLibrary.CMD_SEARCHTEMPLATE_DELETE command.

DTO structure on input

Key	Type	Significance
objectId	String	the id of the SearchTemplate to be deleted

DTO structure on output

Key	Type	Significance

```
public static final int CMD_SEARCHTEMPLATE_UPDATE = 134217748
```

This class handles the SearchLibrary.CMD_SEARCHTEMPLATE_UPDATE command.

DTO structure on input

Key	Type	Significance

object	SearchTemplate	the SearchTemplate to be updated
flags	Long	the flags

DTO structure on output

Key	Type	Significance
object	SearchTemplate	the modified SearchTemplate

```
public static final int CMD_ENUMERATE_USER_SEARCHTEMPLATE = 134242568
```

```
public static final int CMD_ENUMERATE_GROUP_SEARCHTEMPLATE = 134242824
```

```
public static final int CMD_ENUMERATE_ROLE_SEARCHTEMPLATE = 134243080
```

Constructor detail :

```
public SearchLibrary ()
```

Package com.siatel.framework.service.statistics

This package groups statistics related service definitions and utilities.

Classes

StatisticsLibrary

```
public class StatisticsLibrary
```

Library of command id's for the 'workflow' service access point.

Author :

victorc

Constructor summary :

Constructor
StatisticsLibrary ()

Constructor detail :

```
public StatisticsLibrary ()
```

Package com.siatel.proxy.http

This class provides client-side server access support; it is used by the administration client application, but it can be used by third party applications to access the Gargantua server through http protocol

Classes

SiatelClient

```
public class SiatelClient
```

This class provides client-side server access support. It is used by the administration client application, but it can be used by third party applications to access the Gargantua server through http protocol.

It is the responsability of this class to cache a connection token when a login is performed and release it when logout is performed or connection to server is lost.

Internally it dispatches to a service-specific proxy class that invokes a corresponding service servlet on the server.

Constructor summary :

Constructor
SiatelClient (java.lang.String serverUrl)
SiatelClient (java.lang.String serverUrl, boolean trustAny)

Method summary :

Modifier and type	Method
public java.lang.String	getServerUrl ()
public boolean	isConnected ()
public java.lang.String	apiVersion ()
public java.lang.String	serverVersion ()
public boolean	isCompatible ()
public void	login (java.lang.String user, java.lang.String pass, java.lang.String context)
public void	logout ()
public java.util.Map	getConnectionData (java.lang.String fields)
public void	ping ()
public void	createUser (com.siatel.domain.User user)

public void	createUser (com.siatel.domain.User user, java.lang.Boolean expired)
public void	readUser (com.siatel.domain.User user, long flags)
public com.siatel.domain.User	readUser (java.lang.String userId, long flags)
public void	updateUser (com.siatel.domain.User user, long flags)
public void	updateUserPass (java.lang.String userId, java.lang.String pass)
public void	updateUserPass (java.lang.String userId, java.lang.String pass, java.lang.Boolean expired)
public void	deleteUser (java.lang.String userId)
public void	enumUsers (java.util.List criteriaIds, long flags, java.lang.Integer offset, java.lang.Integer count, java.util.List users)
public void	createGroup (com.siatel.domain.Group group)
public void	readGroup (com.siatel.domain.Group group, long flags)
public com.siatel.domain.Group	readGroup (java.lang.String groupId, long flags)
public void	updateGroup (com.siatel.domain.Group group, long flags)
public void	deleteGroup (java.lang.String groupId)
public void	enumGroups (java.util.List criteriaIds, long flags, java.lang.Integer offset, java.lang.Integer count, java.util.List groups)
public void	readDatabase (com.siatel.domain.Database database, long flags)
public com.siatel.domain.Database	readDatabase (java.lang.String databaseId, long flags)
public void	enumDatabasesByConnection (long flags, java.util.List databases)
public void	enumDatabases (java.util.List criteriaIds,

	long flags, java.lang.Integer offset, java.lang.Integer count, java.util.List databases)
public void	createRole (com.siatel.domain.Role role)
public void	readRole (com.siatel.domain.Role role, long flags)
public com.siatel.domain.Role	readRole (java.lang.String roleId, long flags)
public void	updateRole (com.siatel.domain.Role role, long flags)
public void	deleteRole (java.lang.String roleId)
public void	enumRoles (java.util.List criteriaIds, long flags, java.lang.Integer offset, java.lang.Integer count, java.util.List roles)
public void	readStorage (com.siatel.domain.Storage storage, long flags)
public com.siatel.domain.Storage	readStorage (java.lang.String storageId, long flags)
public void	updateStorage (com.siatel.domain.Storage storage, long flags)
public void	enumStorages (java.util.List criteriaIds, long flags, java.lang.Integer offset, java.lang.Integer count, java.util.List storages)
public void	readPortal (com.siatel.domain.Portal portal, long flags)
public com.siatel.domain.Portal	readPortal (java.lang.String portalId, long flags)
public void	enumPortals (java.util.List criteriaIds, long flags, java.lang.Integer offset, java.lang.Integer count, java.util.List portals)
public void	createResource (com.siatel.domain.Resource resource)
public void	readResource (com.siatel.domain.Resource resource, long flags)
public	readResource (java.lang.String resourceId,

com.siatel.domain.Resource	long flags)
public void	updateResource (com.siatel.domain.Resource resource, long flags)
public void	deleteResource (java.lang.String resourceId)
public void	enumResources (java.util.List criteriaIds, long flags, java.lang.Integer offset, java.lang.Integer count, java.util.List resources)
public com.siatel.domain.Assignment	createGroupUserAssig (java.lang.String objGroupId, java.lang.String objUserId)
public com.siatel.domain.Assignment	readGroupUserAssig (java.lang.String objGroupId, java.lang.String objUserId)
public com.siatel.domain.Assignment	readGroupUserAssig (java.lang.String assigId)
public void	deleteGroupUserAssig (java.lang.String objGroupId, java.lang.String objUserId)
public void	deleteGroupUserAssig (java.lang.String assigId)
public void	enumGroupUserAssig (java.util.List criteriaIds, java.lang.Integer offset, java.lang.Integer count, java.util.List assigs)
public com.siatel.domain.Assignment	createGroupPortalAssig (java.lang.String objGroupId, java.lang.String objPortalId)
public com.siatel.domain.Assignment	readGroupPortalAssig (java.lang.String objGroupId, java.lang.String objPortalId)
public com.siatel.domain.Assignment	readGroupPortalAssig (java.lang.String assigId)
public void	deleteGroupPortalAssig (java.lang.String objGroupId, java.lang.String objPortalId)
public void	deleteGroupPortalAssig (java.lang.String assigId)
public void	enumGroupPortalAssig (java.util.List criteriaIds, java.lang.Integer offset, java.lang.Integer count, java.util.List assigs)
public com.siatel.domain.Assignment	createUserDatabaseAssig (java.lang.String objUserId, java.lang.String objDatabaseId)

public com.siatel.domain.Assignment	readUserDatabaseAssig (java.lang.String objUserId, java.lang.String objDatabaseId)
public com.siatel.domain.Assignment	readUserDatabaseAssig (java.lang.String assigId)
public void	deleteUserDatabaseAssig (java.lang.String objUserId, java.lang.String objDatabaseId)
public void	deleteUserDatabaseAssig (java.lang.String assigId)
public void	enumUserDatabaseAssig (java.util.List criteriaIds, java.lang.Integer offset, java.lang.Integer count, java.util.List assigs)
public com.siatel.domain.Assignment	createUserPortalAssig (java.lang.String objUserId, java.lang.String objPortalId)
public com.siatel.domain.Assignment	readUserPortalAssig (java.lang.String objUserId, java.lang.String objPortalId)
public com.siatel.domain.Assignment	readUserPortalAssig (java.lang.String assigId)
public void	deleteUserPortalAssig (java.lang.String objUserId, java.lang.String objPortalId)
public void	deleteUserPortalAssig (java.lang.String assigId)
public void	enumUserPortalAssig (java.util.List criteriaIds, java.lang.Integer offset, java.lang.Integer count, java.util.List assigs)
public com.siatel.domain.Assignment	createGroupDatabaseAssig (java.lang.String objGroupId, java.lang.String objDatabaseId)
public com.siatel.domain.Assignment	readGroupDatabaseAssig (java.lang.String objGroupId, java.lang.String objDatabaseId)
public com.siatel.domain.Assignment	readGroupDatabaseAssig (java.lang.String assigId)
public void	deleteGroupDatabaseAssig (java.lang.String objGroupId, java.lang.String objDatabaseId)
public void	deleteGroupDatabaseAssig (java.lang.String assigId)
public void	enumGroupDatabaseAssig (java.util.List criteriaIds, java.lang.Integer offset,

	java.lang.Integer count, java.util.List assigs)
public com.siatel.domain.Assignment	createRoleDatabaseAssig (java.lang.String objRoleId, java.lang.String objDatabaseId)
public com.siatel.domain.Assignment	readRoleDatabaseAssig (java.lang.String objRoleId, java.lang.String objDatabaseId)
public com.siatel.domain.Assignment	readRoleDatabaseAssig (java.lang.String assigId)
public void	deleteRoleDatabaseAssig (java.lang.String objRoleId, java.lang.String objDatabaseId)
public void	deleteRoleDatabaseAssig (java.lang.String assigId)
public void	enumRoleDatabaseAssig (java.util.List criteriaIds, java.lang.Integer offset, java.lang.Integer count, java.util.List assigs)
public com.siatel.domain.Assignment	createRolePortalAssig (java.lang.String objRoleId, java.lang.String objPortalId)
public com.siatel.domain.Assignment	readRolePortalAssig (java.lang.String objRoleId, java.lang.String objPortalId)
public com.siatel.domain.Assignment	readRolePortalAssig (java.lang.String assigId)
public void	deleteRolePortalAssig (java.lang.String objRoleId, java.lang.String objPortalId)
public void	deleteRolePortalAssig (java.lang.String assigId)
public void	enumRolePortalAssig (java.util.List criteriaIds, java.lang.Integer offset, java.lang.Integer count, java.util.List assigs)
public com.siatel.domain.Assignment	createUserRoleAssig (java.lang.String objUserId, java.lang.String objRoleId)
public com.siatel.domain.Assignment	readUserRoleAssig (java.lang.String objUserId, java.lang.String objRoleId)
public com.siatel.domain.Assignment	readUserRoleAssig (java.lang.String assigId)
public void	deleteUserRoleAssig (java.lang.String objUserId, java.lang.String objRoleId)

public void	deleteUserRoleAssig (java.lang.String assigId)
public void	enumUserRoleAssig (java.util.List criteriaIds, java.lang.Integer offset, java.lang.Integer count, java.util.List assigs)
public com.siatel.domain.Assignment	createGroupRoleAssig (java.lang.String objGroupId, java.lang.String objRoleId)
public com.siatel.domain.Assignment	readGroupRoleAssig (java.lang.String objGroupId, java.lang.String objRoleId)
public com.siatel.domain.Assignment	readGroupRoleAssig (java.lang.String assigId)
public void	deleteGroupRoleAssig (java.lang.String objGroupId, java.lang.String objRoleId)
public void	deleteGroupRoleAssig (java.lang.String assigId)
public void	enumGroupRoleAssig (java.util.List criteriaIds, java.lang.Integer offset, java.lang.Integer count, java.util.List assigs)
public void	enumAttributes (java.util.List criteriaIds, long flags, java.lang.Integer offset, java.lang.Integer count, java.util.List attributes)
public void	enumForms (java.util.List criteriaIds, long flags, java.lang.Integer offset, java.lang.Integer count, java.util.List forms)
public void	enumViews (java.util.List criteriaIds, long flags, java.lang.Integer offset, java.lang.Integer count, java.util.List views)
public void	createFolder (com.siatel.domain.Folder folder, java.lang.String parentId)
public void	readFolder (com.siatel.domain.Folder folder, long flags)
public com.siatel.domain.Folder	readFolder (java.lang.String folderId, long flags)

public void	updateFolder (com.siatel.domain.Folder folder, long flags)
public void	deleteFolder (java.lang.String folderId)
public boolean	checkTrash (java.lang.String databaseId)
public void	enumFolders (java.util.List criteriaIds, long flags, java.lang.Integer offset, java.lang.Integer count, java.util.List folders)
public void	moveFolder (java.lang.String objectId, java.lang.String targetId)
public void	createDocument (com.siatel.domain.Document document, java.lang.String parentId)
public void	createDocument (com.siatel.domain.Document document, java.lang.String parentId, com.siatel.dto.DTO extra)
public void	readDocument (com.siatel.domain.Document document, long flags)
public com.siatel.domain.Document	readDocument (java.lang.String documentId, long flags)
public void	updateDocument (com.siatel.domain.Document document, long flags)
public void	updateDocument (com.siatel.domain.Document document, long flags, com.siatel.dto.DTO extra)
public void	deleteDocument (java.lang.String documentId)
public void	enumDocuments (java.util.List criteriaIds, long flags, java.lang.Integer offset, java.lang.Integer count, java.util.List documents)
public java.lang.String	moveDocument (java.lang.String documentId, java.lang.String newParentId, java.lang.StringBuilder newFicheId)
public void	addDocumentInFulltext (java.lang.String documentId)
public java.lang.String	removeDocumentFromFulltext (java.lang.String documentId)
public java.lang.String	addNoteInFulltext (java.lang.String noteId)

public java.lang.String	removeNoteFromFulltext (java.lang.String noteId)
public void	pinDocuments (com.siatel.domain.Document pin, java.util.List docsIds, java.lang.String parentId)
public void	unpinDocuments (java.util.List docsIds, java.util.Hashtable resultsIds)
public com.siatel.dto.DTO	createNote (com.siatel.domain.Note note, java.lang.String parentId)
public void	readNote (com.siatel.domain.Note note, long flags)
public com.siatel.domain.Note	readNote (java.lang.String noteId, long flags)
public void	updateNote (com.siatel.domain.Note note, long flags)
public void	deleteNote (java.lang.String noteId)
public void	enumNotes (java.util.List criteriaIds, long flags, java.lang.Integer offset, java.lang.Integer count, java.util.List notes)
public java.lang.String	storeFile (java.lang.String databaseId, java.lang.String path)
public java.lang.String	storeSystemFile (java.lang.String databaseId, java.lang.String path)
public java.io.InputStream	retrieveFile (java.lang.String fileId, java.lang.String objectId)
public void	deleteFile (java.lang.String fileId)
public void	replaceFile (java.lang.String fileId, java.lang.String filePath)
public java.lang.String	storeThumbFile (java.lang.String fileId, java.lang.String thumbPath)
public java.io.InputStream	retrieveThumbFile (java.lang.String fileId)
public void	deleteThumbFile (java.lang.String fileId)
public boolean	checkThumbFile (java.lang.String fileId)
public boolean	checkFile (java.lang.String fileId)
public void	enumAuditEntries (java.util.Date startDate, java.util.Date endDate, java.util.List

	criteriaIds, long flags, java.lang.Integer offset, java.lang.Integer count, java.util.List entries)
public int	countEntries (java.util.Date startDate, java.util.Date endDate, java.util.List criteriaIds, long flags)
public void	createDocumentVersion (com.siatel.domain.DocumentVersion version, java.lang.String documentId, java.lang.String versionType)
public com.siatel.dto.DTO	deleteUsersWithCriteria (com.siatel.dto.DTO dtoIn)

Constructor detail :

```
public SiatelClient (String serverUrl)
```

Constructor.

Parameters :

serverUrl - the url of the server to which this client will connect (e.g. 'http://www.mydomain.com:8080/siatel-web/')

```
public SiatelClient (String serverUrl, boolean trustAny)
```

Method detail :

```
public String getServerUrl ()
```

Accessor.

Returns :

the constructor-bound server url

```
public boolean isConnected ()
```

Accessor to get the connection status.

Returns :

true if the client has a connection token internally set by a #login command

```
public String apiVersion ()
```

Get the current API version.

Returns :

the API version

```
public String serverVersion ()
```

Get the current server version.

Returns :

the server version

```
public boolean isCompatible ()
```

Check API with server compatibility.

Returns :

`true` if the `<major>.<minor>` part of the API version is same with server version

```
public void login (String user, String pass, String context)
```

Login. A connection token will be cached to authenticate subsequent calls through the calling instance.

Parameters :

`user` - the user name

`pass` - the user password

`context` - an application context that will distinguish the use of this session. Can be left blank.

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AuthenticationLibrary#CMD_LOGIN command specification

```
public void logout ()
```

Logout. The cached connection token is released.

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AuthenticationLibrary#CMD_LOGOUT command specification

```
public Map getConnectionData (String fields)
```

Request connection data.

Parameters :

`fields` - comma separated list of requested fields

Returns :

a map of requested field - value

Throws :

`SiatelException` -

```
public void ping ()
```

Keep the connection alive.

Throws :

`SiatelException` - in case connection is no longer available

```
public void createUser (User user)
```

Create a user.

Parameters :

`user` - the User object to create

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_CREATE_USER command specification

```
public void createUser (User user, Boolean expired)
```

Create a user.

Parameters :

`user` - the User object to create

`expired` - indicates that the user is to be created with the password expired or not

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_CREATE_USER command specification

```
public void readUser (User user, long flags)
```

Initialize a User object with data from the server. ('lazy load')

Parameters :

`user` - the User object to initialize with persisted data.

`flags` - the fields of the User object to initialize with server data. A combination of Flag defined constants, relevant for the User class objects

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal

connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_READ_USER command specification

```
public User readUser (String userId, long flags)
```

Initialize a User object with data from the server based on the user id. ('lazy load')

Parameters :

`userId` - the id of the user whose data is requested.

`flags` - the fields of the User object to initialize with server data. A combination of Flag defined constants, relevant for the User class objects

Returns :

an initialized {@link User} object

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_READ_USER command specification

```
public void updateUser (User user, long flags)
```

Update user data.

Parameters :

`user` - the User object containing the data to update

`flags` - the fields from the User object with to update. A combination of Flag defined constants, relevant for the User class objects

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_UPDATE_USER command specification

```
public void updateUserPass (String userId, String pass)
```

Change a user password.

Parameters :

`userId` - the id of the user whose passowrd to update

`pass` - the new password

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the

AdministrationLibrary#CMD_UPDATE_USER_PASS command specification

```
public void updateUserPass (String userId, String pass, Boolean expired)
```

Change a user password.

Parameters :

`userId` - the id of the user whose passowrd to update

`pass` - the new password

`expired` - indicates that the user is to be created with the password expired or not

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_UPDATE_USER_PASS command specification

```
public void deleteUser (String userId)
```

Delete a user.

Parameters :

`userId` - id of user to delete

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_DELETE_USER command specification

```
public void enumUsers (List criteriaIds, long flags, Integer offset,
Integer count, List users)
```

Enumerate users.

Parameters :

`criteriaIds` - a list of object ids to be used as enumeration criteria (e.g. a group id criterion will return the users that are members of the group specified as criterion)

`flags` - the fields of the User returned objects to initialize and the order criterion to use for the enumeration

`offset` - starting position for the enumeration. '0' is assumed is none specified

`count` - the maximum count of User objects to return. 'All' is assumed if none specified

`users` - the list to fill with User objects results of the enumeration

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_ENUMERATE_USERS command specification

```
public void createGroup (Group group)
```

Create a group.

Parameters :

`group` - the Group object to create

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_CREATE_GROUP command specification

```
public void readGroup (Group group, long flags)
```

Initialize a Group object with data from the server. ('lazy load')

Parameters :

`group` - the Group object to initialize with persisted data.

`flags` - the fields of the Group object to initialize with server data. A combination of Flag defined constants, relevant for the Group class objects

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_READ_GROUP command specification

```
public Group readGroup (String groupId, long flags)
```

Initialize a Group object with data from the server based on the group id. ('lazy load')

Parameters :

`groupId` - the id of the group whose data is requested.

`flags` - the fields of the Group object to initialize with server data. A combination of Flag defined constants, relevant for the Group class objects

Returns :

an initialized {@link Group} object

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal

connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_READ_GROUP command specification

```
public void updateGroup (Group group, long flags)
```

Update group data.

Parameters :

`group` - the Group object containing the data to update

`flags` - the fields from the Group object with to update. A combination of Flag defined constants, relevant for the Group class objects

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_UPDATE_GROUP command specification

```
public void deleteGroup (String groupId)
```

Delete a group.

Parameters :

`groupId` - id of group to delete

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_DELETE_GROUP command specification

```
public void enumGroups (List criteriaIds, long flags, Integer offset,
Integer count, List groups)
```

Enumerate groups.

Parameters :

`criteriaIds` - a list of object ids to be used as enumeration criteria (e.g. a database id criterion will return the groups that are assigned to the database specified as criterion)

`flags` - the fields of the Group returned objects to initialize and the order criterion to use for the enumeration

`offset` - starting position for the enumeration. '0' is assumed is none specified

`count` - the maximum count of Group objects to return. 'All' is assumed if none specified

`groups` - the list to fill with Group objects results of the enumeration

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_ENUMERATE_GROUPS command specification

```
public void readDatabase (Database database, long flags)
```

Initialize a Database object with data from the server. ('lazy load')

Parameters :

`database` - the Database object to initialize with persisted data.

`flags` - the fields of the Database object to initialize with server data. A combination of Flag defined constants, relevant for the Database class objects

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_READ_DATABASE command specification

```
public Database readDatabase (String databaseId, long flags)
```

Initialize a Database object with data from the server based on the database id. ('lazy load')

Parameters :

`databaseId` - the id of the database whose data is requested.

`flags` - the fields of the Database object to initialize with server data. A combination of Flag defined constants, relevant for the Database class objects

Returns :

an initialized {@link Database} object

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_READ_DATABASE command specification

```
public void enumDatabasesByConnection (long flags, List databases)
```

Enumerate databases accessible for connected user.

Parameters :

`flags` - the fields of the Database returned objects to initialize and the order criterion to use for the enumeration

`databases` - the list to fill with Database objects results of the enumeration

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_ENUMERATE_DATABASES command specification

```
public void enumDatabases (List criteriaIds, long flags, Integer offset,
Integer count, List databases)
```

Enumerate databases.

Parameters :

`criteriaIds` - a list of object ids to be used as enumeration criteria (e.g. a user id criterion will return the databases that are assigned to the user specified as criterion)

`flags` - the fields of the Database returned objects to initialize and the order criterion to use for the enumeration

`offset` - starting position for the enumeration. '0' is assumed is none specified

`count` - the maximum count of Database objects to return. 'All' is assumed if none specified

`databases` - the list to fill with Database objects results of the enumeration

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_ENUMERATE_DATABASES command specification

```
public void createRole (Role role)
```

Create a role.

Parameters :

`role` - the Role object to create

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_CREATE_ROLE command specification

```
public void readRole (Role role, long flags)
```

Initialize a Role object with data from the server. ('lazy load')

Parameters :

`role` - the Role object to initialize with persisted data.

`flags` - the fields of the Role object to initialize with server data. A combination of Flag defined

constants, relevant for the Role class objects

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_READ_ROLE command specification

```
public Role readRole (String roleId, long flags)
```

Initialize a Role object with data from the server based on the role id. ('lazy load')

Parameters :

`roleId` - the id of the role whose data is requested.

`flags` - the fields of the Role object to initialize with server data. A combination of Flag defined constants, relevant for the Role class objects

Returns :

an initialized {@link Role} object

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_READ_ROLE command specification

```
public void updateRole (Role role, long flags)
```

Update role data.

Parameters :

`role` - the Role object containing the data to update

`flags` - the fields from the Role object with to update. A combination of Flag defined constants, relevant for the Role class objects

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_UPDATE_ROLE command specification

```
public void deleteRole (String roleId)
```

Delete a role.

Parameters :

`roleId` - id of role to delete

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_DELETE_ROLE command specification

```
public void enumRoles (List criteriaIds, long flags, Integer offset,
Integer count, List roles)
```

Enumerate roles.

Parameters :

`criteriaIds` - a list of object ids to be used as enumeration criteria (e.g. a database id criterion will return the roles that are assigned to the database specified as criterion)

`flags` - the fields of the Role returned objects to initialize and the order criterion to use for the enumeration

`offset` - starting position for the enumeration. '0' is assumed is none specified

`count` - the maximum count of Role objects to return. 'All' is assumed if none specified

`roles` - the list to fill with Role objects results of the enumeration

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_ENUMERATE_ROLES command specification

```
public void readStorage (Storage storage, long flags)
```

Initialize a Storage object with data from the server. ('lazy load')

Parameters :

`storage` - the Storage object to initialize with persisted data.

`flags` - the fields of the Storage object to initialize with server data. A combination of Flag defined constants, relevant for the Storage class objects

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_READ_STORAGE command specification

```
public Storage readStorage (String storageId, long flags)
```

Initialize a Storage object with data from the server based on the rule id. ('lazy load')

Parameters :

`storageId` - the id of the storage whose data is requested.

`flags` - the fields of the Storage object to initialize with server data. A combination of Flag defined constants, relevant for the Storage class objects

Returns :

an initialized {@link Rule} object

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_READ_STORAGE command specification

```
public void updateStorage (Storage storage, long flags)
```

Update storage data.

Parameters :

`storage` - the Storage object containing the data to update

`flags` - the fields from the Storage object with to update. A combination of Flag defined constants, relevant for the Storage class objects

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_UPDATE_STORAGE command specification

```
public void enumStorages (List criteriaIds, long flags, Integer offset,
Integer count, List storages)
```

Enumerate storages.

Parameters :

`criteriaIds` - **unused**, must be null

`flags` - the fields of the Storage returned objects to initialize and the order criterion to use for the enumeration

`offset` - starting position for the enumeration. '0' is assumed is none specified

`count` - the maximum count of Storage objects to return. 'All' is assumed if none specified

`storages` - the list to fill with Storage objects results of the enumeration

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal

connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_ENUMERATE_STORAGES command specification

```
public void readPortal (Portal portal, long flags)
```

Initialize a Portal object with data from the server. ('lazy load')

Parameters :

`portal` - the Portal object to initialize with persisted data.

`flags` - the fields of the Portal object to initialize with server data. A combination of Flag defined constants, relevant for the Portal class objects

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_READ_PORTAL command specification

```
public Portal readPortal (String portalId, long flags)
```

Initialize a Portal object with data from the server based on the portal id. ('lazy load')

Parameters :

`portalId` - the id of the portal whose data is requested.

`flags` - the fields of the Portal object to initialize with server data. A combination of Flag defined constants, relevant for the Portal class objects

Returns :

an initialized {@link Portal} object

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_READ_PORTAL command specification

```
public void enumPortals (List criteriaIds, long flags, Integer offset,
Integer count, List portals)
```

Enumerate portals.

Parameters :

`criteriaIds` - **unused**, must be null

`flags` - the fields of the Portal returned objects to initialize and the order criterion to use for the enumeration

`offset` - starting position for the enumeration. '0' is assumed is none specified

`count` - the maximum count of Portal objects to return. 'All' is assumed if none specified

`portals` - the list to fill with Portal objects results of the enumeration

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_ENUMERATE_PORTALS command specification

```
public void createResource (Resource resource)
```

Create a resource.

Parameters :

`resource` - the Resource object to create

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_CREATE_RESOURCE command specification

```
public void readResource (Resource resource, long flags)
```

Initialize a Resource object with data from the server. ('lazy load')

Parameters :

`resource` - the Resource object to initialize with persisted data.

`flags` - the fields of the Resource object to initialize with server data. A combination of Flag defined constants, relevant for the Resource class objects

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_READ_RESOURCE command specification

```
public Resource readResource (String resourceId, long flags)
```

Initialize a Resource object with data from the server based on the resource id. ('lazy load')

Parameters :

`resourceId` - the id of the resource whose data is requested.

`flags` - the fields of the Resource object to initialize with server data. A combination of Flag defined constants, relevant for the Resource class objects

Returns :

an initialized {@link Resource} object

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_READ_RESOURCE command specification

```
public void updateResource (Resource resource, long flags)
```

Update resource data.

Parameters :

`resource` - the Resource object containing the data to update

`flags` - the fields from the Resource object with to update. A combination of Flag defined constants, relevant for the Resource class objects

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_UPDATE_RESOURCE command specification

```
public void deleteResource (String resourceId)
```

Delete a resource.

Parameters :

`resourceId` - id of resource to delete

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_DELETE_RESOURCE command specification

```
public void enumResources (List criteriaIds, long flags, Integer offset,
Integer count, List resources)
```

Enumerate resources.

Parameters :

`criteriaIds` - **unused**, must be null

`flags` - the fields of the Resource returned objects to initialize and the order criterion to use for the enumeration

`offset` - starting position for the enumeration. '0' is assumed is none specified

`count` - the maximum count of Resource objects to return. 'All' is assumed if none specified

`resources` - the list to fill with Resource objects results of the enumeration

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_ENUMERATE_RESOURCES command specification

```
public Assignment createGroupUserAssig (String objGroupId, String
objUserId)
```

Create a group-user assignment.

Parameters :

`objGroupId` - id of a Group object

`objUserId` - id of a User object

Returns :

an {@link Assignment} object with the id field set

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_CREATE_GROUP_USER command specification

```
public Assignment readGroupUserAssig (String objGroupId, String
objUserId)
```

Check if a group-user assignment exists.

Parameters :

`objGroupId` - id of a Group object

`objUserId` - id of a User object

Returns :

an {@link Assignment} object with the id field set

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_READ_GROUP_USER command specification

```
public Assignment readGroupUserAssig (String assigId)
```

Check if a group-user assignment exists.

Parameters :

`assigId` - the id of the Assignment object

Returns :

an initialized {@link Assignment} object

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_READ_GROUP_USER command specification

```
public void deleteGroupUserAssig (String objGroupId, String objUserId)
```

Delete a group-user assignment.

Parameters :

`objGroupId` - id of a Group object

`objUserId` - id of a User object

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_DELETE_GROUP_USER command specification

```
public void deleteGroupUserAssig (String assigId)
```

Delete a group-user assignment.

Parameters :

`assigId` - id of Assignment object to delete

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_DELETE_GROUP_USER command specification

```
public void enumGroupUserAssig (List criteriaIds, Integer offset,
Integer count, List assigs)
```

Enumerate group-user assignments.

Parameters :

`criteriaIds` - a list of object ids to be used as enumeration criteria (e.g. a user id criterion will return the groups to which user specified as criterion is assigned)

`offset` - starting position for the enumeration. '0' is assumed is none specified

`count` - the maximum count of Assignment objects to return. 'All' is assumed if none specified

`assigs` - the list to fill with Assignment objects, results of the enumeration

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_ENUMERATE_GROUP_USER command specification

```
public Assignment createGroupPortalAssig (String objGroupId, String
objPortalId)
```

Create a group-portal assignment.

Parameters :

`objGroupId` - id of a Group object

`objPortalId` - id of a Portal object

Returns :

an {@link Assignment} object with the id field set

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_CREATE_GROUP_PORTAL command specification

```
public Assignment readGroupPortalAssig (String objGroupId, String
objPortalId)
```

Check if a group-portal assignment exists.

Parameters :

`objGroupId` - id of a Group object

`objPortalId` - id of a Portal object

Returns :

an {@link Assignment} object with the id field set

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_READ_GROUP_PORTAL command specification

```
public Assignment readGroupPortalAssig (String assigId)
```

Check if a group-portal assignment exists.

Parameters :

`assigId` - the id of the Assignment object

Returns :

an initialized {@link Assignment} object

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_READ_GROUP_PORTAL command specification

```
public void deleteGroupPortalAssig (String objGroupId, String
objPortalId)
```

Delete a group-portal assignment.

Parameters :

`objGroupId` - id of a Group object

`objPortalId` - id of a Portal object

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_DELETE_GROUP_PORTAL command specification

```
public void deleteGroupPortalAssig (String assigId)
```

Delete a group-portal assignment.

Parameters :

`assigId` - id of Assignment object to delete

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_DELETE_GROUP_PORTAL command specification

```
public void enumGroupPortalAssig (List criteriaIds, Integer offset,
Integer count, List assigs)
```

Enumerate group-portal assignments.

Parameters :

`criteriaIds` - a list of object ids to be used as enumeration criteria (e.g. a group id criterion will return the portals to which group specified as criterion is assigned)

`offset` - starting position for the enumeration. '0' is assumed is none specified

`count` - the maximum count of Assignment objects to return. 'All' is assumed if none specified

`assigs` - the list to fill with Assignment objects, results of the enumeration

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_ENUMERATE_GROUP_PORTAL command specification

```
public Assignment createUserDatabaseAssig (String objUserId, String objDatabaseId)
```

Create a user-database assignment.

Parameters :

`objUserId` - id of a User object

`objDatabaseId` - id of a Database object

Returns :

an {@link Assignment} object with the id field set

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_CREATE_USER_DATABASE command specification

```
public Assignment readUserDatabaseAssig (String objUserId, String objDatabaseId)
```

Check if a user-database assignment exists.

Parameters :

`objUserId` - id of a User object

`objDatabaseId` - id of a Database object

Returns :

an {@link Assignment} object with the id field set

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_READ_USER_DATABASE command specification

```
public Assignment readUserDatabaseAssig (String assigId)
```

Check if a user-database assignment exists.

Parameters :

`assigId` - the id of the Assignment object

Returns :

an initialized {@link Assignment} object

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_READ_USER_DATABASE command specification

```
public void deleteUserDatabaseAssig (String objUserId, String
objDatabaseId)
```

Delete a user-database assignment.

Parameters :

`objUserId` - id of a User object

`objDatabaseId` - id of a Database object

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_DELETE_USER_DATABASE command specification

```
public void deleteUserDatabaseAssig (String assigId)
```

Delete a user-database assignment.

Parameters :

`assigId` - id of Assignment object to delete

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_DELETE_USER_DATABASE command specification

```
public void enumUserDatabaseAssig (List criteriaIds, Integer offset,
Integer count, List assigs)
```

Enumerate user-database assignments.

Parameters :

`criteriaIds` - a list of object ids to be used as enumeration criteria (e.g. a user id criterion will return the database to which user specified as criterion is assigned)

`offset` - starting position for the enumeration. '0' is assumed is none specified

`count` - the maximum count of Assignment objects to return. 'All' is assumed if none specified

`assigs` - the list to fill with Assignment objects, results of the enumeration

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_ENUMERATE_USER_DATABASE command specification

```
public Assignment createUserPortalAssig (String objUserId, String
objPortalId)
```

Create a user-portal assignment.

Parameters :

`objUserId` - id of a User object

`objPortalId` - id of a Portal object

Returns :

an {@link Assignment} object with the id field set

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_CREATE_USER_PORTAL command specification

```
public Assignment readUserPortalAssig (String objUserId, String
objPortalId)
```

Check if a user-portal assignment exists.

Parameters :

`objUserId` - id of a User object

`objPortalId` - id of a Portal object

Returns :

an {@link Assignment} object with the id field set

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_READ_USER_PORTAL command specification

```
public Assignment readUserPortalAssig (String assigId)
```

Check if a user-portal assignment exists.

Parameters :

`assigId` - the id of the Assignment object

Returns :

an initialized {@link Assignment} object

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_READ_USER_PORTAL command specification

```
public void deleteUserPortalAssig (String objUserId, String objPortalId)
```

Delete a user-portal assignment.

Parameters :

`objUserId` - id of a User object

`objPortalId` - id of a Portal object

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_DELETE_USER_PORTAL command specification

```
public void deleteUserPortalAssig (String assigId)
```

Delete a user-portal assignment.

Parameters :

`assigId` - id of Assignment object to delete

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_DELETE_USER_PORTAL command specification

```
public void enumUserPortalAssig (List criteriaIds, Integer offset,
Integer count, List assigs)
```

Enumerate user-portal assignments.

Parameters :

`criteriaIds` - a list of object ids to be used as enumeration criteria (e.g. a user id criterion will

return the portals to which user specified as criterion is assigned)

`offset` - starting position for the enumeration. '0' is assumed is none specified

`count` - the maximum count of Assignment objects to return. 'All' is assumed if none specified

`assigs` - the list to fill with Assignment objects, results of the enumeration

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_ENUMERATE_USER_PORTAL command specification

```
public Assignment createGroupDatabaseAssig (String objGroupId, String objDatabaseId)
```

Create a group-database assignment.

Parameters :

`objGroupId` - id of a Group object

`objDatabaseId` - id of a Database object

Returns :

an {@link Assignment} object with the id field set

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_CREATE_GROUP_DATABASE command specification

```
public Assignment readGroupDatabaseAssig (String objGroupId, String objDatabaseId)
```

Check if a group-database assignment exists.

Parameters :

`objGroupId` - id of a Group object

`objDatabaseId` - id of a Database object

Returns :

an {@link Assignment} object with the id field set

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_READ_GROUP_DATABASE command specification

```
public Assignment readGroupDatabaseAssig (String assigId)
```

Check if a group-database assignment exists.

Parameters :

`assigId` - the id of the Assignment object

Returns :

an initialized {@link Assignment} object

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_READ_GROUP_DATABASE command specification

```
public void deleteGroupDatabaseAssig (String objGroupId, String
objDatabaseId)
```

Delete a group-database assignment.

Parameters :

`objGroupId` - id of a Group object

`objDatabaseId` - id of a Database object

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_DELETE_GROUP_DATABASE command specification

```
public void deleteGroupDatabaseAssig (String assigId)
```

Delete a group-database assignment.

Parameters :

`assigId` - id of Assignment object to delete

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_DELETE_GROUP_DATABASE command specification

```
public void enumGroupDatabaseAssig (List criteriaIds, Integer offset,
Integer count, List assigs)
```

Enumerate group-database assignments.

Parameters :

`criteriaIds` - a list of object ids to be used as enumeration criteria (e.g. a group id criterion will return the database to which group specified as criterion is assigned)

`offset` - starting position for the enumeration. '0' is assumed is none specified

`count` - the maximum count of Assignment objects to return. 'All' is assumed if none specified

`assigs` - the list to fill with Assignment objects, results of the enumeration

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_ENUMERATE_GROUP_DATABASE command specification

```
public Assignment createRoleDatabaseAssig (String objRoleId, String objDatabaseId)
```

Create a role-database assignment.

Parameters :

`objRoleId` - id of a Role object

`objDatabaseId` - id of a Database object

Returns :

an {@link Assignment} object with the id field set

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_CREATE_ROLE_DATABASE command specification

```
public Assignment readRoleDatabaseAssig (String objRoleId, String objDatabaseId)
```

Check if a role-database assignment exists.

Parameters :

`objRoleId` - id of a Role object

`objDatabaseId` - id of a Database object

Returns :

an {@link Assignment} object with the id field set

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal

connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_READ_ROLE_DATABASE command specification

```
public Assignment readRoleDatabaseAssig (String assigId)
```

Check if a role-database assignment exists.

Parameters :

`assigId` - the id of the Assignment object

Returns :

an initialized {@link Assignment} object

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_READ_ROLE_DATABASE command specification

```
public void deleteRoleDatabaseAssig (String objRoleId, String
objDatabaseId)
```

Delete a role-database assignment.

Parameters :

`objRoleId` - id of a Role object

`objDatabaseId` - id of a Database object

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_DELETE_ROLE_DATABASE command specification

```
public void deleteRoleDatabaseAssig (String assigId)
```

Delete a role-database assignment.

Parameters :

`assigId` - id of Assignment object to delete

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_DELETE_ROLE_DATABASE command specification

```
public void enumRoleDatabaseAssig (List criteriaIds, Integer offset,
Integer count, List assigs)
```

Enumerate role-database assignments.

Parameters :

`criteriaIds` - a list of object ids to be used as enumeration criteria (e.g. a role id criterion will return the databases to which role specified as criterion is assigned)

`offset` - starting position for the enumeration. '0' is assumed is none specified

`count` - the maximum count of Assignment objects to return. 'All' is assumed if none specified

`assigs` - the list to fill with Assignment objects, results of the enumeration

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_ENUMERATE_ROLE_DATABASE command specification

```
public Assignment createRolePortalAssig (String objRoleId, String
objPortalId)
```

Create a role-portal assignment.

Parameters :

`objRoleId` - id of a Role object

`objPortalId` - id of a Portal object

Returns :

an {@link Assignment} object with the id field set

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_CREATE_ROLE_PORTAL command specification

```
public Assignment readRolePortalAssig (String objRoleId, String
objPortalId)
```

Check if a role-portal assignment exists.

Parameters :

`objRoleId` - id of a Role object

`objPortalId` - id of a Portal object

Returns :

an {@link Assignment} object with the id field set

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_READ_ROLE_PORTAL command specification

```
public Assignment readRolePortalAssig (String assigId)
```

Check if a portal-role assignment exists.

Parameters :

`assigId` - the id of the Assignment object

Returns :

an initialized {@link Assignment} object

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_READ_ROLE_PORTAL command specification

```
public void deleteRolePortalAssig (String objRoleId, String objPortalId)
```

Delete a role-portal assignment.

Parameters :

`objRoleId` - id of a Role object

`objPortalId` - id of a Portal object

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_DELETE_ROLE_PORTAL command specification

```
public void deleteRolePortalAssig (String assigId)
```

Delete a role-portal assignment.

Parameters :

`assigId` - id of Assignment object to delete

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_DELETE_ROLE_PORTAL command specification

```
public void enumRolePortalAssig (List criteriaIds, Integer offset,
```

```
Integer count, List assigs)
```

Enumerate role-portal assignments.

Parameters :

`criteriaIds` - a list of object ids to be used as enumeration criteria (e.g. a role id criterion will return the portals to which role specified as criterion is assigned)

`offset` - starting position for the enumeration. '0' is assumed is none specified

`count` - the maximum count of Assignment objects to return. 'All' is assumed if none specified

`assigs` - the list to fill with Assignment objects, results of the enumeration

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_ENUMERATE_ROLE_PORTAL command specification

```
public Assignment createUserRoleAssig (String objUserId, String
objRoleId)
```

Create a user-role assignment.

Parameters :

`objUserId` - id of a User object

`objRoleId` - id of a Role object

Returns :

an {@link Assignment} object with the id field set

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_CREATE_USER_ROLE command specification

```
public Assignment readUserRoleAssig (String objUserId, String objRoleId)
```

Check if a user-role assignment exists.

Parameters :

`objUserId` - id of a User object

`objRoleId` - id of a Role object

Returns :

an {@link Assignment} object with the id field set

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_READ_USER_ROLE command specification

```
public Assignment readUserRoleAssig (String assigId)
```

Check if a user-role assignment exists.

Parameters :

`assigId` - the id of the Assignment object

Returns :

an initialized {@link Assignment} object

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_READ_USER_ROLE command specification

```
public void deleteUserRoleAssig (String objUserId, String objRoleId)
```

Delete a user-role assignment.

Parameters :

`objUserId` - id of a User object

`objRoleId` - id of a Role object

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_DELETE_USER_ROLE command specification

```
public void deleteUserRoleAssig (String assigId)
```

Delete a user-role assignment.

Parameters :

`assigId` - id of Assignment object to delete

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_DELETE_USER_ROLE command specification

```
public void enumUserRoleAssig (List criteriaIds, Integer offset, Integer
```

```
count, List assigs)
```

Enumerate user-role assignments.

Parameters :

`criteriaIds` - a list of object ids to be used as enumeration criteria (e.g. a user id criterion will return the roles assigned to user specified as criterion)

`offset` - starting position for the enumeration. '0' is assumed is none specified

`count` - the maximum count of Assignment objects to return. 'All' is assumed if none specified

`assigs` - the list to fill with Assignment objects, results of the enumeration

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_ENUMERATE_USER_ROLE command specification

```
public Assignment createGroupRoleAssig (String objGroupId, String
objRoleId)
```

Create a group-role assignment.

Parameters :

`objGroupId` - id of a Group object

`objRoleId` - id of a Role object

Returns :

an {@link Assignment} object with the id field set

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_CREATE_GROUP_ROLE command specification

```
public Assignment readGroupRoleAssig (String objGroupId, String
objRoleId)
```

Check if a group-role assignment exists.

Parameters :

`objGroupId` - id of a Group object

`objRoleId` - id of a Role object

Returns :

an {@link Assignment} object with the id field set

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_READ_GROUP_ROLE command specification

```
public Assignment readGroupRoleAssig (String assigId)
```

Check if a group-role assignment exists.

Parameters :

`assigId` - the id of the Assignment object

Returns :

an initialized {@link Assignment} object

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_READ_GROUP_ROLE command specification

```
public void deleteGroupRoleAssig (String objGroupId, String objRoleId)
```

Delete a group-role assignment.

Parameters :

`objGroupId` - id of a Group object

`objRoleId` - id of a Role object

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_DELETE_GROUP_ROLE command specification

```
public void deleteGroupRoleAssig (String assigId)
```

Delete a group-role assignment.

Parameters :

`assigId` - id of Assignment object to delete

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the

AdministrationLibrary#CMD_DELETE_GROUP_ROLE command specification

```
public void enumGroupRoleAssig (List criteriaIds, Integer offset,
Integer count, List assigs)
```

Enumerate group-role assignments.

Parameters :

`criteriaIds` - a list of object ids to be used as enumeration criteria (e.g. a role id criterion will return the groups to which role specified as criterion is assigned)

`offset` - starting position for the enumeration. '0' is assumed is none specified

`count` - the maximum count of Assignment objects to return. 'All' is assumed if none specified

`assigs` - the list to fill with Assignment objects, results of the enumeration

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_ENUMERATE_GROUP_ROLE command specification

```
public void enumAttributes (List criteriaIds, long flags, Integer
offset, Integer count, List attributes)
```

Enumerate attributes.

Parameters :

`criteriaIds` - a list of object ids to be used as enumeration criteria

`flags` - the fields of the Attribute returned objects to initialize and the order criterion to use for the enumeration

`offset` - starting position for the enumeration. '0' is assumed is none specified

`count` - the maximum count of Attribute objects to return. 'All' is assumed if none specified

`attributes` - the list to fill with Attribute objects results of the enumeration

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the DesignLibrary#CMD_ENUMERATE_ATTRIBUTES command specification

```
public void enumForms (List criteriaIds, long flags, Integer offset,
Integer count, List forms)
```

Enumerate forms.

Parameters :

`criteriaIds` - a list of object ids to be used as enumeration criteria

`flags` - the fields of the Form returned objects to initialize and the order criterion to use for the enumeration

`offset` - starting position for the enumeration. '0' is assumed is none specified

`count` - the maximum count of Form objects to return. 'All' is assumed if none specified

`attributes` - the list to fill with Form objects results of the enumeration

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the DesignLibrary#CMD_ENUMERATE_FORMS command specification

```
public void enumViews (List criteriaIds, long flags, Integer offset,
Integer count, List views)
```

Enumerate views.

Parameters :

`criteriaIds` - a list of object ids to be used as enumeration criteria

`flags` - the fields of the View returned objects to initialize and the order criterion to use for the enumeration

`offset` - starting position for the enumeration. '0' is assumed is none specified

`count` - the maximum count of View objects to return. 'All' is assumed if none specified

`attributes` - the list to fill with View objects results of the enumeration

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the DesignLibrary#CMD_ENUMERATE_VIEWS command specification

```
public void createFolder (Folder folder, String parentId)
```

Create a Folder.

Parameters :

`folder` - the folder to create

`parentId` - id of database to create folder if no `parentId` specified by the `folder` object

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the GedLibrary#CMD_CREATE_FOLDER command specification

```
public void readFolder (Folder folder, long flags)
```

Initialize a Folder object with data from the server. ('lazy load')

Parameters :

`folder` - the Folder object to initialize with persisted data. Normally it will specify the `id`

`flags` - the fields of the Folder object to initialize with server data. A combination of Flag defined constants, relevant for the Folder class objects

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the GedLibrary#CMD_READ_FOLDER command specification

```
public Folder readFolder (String folderId, long flags)
```

Get a Folder object from the server based on it's id. ('lazy load')

Parameters :

`folderId` - the id of a Folder object

`flags` - the fields of the Folder object to initialize with server data. A combination of Flag defined constants, relevant for the Folder class objects

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the GedLibrary#CMD_READ_FOLDER command specification

```
public void updateFolder (Folder folder, long flags)
```

Update a Folder object.

Parameters :

`folder` - the Folder object to containing the data to update

`flags` - the fields of the Folder object to update. A combination of Flag defined constants, relevant for the Folder class objects

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal

connection error (server inaccessible, expired connection etc.) and other errors defined by the GedLibrary#CMD_UPDATE_FOLDER command specification

```
public void deleteFolder (String folderId)
```

Delete a Folder object.

Parameters :

`folderId` - the id of the Folder object to delete

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the GedLibrary#CMD_DELETE_FOLDER command specification

```
public boolean checkTrash (String databaseId)
```

Check trash

Returns :

`True` if trash in use , `False` otherwise

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the GedLibrary#CMD_CHECK_TRASH command specification

```
public void enumFolders (List criteriaIds, long flags, Integer offset,
Integer count, List folders)
```

Enumerate folders based on a list of criteria.

Parameters :

`citeriaIds` - the list of criteria ids. Generally this will contain 1 element - the id of parent folder.

`flags` - the fields of the Folder object to initialize with server data. A combination of Flag defined constants, relevant for the Folder class objects. Also, a list of order options defined by the Flag class can be passed through this parameter

`offset` - offset from which to start enumeration. E.g. if offset is 2, then the enumeration will return elements starting with the third folder.

`count` - number of maximum elements to return

`folders` - the list to fill with enumeration results

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal

connection error (server inaccessible, expired connection etc.) and other errors defined by the GedLibrary#CMD_ENUMERATE_FOLDERS command specification

```
public void moveFolder (String objectId, String targetId)
```

Move a folder to another location (folder/database root).

Parameters :

`objectId` - the id of the Folder object to move to a new parent

`targetId` - the id of the new parent for the object to move

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the GedLibrary#CMD_MOVE_FOLDER command specification

```
public void createDocument (Document document, String parentId)
```

Create a Document.

Parameters :

`document` - the document to create

`parentId` - id of folder or pin where to create the document

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the GedLibrary#CMD_CREATE_DOCUMENT command specification

```
public void createDocument (Document document, String parentId, DTO
extra)
```

Create a Document.

Parameters :

`document` - the document to create

`parentId` - id of folder or pin where to create the document

`extra` - additional parameters as DTO object

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the GedLibrary#CMD_CREATE_DOCUMENT command specification

```
public void readDocument (Document document, long flags)
```

Initialize a Document object with data from the server. ('lazy load') 'Pin' documents are also read using this call.

Parameters :

`document` - the Document object to initialize with persisted data. Normally it will specify the `id`

`flags` - the fields of the Document object to initialize with server data. A combination of Flag defined constants, relevant for the Document class objects

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the GedLibrary#CMD_READ_DOCUMENT command specification

```
public Document readDocument (String documentId, long flags)
```

Get a Document object from the server based on it's id. ('lazy load')

Parameters :

`documentId` - the id of a Folder object

`flags` - the fields of the Document object to initialize with server data. A combination of Document defined constants, relevant for the Document class objects

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the GedLibrary#CMD_READ_DOCUMENT command specification

```
public void updateDocument (Document document, long flags)
```

Update a Document object.

Parameters :

`document` - the Document object to containing the data to update

`flags` - the fields of the Document object to update. A combination of Flag defined constants, relevant for the Document class objects

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the GedLibrary#CMD_UPDATE_DOCUMENT command specification

```
public void updateDocument (Document document, long flags, DTO extra)
```

Update a Document object.

Parameters :

`document` - the Document object to containing the data to update

`flags` - the fields of the Document object to update. A combination of Flag defined constants, relevant for the Document class objects

`extra` - additional parameters as DTO object

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the GedLibrary#CMD_UPDATE_DOCUMENT command specification

```
public void deleteDocument (String documentId)
```

Delete a Document object.

Parameters :

`documentId` - the id of the Document object to delete

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the GedLibrary#CMD_DELETE_DOCUMENT command specification

```
public void enumDocuments (List criteriaIds, long flags, Integer offset,
Integer count, List documents)
```

Enumerate document based on a list of criteria.

Parameters :

`citeriaIds` - the list of criteria ids. Generally this will contain 1 element - the id of parent folder, process or pin.

`flags` - the fields of the Document object to initialize with server data. A combination of Flag defined constants, relevant for the Document class objects. Also, a list of order options defined by the Flag class can be passed through this parameter

`offset` - offset from which to start enumeration. E.g. if offset is 2, then the enumeration will return elements starting with the third document.

`count` - number of maximum objects to return

`folders` - the list to fill with enumeration results

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the GedLibrary#CMD_ENUMERATE_DOCUMENTS command specification

```
public String moveDocument (String documentId, String newParentId,
StringBuilder newFicheId)
```

Move a Document from one location to another.

Parameters :

`documentId` - id of document to move

`newParentId` - id of folder, process or pin where to move the document

`newFicheId` - id of resulting fiche document if document to move was last component of a pin. This parameter will be set by the call if not null

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the GedLibrary#CMD_MOVE_DOCUMENT command specification

```
public void addDocumentInFulltext (String documentId)
```

Index the content of a text document in the fulltext database.

Parameters :

`documentId` - id of document whose content to index

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the GedLibrary#CMD_ADD_DOCUMENT_IN_FULLTEXT command specification

```
public String removeDocumentFromFulltext (String documentId)
```

Delete the content index of a text document from the fulltext database.

Parameters :

`documentId` - id of document whose content index to delete

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the GedLibrary#CMD_ADD_DOCUMENT_IN_FULLTEXT command specification

```
public String addNoteInFulltext (String noteId)
```

Index the content of a note in the fulltext database.

Parameters :

`noteId` - id of note whose content to index

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the GedLibrary#CMD_ADD_NOTE_IN_FULLTEXT command specification

```
public String removeNoteFromFulltext (String noteId)
```

Delete the content index of a note from the fulltext database.

Parameters :

`noteId` - id of note whose content index to delete

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the GedLibrary#CMD_REMOVE_NOTE_FROM_FULLTEXT command specification

```
public void pinDocuments (Document pin, List docsIds, String parentId)
```

Pin a group of documents.

Parameters :

`pin` - the 'pin' document to initialize. This will be the resulting pin. On return it will be initialized with fields relevant to a 'create' operation

`docsIds` - the list of documents to pin specify as ids

`parentId` - the id of the folder or process where the 'pin' document will be created

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the GedLibrary#CMD_PIN_DOCUMENTS command specification

```
public void unpinDocuments (List docsIds, Hashtable resultsIds)
```

Unpin pin/pinned documents.

Parameters :

`docsIds` - a list of 'pin'/'pinned' documents to 'explode'/'extract'

`resultsIds` - the (original id) - to - (new id) correspondence. This parameter will filled by the command

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the GedLibrary#CMD_UNPIN_DOCUMENTS command specification

```
public DTO createNote (Note note, String parentId)
```

Create a note.

Parameters :

`note` - the Note object to create

`parentId` - if of a Database object

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the GedLibrary#CMD_CREATE_NOTE command specification

```
public void readNote (Note note, long flags)
```

Initialize a Note object with data from the server. ('lazy load')

Parameters :

`note` - the Note object to initialize with persisted data.

`flags` - the fields of the Note object to initialize with server data. A combination of Flag defined constants, relevant for the Note class objects

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the GedLibrary#CMD_READ_NOTE command specification

```
public Note readNote (String noteId, long flags)
```

Initialize a Note object with data from the server based on the note id. ('lazy load')

Parameters :

`noteId` - the id of the note whose data is requested.

`flags` - the fields of the Note object to initialize with server data. A combination of Flag defined constants, relevant for the Note class objects

Returns :

an initialized {@link Note} object

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the GedLibrary#CMD_READ_NOTE command specification

```
public void updateNote (Note note, long flags)
```

Update note data.

Parameters :

`note` - the Note object containing the data to update

`flags` - the fields from the Note object with to update. A combination of Flag defined constants, relevant for the Note class objects

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the GedLibrary#CMD_UPDATE_NOTE command specification

```
public void deleteNote (String noteId)
```

Delete a note.

Parameters :

`noteId` - id of note to delete

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the GedLibrary#CMD_DELETE_NOTE command specification

```
public void enumNotes (List criteriaIds, long flags, Integer offset,
Integer count, List notes)
```

Enumerate notes.

Parameters :

`criteriaIds` - a list of object ids to be used as enumeration criteria

`flags` - the fields of the Note returned objects to initialize and the order criterion to use for the enumeration

`offset` - starting position for the enumeration. '0' is assumed is none specified

`count` - the maximum count of Note objects to return. 'All' is assumed if none specified

`attributes` - the list to fill with Note objects results of the enumeration

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the GedLibrary#CMD_ENUMERATE_NOTES command specification

```
public String storeFile (String databaseId, String path)
```

Store a file on the server

Parameters :

`databaseId` - the id of com.siatel.domain.Database object. The administration database is specified as 'DATA00000000000000000000000000000000000000'.

`path` - the path to the file to store

Returns :

the id of the stored file

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the StorageLibrary#CMD_STORE_FILE command specification

```
public String storeSystemFile (String databaseId, String path)
```

Store a file on the server, using the system storage

Parameters :

`databaseId` - the id of Database object. The administration database is specified as 'DATA00000000000000000000000000000000000000'.

`path` - the path to the file to store

Returns :

the id of the stored file

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the StorageLibrary#CMD_STORE_FILE command specification

```
public InputStream retrieveFile (String fileId, String objectId)
```

Retrieve a previously stored file.

Parameters :

`fileId` - id of previously stored file

Returns :

the {@link InputStream} of the stored file

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the StorageLibrary#CMD_RETRIEVE_FILE command specification

```
public void deleteFile (String fileId)
```

Delete a previously stored file.

Parameters :

`fileId` - the id of the stored file to delete

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the StorageLibrary#CMD_DELETE_FILE command specification

```
public void replaceFile (String fileId, String filePath)
```

Replace a previously stored file.

Parameters :

`fileId` - the id of the stored file to replace

`filePath` - the path to the file to replace with

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the StorageLibrary#CMD_REPLACE_FILE command specification

```
public String storeThumbFile (String fileId, String thumbPath)
```

Store a thumbnail image file for a specified document file.

Parameters :

`fileId` - the document file id for whom the thumbnail image is stored

`thumbPath` - the path to the thumbnail image file

Returns :

the id of the stored thumbnail file

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the StorageLibrary#CMD_STORE_THUMB command specification

```
public InputStream retrieveThumbFile (String fileId)
```

Retrieve a previously stored thumbnail file.

Parameters :

`fileId` - id of previously stored document/thumbnail file

Returns :

the {@link InputStream} of the stored thumbnail file

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the StorageLibrary#CMD_RETRIEVE_THUMB command specification

```
public void deleteThumbFile (String fileId)
```

Delete a previously stored thumbnail file.

Parameters :

`fileId` - the id of the stored thumbnail file to delete or id of document file whose thumb to delete

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the StorageLibrary#CMD_DELETE_THUMB command specification

```
public boolean checkThumbFile (String fileId)
```

Check if there is a thumnail image associated with a previously stored file.

Parameters :

`fileId` - id of the file to check for assicated thumnail image file

Returns :

`true` if thumbnail exists, `false` otherwise

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the StorageLibrary#CMD_CHECK_THUMB command specification

```
public boolean checkFile (String fileId)
```

Check if there is a physical file associated with a previously stored file.

Parameters :

`fileId` - id of the file to check for

Returns :

`true` if file is accessible, `false` otherwise

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the StorageLibrary#CMD_CHECK_THUMB command specification

```
public void enumAuditEntries (Date startDate, Date endDate, List
criteriaIds, long flags, Integer offset, Integer count, List entries)
```

Query the log/audit of an object.

Parameters :

`startDate` - lower limit of time domain to query

`endDate` - upper limit of time domain to query

`criteriaIds` - a list criteria that specify what to look for.See AuditLibrary#CMD_QUERY for valid criteria combinations.

`flags` - a group of flags to further customize the query

`offset` - an initial offset for the enumeration

`count` - the maximum number of entries to return

`entries` - the list to fill with results

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AuditLibrary#CMD_QUERY command specification

```
public int countEntries (Date startDate, Date endDate, List criteriaIds,
long flags)
```

Count the log/audit of an object.

Parameters :

`startDate` - lower limit of time domain to query

`endDate` - upper limit of time domain to query

`criteriaIds` - a list criteria that specify what to look for.See AuditLibrary#CMD_COUNT for valid criteria combinations.

`flags` - a group of flags to further customize the query

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AuditLibrary#CMD_COUNT command specification

```
public void createDocumentVersion (DocumentVersion version, String documentId, String versionType)
```

Create a DocumentVersion.

Parameters :

`version` - the version object to create

`documentId` - id of document whose version is being created

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the GedLibrary#CMD_CREATE_VERSION command specification

```
public DTO deleteUsersWithCriteria (DTO dtoIn)
```

Delete user with criteria.

Parameters :

`criteriaIds` - criteria ids for users to delete

Throws :

`SiatelException` - generic exception specifier that wraps CommunicationException to signal connection error (server inaccessible, expired connection etc.) and other errors defined by the AdministrationLibrary#CMD_DELETE_USER command specification

Java API Examples

This chapter demonstrates how to use the `SiatelClient` object. Along this chapter we will create a helper object named `Toolbox` that will encapsulate all functionality needed for communicating with a Gargantua server and accomplish basic tasks.

The Toolbox object

```
package utils.connecteursJava

import java.util.ArrayList;
import java.util.Collections;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

import utils.common.ConnectionInfos;
import utils.common.Utils;
import utils.connecteursJava.Toolbox;

import com.siatel.error.DeniedSecurityException;
import com.siatel.error.NotFoundPersistenceException;
import com.siatel.error.SiatelException;
import com.siatel.framework.web.AuthContext;
import com.siatel.proxy.http.SiatelClient;
import com.siatel.defs.Identified;
import com.siatel.domain.*;
import com.siatel.dto.DTO;
import com.siatel.dto.DTOFactory;

public final class Toolbox {
    private static volatile Toolbox instance = null;
    private SiatelClient client = null;

    // keep a connections cache
    private Map<String, ConnectionInfos>
        connectionsMap = Collections.synchronizedMap(
            new HashMap<String, ConnectionInfos>());

    private Toolbox(String urlServer) throws Exception{
        try {
            this.client = new SiatelClient(urlServer);
        } catch (Exception e) {
```

```
            StringBuffer sb = new StringBuffer();
            sb
                .append("<ERROR> The SIATEL client connector is not initialized.")
                .append("[url : ")
                .append(urlServer)
                .append(" ] ")
                .append(e.getMessage());
            Utils.log(sb.toString());
            throw new Exception(sb.toString());
        }
    }

    public static void initToolbox(String urlServer)throws Exception{
        Utils.log("------ Gargantua Java connector demo ------");
        if(Toolbox.instance == null){
            synchronized(Toolbox.class) {
                if (Toolbox.instance == null) {
                    Toolbox.instance = new Toolbox(urlServer);
                }
            }
        }
    }

    // --------------------------------------------------------
    // code from the following sections will be inserted here
    // --------------------------------------------------------
}
```

Managing the connection cache

Add the following code to the `Toolbox` object to handle the connection cache.

```
private void addToCache(String conn, String url, String userName){
    if(!instance.connectionsMap.containsKey(conn)){
        ConnectionInfos infos = new ConnectionInfos(url, userName);
        instance.connectionsMap.put(conn, infos);
    }
}

private ConnectionInfos getFromCache(String conn) throws Exception{
    if(instance.connectionsMap.containsKey(conn)){
        return instance.connectionsMap.get(conn);
    }else{
        throw new Exception("The connection cache does not " +
```

```
            "hold the specified connection (" + conn + ")");
    }
}

private String getUserNameFromCache(){
    String conn = instance.client.getConn();
    String userName = "";
    ConnectionInfos infos;
    try {
        infos = instance.getFromCache(conn);
        userName = infos.getUserName();
    } catch (Exception e) {
        e.printStackTrace();
    }
    return userName;
}

private void removeFromCache(String conn){
    if(instance.connectionsMap.containsKey(conn)){
        instance.connectionsMap.remove(conn);
    }
}
```

Getting server information

The `aboutServer` method retrieves server information. The `Toolbox` objects needs to be valid and the connection already opened to the server.

```
// --- Informations about server
public static void aboutServer() throws Exception {
    try {
        StringBuffer sb = new StringBuffer();
        DTO dtoIn = DTOFactory.newDTO();

        dtoIn.put(Constants.SERVER_VERSION_MAJOR, "");
        dtoIn.put(Constants.SERVER_VERSION_MINOR, "");
        dtoIn.put(Constants.SERVER_VERSION_REVISION, "");

        DTO dtoOut = instance.client.getServerInfo(dtoIn);

        // server url
        sb.append("\n").append("Server URL = ").append(instance.client.getServerUrl());

        // server version
```

```
        sb
            .append("\n")
            .append(" -> Major version = ")
            .append(dtoOut.get(Constants.SERVER_VERSION_MAJOR))
            .append("\n")
            .append(" -> Minor version = ")
            .append(dtoOut.get(Constants.SERVER_VERSION_MINOR))
            .append("\n")
            .append(" -> Revision = ")
            .append(dtoOut.get(Constants.SERVER_VERSION_REVISION));
        Utils.log(sb.toString());
    }
    catch (SiatelException e) {
        throw new Exception("Unable to retrieve server " +
        "information.\n"+e.getMessage());
    }
}
```

Opening a user connection to the server

The code below demonstrates how to login to the Gargantua server using user credentials.

```
// --- Open a user connection to the server
public static void doLogin(String username, String password) throws Exception {
    String conn = "";
    String urlServer = instance.client.getServerUrl();
    StringBuffer sb = new StringBuffer();

    try {
        instance.client.login(username, password, AuthContext.RUNTIME);
        conn = instance.client.getConn();
        instance.addToCash(conn, urlServer, username);

        sb.append("Login / ")
            .append(conn)
            .append(" / server [")
            .append(urlServer)
            .append("] / user [")
            .append(username)
            .append("]");
        Utils.log(sb.toString());
    }
    catch (Exception e) {
        sb.append("<ERROR> Login failed ")
```

```
            .append(" / server [")
            .append(urlServer)
            .append("] / user [")
            .append(username)
            .append("]");
        Utils.log(sb.toString());
        sb
             .append(" - e.getMessage : ")
             .append(e.getMessage());
        throw new Exception(sb.toString());
    }
}
```

Closing a user connection

The code below demonstrates how to close a connection to the server.

```
// --- Close a user connection to the server
public static void doLogout() throws Exception {
    String urlserver =  instance.client.getServerUrl();
    String username = "";
    String conn = instance.client.getConn();
    StringBuffer sb = new StringBuffer();

    try {
        instance.client.logout();

        ConnectionInfos infos = instance.getFromCache(conn);
        username = infos.getUserName();
        sb
            .append("Logout / ")
            .append(conn)
            .append(" / server [")
            .append(urlserver)
            .append("] / user [")
            .append(username)
            .append("]");
        Utils.log(sb.toString());

        instance.removeFromCash(conn);

    }
    catch (SiatelException e) {
        sb
```

```
            .append("<ERROR> Logout failed / ")
            .append(conn)
            .append(" / server [")
            .append(urlserver)
            .append("] / user [")
            .append(username)
            .append("]");
        Utils.log(sb.toString());
        sb
            .append(" - e.getMessage : ")
            .append(e.getMessage());
        throw new Exception(sb.toString());
    }
}
```

Reading user information by label

The code below demonstrates how to read a `User` object from the server given it's name. In general the same technique applies to all objects that be requested by label.

```
// read a user identified by it's name
public static User doReadUser(String label) throws Exception {
    User aUser = (User) SiatelFactory.newObject(Prefix.USER);
    aUser.setLabel(label);

    try {
        instance.client.readUser(aUser, Flag.USER_ALL);
        Utils.log(aUser.toString());
        return aUser;
    }
    catch(NotFoundPersistenceException nfp) {
        StringBuffer sb = new StringBuffer();
        sb
            .append("The user identified by label [")
            .append(label)
            .append("] does not exist.");
        Utils.log(sb.toString());
        throw new Exception(sb.toString());
    }
    catch (SiatelException e) {
        StringBuffer sb = new StringBuffer();
        sb
            .append("<ERROR> Read user [")
            .append(label)
```

```
            .append("] command failed.")
            .append(" - e.getMessage() : ")
            .append(e.getMessage());
        Utils.log(sb.toString());
        throw new Exception(sb.toString());
    }
}
```

Update a user

The code below demonstrates how to update a `User` object on the server. In general the same technique applies to all objects that can be updated.

```
// update a user object
public static void doUpdateUser(User aUser) throws Exception {
    try {
        instance.client.updateUser(aUser, Flag.USER_UPDATE_ALL, true);
        Utils.log(aUser.toString());
    }
    catch(NotFoundPersistenceException nfp) {
        StringBuffer sb = new StringBuffer();
        sb
            .append(The user [")
            .append(aUser.getLabel())
            .append("] does not exist.");
        Utils.log(sb.toString());
        throw new Exception(sb.toString());
    }
    catch (SiatelException e) {
        StringBuffer sb = new StringBuffer();
        sb
            .append("<ERROR> Update user [")
            .append(aUser.getLabel())
            .append("] command failed.")
            .append(" - e.getMessage() : ")
            .append(e.getMessage());
        Utils.log(sb.toString());
        throw new Exception(sb.toString());
    }
}
```

List users

The code below demonstrates how to retrieve a list of objects given some selection criteria; in this example, a `User` list is retrieved. In general the same technique is used for reading a collection of objects using selection criteria; however, these selection criterias will vary by object type. See the object definition for more details.

```
public static List doListUsers() throws Exception {
    List criteriaIds = new ArrayList();
    long flags = Flag.ID | Flag.LABEL | Flag.DESCRIPTION | Flag.REAL_NAME;
    Integer offset = 0;
    Integer count = -1;
    List users = new ArrayList();

    try {
        instance.client.enumUsers(criteriaIds, flags, offset, count, users);
        Utils.log(users.toString());
        return users;
    }
    catch (SiatelException e) {
        StringBuffer sb = new StringBuffer();
        sb
            .append("<ERROR> User list query failed")
            .append(" - e.getMessage() : ")
            .append(e.getMessage());
        Utils.log(sb.toString());
        throw new Exception(sb.toString());
    }
}
```

Create a new user

The code below demonstrates how to create a new object; this example is still about users, but the same technique applies to any object.

```
public static User doNewUser(String login, String password,
        String realName, String description, int privilege) {
    User aNewUser = null;
    try {
        aNewUser = (User) SiatelFactory.newObject(Prefix.USER);
        aNewUser.setLabel(login);
        aNewUser.setPass(password);
        aNewUser.setRealName(realName);
        aNewUser.setDescription(description);
```

```
        aNewUser.setPrivileges(privilege);

        instance.client.createUser(aNewUser);
        Utils.log(aNewUser.toString());
    }
    catch(DeniedSecurityException dse) {
        StringBuffer sb = new StringBuffer();
        sb
            .append("<ERROR> User [")
            .append(instance.getUserNameFromCache())
            .append("] cannot create another user.");
        Utils.log(sb.toString());
    }
    catch(Exception e){
        StringBuffer sb = new StringBuffer();
        sb
            .append("<ERROR> Create user [")
            .append(login)
            .append("] command failed");
        Utils.log(sb.toString());
    }

    return aNewUser;
}
```

ASSIGN A USER TO A GROUP

The code below demonstrates how to assign a `User` to a `Group`. The same technique is used for any type of assignment available in Gargantua (for instance, user to roles, groups to databases, etc.).

```
public static void doAssignUserToGroup(String idGrp, String idUsr) {
    try {
        Assignment userToGroup = instance.client.createGroupUserAssig(idGrp, idUsr);
        Utils.log(userToGroup.toString());
    }
    catch (SiatelException e) {
        StringBuffer sb = new StringBuffer();
        sb
            .append("<ERROR> Assign user [")
            .append(idUsr)
            .append("] to group [")
            .append(idGrp).append("] failed.");
        Utils.log(sb.toString());
        e.printStackTrace();
```

```
    }
}
```

COMPLETE TOOLBOX CLASS SOURCE CODE

Now that we have seen differents bits of the Toolbox class, here is the complete listing of the class.

```
package utils.connecteursJava

import java.util.ArrayList;
import java.util.Collections;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

import utils.common.ConnectionInfos;
import utils.common.Utils;
import utils.connecteursJava.Toolbox;

import com.siatel.error.DeniedSecurityException;
import com.siatel.error.NotFoundPersistenceException;
import com.siatel.error.SiatelException;
import com.siatel.framework.web.AuthContext;
import com.siatel.proxy.http.SiatelClient;
import com.siatel.defs.Identified;
import com.siatel.domain.*;
import com.siatel.dto.DTO;
import com.siatel.dto.DTOFactory;

public final class Toolbox {
    private static volatile Toolbox instance = null;
    private SiatelClient client = null;

    // keep a connections cache
    private Map<String, ConnectionInfos>
        connectionsMap = Collections.synchronizedMap(
            new HashMap<String, ConnectionInfos>());

    private Toolbox(String urlServer) throws Exception{
        try {
            this.client = new SiatelClient(urlServer);
        } catch (Exception e) {
            StringBuffer sb = new StringBuffer();
            sb
```

```
                .append("<ERROR> The SIATEL client connector is not initialized.")
                .append("[url : ")
                .append(urlServer)
                .append(" ] ")
                .append(e.getMessage());
            Utils.log(sb.toString());
            throw new Exception(sb.toString());
        }
    }

    public static void initToolbox(String urlServer)throws Exception {
        Utils.log("------ Gargantua Java connector demo ------");
        if(Toolbox.instance == null){
            synchronized(Toolbox.class) {
                if (Toolbox.instance == null) {
                    Toolbox.instance = new Toolbox(urlServer);
                }
            }
        }
    }

    private void addToCache(String conn, String url, String userName) {
        if(!instance.connectionsMap.containsKey(conn)){
            ConnectionInfos infos = new ConnectionInfos(url, userName);
            instance.connectionsMap.put(conn, infos);
        }
    }

    private ConnectionInfos getFromCache(String conn) throws Exception {
        if(instance.connectionsMap.containsKey(conn)){
            return instance.connectionsMap.get(conn);
        }else{
            throw new Exception("The connection cache does not " +
                "hold the specified connection (" + conn + ")");
        }
    }

    private String getUserNameFromCache() {
        String conn = instance.client.getConn();
        String userName = "";
        ConnectionInfos infos;
        try {
            infos = instance.getFromCache(conn);
            userName = infos.getUserName();
        } catch (Exception e) {
            e.printStackTrace();
        }
```

```
        return userName;
    }

    private void removeFromCache(String conn){
        if(instance.connectionsMap.containsKey(conn)){
            instance.connectionsMap.remove(conn);
        }
    }
    public static void doListExtendedProperty(Identified obj) {
        List criteriaIds = new ArrayList();
        criteriaIds.add(obj.getId());
        long flags = Flag.VALUE;
        Integer offset = 0;
        Integer count = -1;
        List assignments = new ArrayList();

        try {
            instance.client.enumObjectExtendedPropertyAssignmentsForObject(
                obj, criteriaIds, flags, offset, count, assignments);
            Utils.log(assignments.toString());
            Utils.log("\n..... DETAILS .....");

            for(ObjectExtendedProperty oep : assignments) {
                Utils.log("[" + oep.getIdExtendedProperty() + " : " +
                    oep.getExtendedPropertyValue().toString() + "]");
            }

        }
        catch (SiatelException e) {
            e.printStackTrace();
        }
    }

    // --- Informations about server
    public static void aboutServer() throws Exception {
        try {
            StringBuffer sb = new StringBuffer();
            DTO dtoIn = DTOFactory.newDTO();

            dtoIn.put(Constants.SERVER_VERSION_MAJOR, "");
            dtoIn.put(Constants.SERVER_VERSION_MINOR, "");
            dtoIn.put(Constants.SERVER_VERSION_REVISION, "");

            DTO dtoOut = instance.client.getServerInfo(dtoIn);

            // server url
            sb
```

```
                .append("\n")
                .append("Url du serveur = ")
                .append(instance.client.getServerUrl());

            // server version
            sb
                .append("\n")
                .append(" -> Version majeur = ")
                .append(dtoOut.get(Constants.SERVER_VERSION_MAJOR))
                .append("\n")
                .append(" -> Version mineur = ")
                .append(dtoOut.get(Constants.SERVER_VERSION_MINOR))
                .append("\n")
                .append(" -> Revision = ")
                .append(dtoOut.get(Constants.SERVER_VERSION_REVISION));
            Utils.log(sb.toString());
        }
        catch (SiatelException e) {
            throw new Exception("Echec de lecture de la version " +
                "du serveur GED. \n détails : "+e.getMessage());
        }
    }

    // --- Handle authentification
    public static void doLogin(
            String username, String password) throws Exception {
        String conn = "";
        String urlServer = instance.client.getServerUrl();
        StringBuffer sb = new StringBuffer();

        try
        {
            instance.client.login(username, password, AuthContext.RUNTIME);
            conn = instance.client.getConn();
            instance.addToCache(conn, urlServer, username);

            sb
                .append("Cennexion etablie / ")
                .append(conn)
                .append(" / serveur [")
                .append(urlServer)
                .append("] / user [")
                .append(username)
                .append("]");
            Utils.log(sb.toString());
        }
        catch (Exception e) {
```

```
            sb
                .append(" Tentative de connexion ")
                .append(" / serveur [")
                .append(urlServer)
                .append("] / user [")
                .append(username).append("]");
            Utils.log(sb.toString());
            sb
                .append(" - e.getMessage : ")
                .append(e.getMessage());
            throw new Exception(sb.toString());
        }
    }

    public static void doLogout() throws Exception {
        String urlserver =  instance.client.getServerUrl();
        String username = "";
        String conn = instance.client.getConn();
        StringBuffer sb = new StringBuffer();

        try {
            instance.client.logout();

            ConnectionInfos infos = instance.getFromCache(conn);
            username = infos.getUserName();
            sb
                .append("Deconnexion / ")
                .append(conn)
                .append(" / serveur [")
                .append(urlserver)
                .append("] / user [")
                .append(username)
                .append("]");
            Utils.log(sb.toString());

            instance.removeFromCache(conn);
        }
        catch (SiatelException e) {
            sb
                .append(" Tentative de déconnexion / ")
                .append(conn)
                .append(" / serveur [")
                .append(urlserver)
                .append("] / user [")
                .append(username)
                .append("]");
            Utils.log(sb.toString());
```

```
            sb
                .append(" - e.getMessage : ")
                .append(e.getMessage());
            throw new Exception(sb.toString());
        }
    }

    // --- Handle User
    public static User doReadUser(String label) throws Exception {
        User aUser = (User) SiatelFactory.newObject(Prefix.USER);
        aUser.setLabel(label);

        try {
            instance.client.readUser(aUser, Flag.USER_ALL);
            Utils.log(aUser.toString());
            return aUser;
        }
        catch(NotFoundPersistenceException nfp) {
            StringBuffer sb = new StringBuffer();
            sb
                .append(" Tentative de consultation d'utilisateur [")
                .append(label)
                .append("] non existant.");
            Utils.log(sb.toString());
            throw new Exception(sb.toString());
        }
        catch (SiatelException e) {
            StringBuffer sb = new StringBuffer();
            sb
                .append(" Tentative de consultation de l'utilisateur [")
                .append(label)
                .append("] a échoué.")
                .append(" - e.getMessage() : ")
                .append(e.getMessage());
            Utils.log(sb.toString());
            throw new Exception(sb.toString());
        }
    }

    public static void doUpdateUser(User aUser) throws Exception {
        try {
            instance.client.updateUser(aUser, Flag.USER_UPDATE_ALL, true);
            Utils.log(aUser.toString());
        }
        catch(NotFoundPersistenceException nfp) {
            StringBuffer sb = new StringBuffer();
            sb
```

```
                .append(" Tentative de mise à jour d'utilisateur [")
                .append(aUser.getLabel())
                .append("] non existant.");
            Utils.log(sb.toString());
            throw new Exception(sb.toString());
        }
        catch (SiatelException e) {
            StringBuffer sb = new StringBuffer();
            sb
                .append(" Tentative de mise à jour de l'utilisateur [")
                .append(aUser.getLabel())
                .append("] a échoué.")
                .append(" - e.getMessage() : ")
                .append(e.getMessage());
            Utils.log(sb.toString());
            throw new Exception(sb.toString());
        }
    }

    public static List doListUsers() throws Exception {
        List criteriaIds = new ArrayList();
        long flags = Flag.ID | Flag.LABEL | Flag.DESCRIPTION | Flag.REAL_NAME;
        Integer offset = 0;
        Integer count = -1;
        List users = new ArrayList();

        try {
            instance.client.enumUsers(criteriaIds, flags, offset, count, users);
            Utils.log(users.toString());
            return users;
        }
        catch (SiatelException e) {
            StringBuffer sb = new StringBuffer();
            sb
                .append(" Tentative de lister les utilisateurs a échoué.")
                .append(" - e.getMessage() : ")
                .append(e.getMessage());
            Utils.log(sb.toString());
            throw new Exception(sb.toString());
        }
    }

    public static User doNewUser(
            String login, String password, String realName,
            String description, int privilege) {
        User aNewUser = null;
        try {
```

```
            aNewUser = (User) SiatelFactory.newObject(Prefix.USER);
            aNewUser.setLabel(login);
            aNewUser.setPass(password);
            aNewUser.setRealName(realName);
            aNewUser.setDescription(description);
            aNewUser.setPrivileges(privilege);

            instance.client.createUser(aNewUser);
            Utils.log(aNewUser.toString());
        }
        catch(DeniedSecurityException dse){
            StringBuffer sb = new StringBuffer();
            sb
                .append(" L'utilisateur [")
                .append(instance.getUserNameFromCache())
                .append("] n'a pas le droit de créer d'autres utilisateurs.");
            Utils.log(sb.toString());
        }
        catch(Exception e) {
            StringBuffer sb = new StringBuffer();
            sb
                .append(" Tentative de création de l'utilisateur [")
                .append(login)
                .append("]");
            Utils.log(sb.toString());
        }

        return aNewUser;
    }

    // --- Handle Group
    public static void doReadGroup(String label) {
        Group aGroup = (Group) SiatelFactory.newObject(Prefix.GROUP);
        aGroup.setLabel(label);

        try {
            instance.client.readGroup(aGroup, Flag.GROUP_ALL);
            Utils.log(aGroup.toString());
        }
        catch(NotFoundPersistenceException nfp) {
            StringBuffer sb = new StringBuffer();
            sb
                .append(" Tentative de consultation du groupe [")
                .append(label)
                .append("] non existant.");
            Utils.log(sb.toString());
        }
```

```
        catch (SiatelException e) {
            StringBuffer sb = new StringBuffer();
            sb
                .append(" Tentative de consultation du groupe [")
                .append(label)
                .append("] a échoué.")
                .append(" - e.getMessage() : ")
                .append(e.getMessage());
            Utils.log(sb.toString());
        }
    }

    public static List doListGroups() throws Exception {
        List criteriaIds = new ArrayList();
        long flags = Flag.ID | Flag.LABEL | Flag.DESCRIPTION;
        Integer offset = 0;
        Integer count = -1;
        List groups = new ArrayList();

        try {
            instance.client.enumGroups(criteriaIds, flags, offset, count, groups);
            Utils.log(groups.toString());
            return groups;
        }
        catch (SiatelException e) {
            StringBuffer sb = new StringBuffer();
            sb
                .append(" Tentative de lister les groupes a échoué.")
                .append(" - e.getMessage() : ")
                .append(e.getMessage());
            Utils.log(sb.toString());
            throw new Exception(sb.toString());
        }
    }

    public static Group doNewGroup(
            String label, String description, int privilege) {
        Group aNewGroup = null;
        try {
            aNewGroup = (Group) SiatelFactory.newObject(Prefix.GROUP);
            aNewGroup.setLabel(label);
            aNewGroup.setDescription(description);
            aNewGroup.setPrivileges(privilege);

            instance.client.createGroup(aNewGroup);
            Utils.log(aNewGroup.toString());
        }
```

```
        catch(DeniedSecurityException dse) {
            StringBuffer sb = new StringBuffer();
            sb
                .append(" L'utilisateur [")
                .append(instance.getUserNameFromCache())
                .append("] n'a pas le droit de créer des groupes.");
            Utils.log(sb.toString());
        }
        catch(Exception e) {
            StringBuffer sb = new StringBuffer();
            sb
                .append(" Tentative de création du groupe [")
                .append(label)
                .append("]");
            Utils.log(sb.toString());
        }

        return aNewGroup;
    }

    public static void doAssignUserToGroup(String idGrp, String idUsr) {
        try {
            Assignment userToGroup =
                instance.client.createGroupUserAssig(idGrp, idUsr);
            Utils.log(userToGroup.toString());
        }
        catch (SiatelException e) {
            StringBuffer sb = new StringBuffer();
            sb
                .append(" Tentative d'affectation de l'utilisateur [")
                .append(idUsr)
                .append("] vers le groupe [")
                .append(idGrp)
                .append("]");
            Utils.log(sb.toString());
            e.printStackTrace();
        }
    }

    // --- Handle database
    public static List doListDatabases() throws Exception {
        List criteriaIds = new ArrayList();
        long flags = Flag.ID | Flag.LABEL | Flag.DESCRIPTION_URL;
        Integer offset = 0;
        Integer count = -1;
        List databases = new ArrayList();
```

```
        try {
            instance.client.enumDatabases(
                criteriaIds, flags, offset, count, databases);
            Utils.log(databases.toString());
            return databases;
        }
        catch(SiatelException ex) {
            StringBuffer sb = new StringBuffer();
            sb
                .append(" Tentative de lister les bases a échoué.")
                .append(" - e.getMessage() : ")
                .append(ex.getMessage());
            Utils.log(sb.toString());
            throw new Exception(sb.toString());
        }
    }
}
```

Using the Toolbox class

Now that we have created a wrapper class for the `SiatelClient` object, let's use it.

```
package exemples.connecteursJava;

import utils.connecteursJava.Toolbox;

public class AboutServer {

    // the default local server connection
    public static final String SERVER_URL = "http://localhost:8080/siatel-web/";

    /**
     * @param args
     */
    public static void main(String[] args) throws Exception {
        try {
            Toolbox.initToolbox(SERVER_URL);
            Toolbox.aboutServer();
        }
        catch(Exception ex) {
        }
        finally {
        }
    }
```

```
}
```

You can now run this code as a Java application from your development environment.

GARGANTUA API FOR FORMS

The Gargantua 8 server provides a set of JavaScript functions that allow you to modify the contents of a form or to perform a number of actions.

SCRIPT.NAME

```
//@script.name " <script_name>"
```

Allows to include a form script within another one, created in the **Scripts** tab.

For instance, we created a script named "library.script.message" in the Scripts tab. To link this script to the form script, we will enter the following code:

```
//@script.name "library.scripts.message".
```

SCRIPT.NO_CHECK

```
//@script.no_check
```

Requests the compiler to ignore compiler errors. The compiler uses a very strict grammar; some code, even if correct, may raise errors due to this.

When using this directive, please make sure you validate the script yourself or be prepared for u7nexpected errors.

JavaScript functions for form elements

gfxGetAttrIdByLabel

```
function gfxGetAttrIdByLabel(label)
```

Retrieves the given attribute's ID based on the label.

Parameters

label

The label of the attribute whose id must be returned.

Return value

ID of the given attribute.

Exceptions

n/a

Example

```
window.alert(gfxGetAttrIdByLabel("label"));
```

will display

```
DFATxxxx00000001000000000000000000000000
```

gfxGetAttrLabelById

```
function gfxGetAttrLabelById(attributeId)
```

Retrieves the given attribute's label using it's ID.

Parameters

attributeId

The ID of the attribute whose label must be returned.

Return value

The label of the given attribute.

Exceptions

n/a

Example

```
window.alert(gfxGetAttrLabelById("DFATxxxx00000001000000000000000000000000"));
```

will display

```
label
```

getElementValue

```
function getElementValue(elementID)
```

Returns the value of the specified element.

This API function has been replaced by the gfxVal function; please use the new function whenever possible.

Parameters

elementID

The ID of the attribute whose value must be returned.

Return value

Value of the specified attribute.

Exceptions

n/a

Example

```
window.alert(getElementValue("DFATxxxx00000001000000000000000000000000"));
```

will display the current value of the label attribute.

getElementKey

```
function getElementKey(elementID)
```

Returns the key of the specified element if it is a keyed (translated) list or `null` otherwise.

Parameters

elementID

The ID of the attribute whose key must be returned.

Return value

Key of the specified attribute. If the attribute is not a keyed list, the current element value is returned.

Exceptions

n/a

Example

Let's assume we have the following two lists and two attributes of these types.

Simple list :

```
1
   1.1
   1.2
   1.3
2
   2.1
   2.2
   2.3
3
4
5
```

Keyed (translated) list :

```
1              one                 premier
   1.1            one.one             premier.premier
   1.2            one.two             premier.deuxième
   1.3            one.three           premier.troisième
2              two                 deuxième
   2.1            two.one             deuxième.premier
   2.2            two.two             deuxième.deuxième
   2.3            two.three           deuxième.troisième
3              three               troisième
4              four                quatrième
```

```
5               five                    cinquième
```

When the first value is selected, then

```
window.alert(getElementKey("DFAT00070000066000000000000000000000000"));
```

will return `1` for the first attribute and `1` for the second attribute, while

```
window.alert(getElementValue("DFAT00070000066000000000000000000000000"));
```

will return `1` for the first attribute and `premier` for the second attribute.

GFXVAL

```
function gfxVal(element [, value])
```

`gfxVal` will either read or write the given attribute's value depeding on the presence of the `value` parameter

Parameters

element

The label of the attribute whose value must be returned or set.

value

(optional) The value to be set

Return value

Value of the specified attribute or undefined if the set variant is used.

Exceptions

n/a

Example

```
gfxVal("label", "test");

window.alert(gfxVal("label"));
```

will display `test`.

gfxEnable

```
function gfxEnable(element)
```

Enables the control for the given attribute.

Parameters

element

The label of the attribute whose control must be enabled

Return value

None

Exceptions

n/a

Example

```
gfxEnable('liste simple');
```

GFXDISABLE

```
function gfxDisable(element)
```

Disables the control for the given attribute.

Parameters

element

The label of the attribute whose control must be disabled

Return value

None

Exceptions

n/a

Example

```
gfxDisable('liste simple');
```

GFXVISIBLE

```
function gfxVisible(element, visible)
```

Changes the visibility of the control for the given attribute.

Parameters

element

The label of the attribute whose control must be shown or hidden

visible

Boolean value to indicate visibility status

Return value

None

Exceptions

n/a

Example

```
gfxVisible('liste simple', false);
```

will hide the form control for the given attribute.

```
gfxVisible('liste simple', true);
```

will show the form control for the given attribute.

getElementValueEx

```
function getElementValueEx(elementID)
```

Returns the key of the specified element if it is a keyed (translated) list or the current value otherwise.

Parameters

elementID

The ID of the attribute whose key must be returned.

Return value

Key of the specified attribute. If the attribute is not a keyed list, the current element value is returned.

Exceptions

n/a

Example

Let's assume we have the following two lists and two attributes of these types.

Simple list :

```
1
   1.1
   1.2
   1.3
2
   2.1
   2.2
   2.3
3
4
5
```

Keyed (translated) list :

```
1              one                 premier
   1.1            one.one             premier.premier
   1.2            one.two             premier.deuxième
   1.3            one.three           premier.troisième
2              two                 deuxième
   2.1            two.one             deuxième.premier
   2.2            two.two             deuxième.deuxième
   2.3            two.three           deuxième.troisième
3              three               troisième
4              four                quatrième
```

```
5               five                    cinquième
```

When the first value is selected, then

```
window.alert(getElementValueEx("DFAT00070000066000000000000000000000000"));
```

will return `1` for the first attribute and `1` for the second attribute.

setElementValue

```
function setElementValue(elementID, value)
```

Changes the value of the specified element.

This API function has been replaced by the gfxVal function; please use the new function whenever possible.

Parameters

elementID

The ID of the attribute whose value must be changed.

value

New value.

Return value

None.

Example

```
setElementValue("DFATxxxx00000001000000000000000000000000", "test"));
```

will set the label attribute to the value `test`.

setElementKey

```
function setElementKey(elementID, value)
```

Sets the value of an keyed (translated) list attribute to the given key

Parameters

elementID

The ID of the attribute whose value must be changed.

value

New value.

Return value

None.

Example

Let's assume we have the following keyed (translated) list and an attributea of these type.

```
1               one                 premier
   1.1             one.one             premier.premier
   1.2             one.two             premier.deuxième
   1.3             one.three           premier.troisième
2               two                 deuxième
   2.1             two.one             deuxième.premier
   2.2             two.two             deuxième.deuxième
   2.3             two.three           deuxième.troisième
3               three               troisième
4               four                quatrième
5               five                cinquième
```

When the following code is executed

```
setElementKey("DFAT00070000006600000000000000000000000000", "1");
```

The value `premier` will be selected in the form control.

getElementColor

```
function getElementColor(elementID)
```

Returns the color of the specified element.

Parameters

elementID

The ID of the attribute whose value must be returned.

Return value

Color of the specified attribute or `undefined` if there is no custom color set for the control.

Example

```
window.alert(getElementColor("DFAT00070000006600000000000000000000000000"));
```

will display the color of the element or `undefined` if the color is not set.

setElementValueEx

```
function setElementValueEx(elementID, value)
```

Changes the value of the specified element using either the value or the key depending if it is a keyed (translated) list or not.

This API function has been replaced by the gfxVal function; please use the new function whenever possible.

Parameters

elementID

The ID of the attribute whose value must be changed.

value

New key or value.

Return value

None

Example

Let's assume we have the following keyed (translated) list and an attribute of this type.

```
1               one                 premier
   1.1             one.one             premier.premier
   1.2             one.two             premier.deuxième
   1.3             one.three           premier.troisième
2               two                 deuxième
   2.1             two.one             deuxième.premier
   2.2             two.two             deuxième.deuxième
   2.3             two.three           deuxième.troisième
3               three               troisième
4               four                quatrième
5               five                cinquième
```

```
setElementValueEx("DFAT00070000006700000000000000000000000000", "1");
```

will set the attribute to the value `one`.

setElementColor

```
function setElementColor(elementID, color)
```

Changes the color of the specified element.

Parameters

elementID

The ID of the attribute whose color must be changed.

color

New color.

Return value

None.

Example

```
setElementColor("DFAT0007000006600000000000000000000000", "#ff0000");
```

After executing the previous line, the following line

```
window.alert(getElementColor("DFAT0007000006600000000000000000000000"));
```

will display `#ff0000`

setElementColor2

```
function setElementColor2(elementID)
```

This API function has been deprecated in Gargantua 8. Please replace it when porting existing forms from older Gargantua versions.

Parameters

elementID

The ID of the attribute whose color must be changed.

color

New color.

all

Boolean indicating whether to change the background for all radio inputs in case of a radio element

Return value

None

setElementTitle

```
function setElementTitle(elementID, title)
```

Modifies the tooltip of the specified element.

Parameters

elementID

The ID of the attribute whose tooltip must be modified.

title

New tooltip.

Return value

None

Example

```
setElementTitle("DFAT000700000066000000000000000000000000", "TEST !");
```

setElementVisible

```
function setElementVisible(elementID, value)
```

Changes the visibility of the specified element (visible or invisible).

This API function has been replaced by the gfxVisible function; please use the new function whenever possible.

Parameters

elementID

The ID of the attribute whose visibility needs to be changed.

value

Visibility indicator (”true” or “false”).

Return value

None

Example

```
setElementVisible("DFAT0007000000660000000000000000000000000", false);
```

setElementVisible2

```
function setElementVisible2(elementID)
```

This API function has been deprecated in Gargantua 8. Please replace it when porting existing forms from older Gargantua versions.

Parameters

elementID

The ID of the attribute whose visibility needs to be changed.

Return value

None

setElementEdit

```
function setElementEdit(elementID, value)
```

Modifies the edit state of the specified element (editable or read-only).

This API function has been replaced by the gfxEnable and gfxDisable functions; please use the new functions whenever possible.

Parameters

elementID

The ID of the attribute whose edit state must be changed.

value

Edit state indicator (”true” or “false”).

Return value

None.

Example

```
setElementEdit('DFAT0007000000010000000000000000000000000', false);
```

setElementEdit2

```
function setElementEdit2(elementID)
```

This API function has been deprecated in Gargantua 8. Please replace it when porting existing forms from older Gargantua versions.

Parameters

elementID

The ID of the attribute whose edit state must be changed.

value

Edit state indicator (”true” or “false”).

Return value

None

Example

```
setElementEdit2('DFAT00070000001000000000000000000000000', false);
```

setElementEnabled

```
function setElementEnabled (elementID, bValue)
```

Modifies the enable state of the specified element (enabled or disabled).

This API function has been replaced by the gfxEnable and gfxDisable functions; please use the new functions whenever possible.

Parameters

elementID

The ID of the attribute whose enable state must be changed.

value

Enable state indicator (”true” or “false”).

Return value

None.

Example

```
setElementEnabled('DFAT0007000000010000000000000000000000000', false);
```

isElementEnabled

```
function isElementEnabled (elementID)
```

Returns the enable state of the specified element (enabled or disabled).

Parameters

elementID

The ID of the attribute whose enable state must be returned.

Return value

The enable state of the attribute.

Example

```
window.alert(isElementEnabled('DFAT0007000000660000000000000000000000000'));
```

Will display either `true` or `false`.

disabledEditor

```
function disabledEditor(elementID)
```

This API function has been deprecated in Gargantua 8. Please replace it when porting existing forms from older Gargantua versions.

Parameters

elementID

The ID of the attribute whose edit state must be changed.

Return value

None.

Example

```
disabledEditor('DFAT0007000000010000000000000000000000000');
```

enabledEditor

```
function enabledEditor(elementID)
```

This API function has been deprecated in Gargantua 8. Please replace it when porting existing forms from older Gargantua versions.

Parameters

elementID

The ID of the attribute whose edit state must be changed.

Return value

None.

Example

```
enabledEditor('DFAT0007000000010000000000000000000000000');
```

FORM_GETVALUES

```
function form_getValues()
```

Retrieves all the form values as a JSON object. The returned value also includes hidden attributes.

PARAMETERS

None.

RETURN VALUE

A JSON object with all form values.

EXAMPLE

```
window.alert(JSON.stringify(form_getValues()));
```

will display an output similar to this :

```
{
    "formid":"DFFR0007000000070000000000000000000000000",
    "pageid":"Défaut","parentid":"DATA00070000000000000000000000000000000000",
    "objectid":"",
    "command":"view",
    "target":"",
    "multi":"0",
    "search":"0",
    "param1":"",
    "param2":"",
    "param3":"",
    "param4":"",
    "param5":"",
    "DFAT0007000000010000000000000000000000000":"",
    "DFAT0007000000660000000000000000000000000":"",
    "DFAT0007000000670000000000000000000000000":""
}
```

form_getAttributeType

```
function form_getAttributeType(elementID)
```

Retrieves the numeric attribute type for the specified attribute.

Parameters

elementID

The ID of the attribute for which the numeric type ID must be retrieved

Return value

The numeric representation of the type ID for the specified attribute.

Example

```
window.alert(form_getAttributeType('DFAT000700000066000000000000000000000000'));
```

will display `100`.

form_getAttributeTypeId

```
function form_getAttributeTypeId(elementID)
```

Retrieves the attribute type for the specified attribute.

Parameters

elementID

The ID of the attribute for which the type ID must be retrieved

Return value

The string representation of the type ID for the specified attribute.

Example

```
window.alert(form_getAttributeTypeId('DFAT0007000000660000000000000000000000000'));
```

will display `DFTY0007000000640000000000000000000000000`.

form_getPage

```
form_getPage()
```

Retrieves the name of the current form page.

Parameters

None

Return value

The name of the form page.

Example

```
var page = form_getPage();
```

Miscellaneous JavaScript functions

form_getVersion

```
function form_getVersion()
```

Retrieves the version of the Gargantua server.

Parameters

None

Return value

A string representation of the Gargantua 8 server version.

Example

```
window.alert(form_getVersion());
```

will display a string like `8.0.7955`.

FORM_LANG

```
function form_lang()
```

Returns the language code of the user interface.

Parameters

None.

Return value

Language code. fr: French; ar: Arabic; en: English; es: Spanish; ru: Russian; ro: Romanian; hu: Hungarian.

The Gargantua 8 does no longer implement arabic translations, so the "ar" return value is no longer used, even for forms that have been migrated from older versions of Gargantua.

Example

form_getDisplayMode

```
function form_getDisplayMode()
```

Retrieves the current display mode of the form.

Parameters

None.

Return value

The current display mode of the form, which can be one of the following values (both symbolic and numeric constants can be used):

```
0 = DISPLAY_NORMAL
1 = DISPLAY_READONLY
2 = DISPLAY_DISABLEDINPUTS
4 = DISPLAY_DISABLEDBUTTONS
```

Example

```
window.alert(form_getDisplayMode());
```

form_enableGroup

```
function form_enableGroup(group, enabled);
```

Enables or disables a group of controls at once.

The group property can be set for individual attributes or other form controls in the form designer.

The effect is the same as enabling or disabling individual controls by using gfxEnable and gfxDisable.

Parameters

group

The name of the group.

enabled

Enabled indicator (”true” or “false”).

Return value

None.

Example

```
form_enableGroup("myGroup", false);
```

form_setGroupVisible

```
function form_setGroupVisible(group, visible);
```

Sets the visibility of a group of controls at once.

The group property can be set for individual attributes or other form controls in the form designer.

The effect is the same as showing or hiding individual controls by using gfxVisible.

Parameters

group

The name of the group.

visible

Visibility indicator (”true” or “false”).

Return value

None.

Example

```
form_setGroupVisible("myGroup", false);
```

form_expandSection

```
function form_expandSection(id);
```

Expands a section in grid layout mode.

Parameters

id

The ID of the section.

Return value

None.

Example

```
form_expandSection("theSection");
```

form_collapseSection

```
function form_collapseSection(id);
```

Collapses a section in grid layout mode.

Parameters

id

The ID of the section.

Return value

None.

Example

```
form_collapseSection("theSection");
```

form_toggleSection

```
function form_toggleSection(id);
```

Toggles (expands or collapses) a section in grid layout mode.

Parameters

id

The ID of the section.

Return value

None.

Example

```
form_toggleSection("theSection");
```

form_toggleAllSections

```
form_toggleAllSections(group, showOrHide)
```

Toggles (expands or collapses) all sections (or the sections having the same group) in grid layout mode.

The group property can be set for sections in the form designer.

Parameters

group

(optional) The name of the group

showOrHide

Visibility indicator (”true” or “false”)

Return value

None

Example

```
form_toggleAllSections("myGroup", false);
form_toggleAllSections(null, true);
```

form_isSectionExpanded

```
function form_isSectionExpanded(id);
```

Retrieves the expanded status of the specified section.

Parameters

id

The ID of the section.

Return value

The expanded status of the specified section.

Example

```
var expanded = form_isSectionExpanded("mySection");
```

form_addSectionIcon

```
function form_addSectionIcon(id, options)
```

Function that adds an icon to the form section identified by the given id.

Parameters

id

The ID of the section.

options

A JSON object containing the following key-pairs:

- id - the icon id
- src - the icon source
- position - `start | end`
- className - a class to add to the icon
- title - the icon tooltip
- hidden - boolean indicating if the icon should not be visible
- width - the icon width
- height - the icon height.

Return value

None.

Example

```
form_addSectionIcon('section',{
    id : 'icon',
    src : '/images/plugins/iconpack/svg/bubble.svg#bubble',
    position : 'end',
    hidden : false,
    width : '3rem',
    height : '3rem'
});

form_setSectionIconStyle('icon3', {
    'fill' : 'green'
});
```

form_toggleSectionIcon

```
function form_toggleSectionIcon(id, showOrHide)
```

Function that toggles the visibility of the section icon.

Parameters

id

The ID of the section icon.

showOrHide

Boolean indicating if the icon should be visible or not.

Return value

None.

Example

```
form_toggleSectionIcon('icon');
```

form_removeSectionIcon

```
function form_removeSectionIcon(id)
```

Function that removes the section icon.

Parameters

id

The ID of the section icon.

Return value

None.

Example

```
form_removeSectionIcon('icon');
```

form_setSectionIconStyle

```
function form_setSectionIconStyle(id, cssData)
```

Function that sets the section icon's style.

Parameters

id

The ID of the section icon.

cssData

A JSON object containing the styles to be applied to the icon.

Return value

None.

Example

```
form_addSectionIcon('section',{
    id : 'icon',
    src : '/images/plugins/iconpack/svg/bubble.svg#bubble',
    position : 'end',
    hidden : false,
    width : '3rem',
    height : '3rem'
});

form_setSectionIconStyle('icon3', {
    'fill' : 'green'
});
```

FORM_STICKYSECTION

```
function form_stickySection(id [, options])
```

Positions a section as sticky at the top of the form.

Parameters

id

The section ID.

options

An object containing the following possible options:

- `top` or `stickyTop` parameter indicating the viewport top position when the section should become sticky. By default this is 0.
- `zIndex` or `stickyZIndex` parameter indicating the z-index property for the section. By default this is 10.

Return value

None.

Example

```
form_stickySection('s1');
```

or

```
form_stickySection('s1', { zIndex: 25 });
```

FORM_SECTIONSTOTABS

```
function form_sectionsToTabs(ids, tabId [, options])
```

Transforms the sections identified by the given IDs array into tabs, each section becoming a tab. The section headers will be removed.

Parameters

ids

An array of section IDs.

tabId

The ID of the newly created tabs wrapper.

options

An object containing the following possible options:

- `border` (true/false) parameter indicating weather the tab content should or not have a visible border. By default this is `true`.
- `sticky` (true/false) parameter indicating weather the tab navigation should be rendered as sticky. By default this is `false`.
- `top` or `stickyTop` parameter indicating the viewport top position when the tab navigation should become sticky. Should be specified only in combination with sticky true. By default this is 0.
- `zIndex` or `stickyZIndex` parameter indicating the z-index property for the tab navigation. Should be specified only in combination with sticky true. By default this is 10.
- `height` parameter allowing you to control the height of the tab content. When the content exceeds this height, the content area becomes scrollable.
- `minHeight` parameter allowing you to control the minimum height of the tab content.
- `maxHeight` parameter allowing you to control the maximum height of the tab content. When the content exceeds this height, the content area becomes scrollable.

Return value

None.

Example

```
form_sectionsToTabs(['s1', 's2', 's3', 's4'], 'mytab');
```

or

```
form_sectionsToTabs(['s1', 's2', 's3', 's4'], 'mytab', { border: false });
```

FORM_SECTIONGROUPSTOTABS

```
form_sectionGroupsToTabs(group_names_array, tab_id [, options])
```

Transforms the sections that have the groups specified in the groups array into tabs. This way a tab can contain multiple sections. The group names array should be unique (no duplicates). If any of the sections in a group has the visible mandatory marker then the tab will also have one.

Parameters

group_names_array

An array of group names.

tabId

The ID of the newly created tabs wrapper.

options

An object containing the following possible options:

- `border` (true/false) parameter indicating weather the tab content should or not have a visible border. By default this is `true`.
- `sticky` (true/false) parameter indicating weather the tab navigation should be rendered as sticky. By default this is `false`.
- `top` or `stickyTop` parameter indicating the viewport top position when the tab navigation should become sticky. Should be specified only in combination with sticky true. By default this is 0.
- `zIndex` or `stickyZIndex` parameter indicating the z-index property for the tab navigation. Should be specified only in combination with sticky true. By default this is 10.
- `height` parameter allowing you to control the height of the tab content. When the content exceeds this height, the content area becomes scrollable.
- `minHeight` parameter allowing you to control the minimum height of the tab content.
- `maxHeight` parameter allowing you to control the maximum height of the tab content. When the content exceeds this height, the content area becomes scrollable.
- `tabs` (array of objects) parameter provides custom options for each tab:
  - `id` - the id of the tab (if not provided it will be generated)
  - `name` - the name/label of the tab (if not provided it will be copied from the first section)
  - `onlyContent` (true/false) - indicates weather only the content of the section or the entire section (with header) should be added in the tab content (by default this is `false`).

Return value

None.

Example

```
form_sectionGroupsToTabs(['g1', 'g2', 'g3'], 'mytab', { border: false });
```

or

```
form_sectionGroupsToTabs(['g1', 'g2', 'g3'], 'mytab', {
  border: true,
  sticky: true,
  maxHeight: '200px',
  tabs: [{
    id: 'tab1',
    name: 'First tab'
  }, {
    id: 'tab2',
    name: '<strong>Second tab</strong>',<
    onlyContent: true
  }, {
    id: 'tab3',
    name: '<strong>Third tab</strong>'
  }]
});
```

FORM_REFRESH

```
function form_refresh();
```

Refreshes the entire iframe that contains the form.

Parameters

None.

Return value

None.

Example

```
form_refresh();
```

form_escapeHtml

```
function form_escapeHtml(text);
```

Escapes a block of text according to HTML escaping rules.

Parameters

text

The text to escape.

Return value

The escaped text.

Example

```
window.alert(form_escapeHtml("\"this is a text");
```

will display

```
&quot;this is a &lt;bracketed&gt; text
```

form_refreshViewer

```
function form_refreshViewer();
```

Refreshes the viewer that may be displayed next to the form. This is useful when form actions trigger document changes and the document must be re-displayed to reflect those changes.

This API function has effect only in the inbox component, where a form may be displayed side by side with a viewer panel.

Parameters

None.

Return value

None.

Example

```
form_refreshViewer();
```

form_cleanupHtml

```
function form_cleanupHtml(text);
```

Removes <p> and <br> tags from the input text.

Also replaces any <br> tags by a hard line break (\r).

Parameters

text

The text to cleanup.

Return value

The cleaned-up text.

Example

```
window.alert(form_escapeHtml( "<p>this is a paragraph<br>and a line break</p>");
```

will display

```
this is a paragraph
and a line break
```

FORM_LOG

```
function form_log(text);
```

Outputs a log message to the browser console.

Parameters

text

The message to write to the console.

Return value

None.

Example

```
form_log("message");
```

FORM_INFO

```
function form_info(text);
```

Outputs an info message to the browser console.

Parameters

text

The message to write to the console.

Return value

None.

Example

```
form_info("message");
```

FORM_WARN

```
function form_warn(text);
```

Outputs a warning message to the browser console.

Parameters

text

The message to write to the console.

Return value

None.

Example

```
form_warn("message");
```

FORM_ERROR

```
function form_error(text);
```

Outputs an error message to the browser console.

Parameters

text

The message to write to the console.

Return value

None.

Example

```
form_error("message");
```

FORM_WAITSTART

```
function form_waitstart(inForm);
```

Displays a ‘wait’ rotating glyph to indicate the application is busy.

Parameters

inForm

Indicates whether the wait rotating glyph should block the form or the entire view.

Return value

None.

Example

```
form_waitstart(true);
```

FORM_WAITSTOP

```
function form_waitstop(inForm, immediately);
```

Closes the 'wait' rotating glyph used to indicate the application is busy.

Parameters

inForm

Indicates whether the wait rotating glyph is from the form or the entire view.

immediately

(optional) Indicates whether the unblock should be performed immediately or with the system predefined timeout.

Return value

None.

Example

```
form_waitstop(true);
```

form_formatDate

```
function form_formatDate(date);
```

Formats a date with the “dd/MM/yyyy” pattern.

Parameters

date

the date object to format

Return value

The formatted date.

Example

```
var date = new Date();
var formattedDate = form_formatDate(date); // something like "05/05/2024"
```

form_parseDate

```
function form_parseDate(date);
```

Parses a date formatted with the “dd/MM/yyyy” pattern and returns a valid date object.

Parameters

date

A string representation of a date in the “dd/MM/yyyy” pattern

Return value

A date object.

Example

```
var formattedDate = "05/05/2024";
var date = form_parseDate(formattedDate);
```

form_formatUserName

```
function form_formatUserName(userName, realName)
```

Formats the given user name and real name based on the siatel.config.user.name.display.format server option.

Parameters

userName

The user name

realName

The real name of the user

Return value

The formatted user name.

Example

```
var name = form_formatUserName("jeanw", "Jean Winters"); // can return something
like "jeanw (Jean Winters)"
```

Depending on the siatel.config.user.name.display.format server option the result will differ. In the example above the option is set to {{userName}} ({{realName}}).

form_hasDocuments

```
function form_hasDcuments(form, callback)
```

Checks to see if the process has documents or not that have the given form.

This API function has effect only in the inbox component.

Parameters

form

The name or the id of the form

callback

A callback function that receives the result as a parameter (optional)

Return value

If the callback option is set, then the operation will be asynchronous and the result will be returned through it, otherwise the operation is synchronous and the result will be returned by the function its self.

The result will be `true` or `false`.

Example

```
var hasDocs = form_hasDocuments("Default"); // sync call
form_hasDocuments("Default", function(hasDocs) {
    ...
}); // async call
```

form_getDocuments

```
function form_getDocuments(form, callback)
```

Returns the ids of the documents in a process that have the given form.

Parameters

form

The name or the id of the form

callback

A callback function that receives the result as a parameter (optional)

Return value

If the callback option is set, then the operation will be asynchronous and the result will be returned through it, otherwise the operation is synchronous and the result will be returned by the function its self.

The result will be an array of ids.

Example

```
var ids = form_getDocuments("Default"); // sync call
form_getDocuments("Default", function(ids) {
    ...
}); // async call
```

form_filterList

```
function form_filterList(id, arguments)
```

Filters the options of a list element.

Parameters

id

The id of the list element

arguments

There are 3 possibilities:

- `regexp` - a regular expression either as object or string
- `callback` - a function callback that receives the current option and level as parameters and must return true if the option should be kept or false if it should be removed
- `searchOptions, contains` - an array of options and a boolean indicating whether to keep only the options contained in the searchOptions array (if `true`) or if to remove any of the options that are contained in the array (if `false`)

Return value

None

Example

```
form_filterList(id, /^test/); // keeps only options who's value starts with test and
removes all others
form_filterList(id, function (option, level) {
    return option.startsWith("test");
});
form_filterList(id, ["test element 1", "test element 2"], true); // keeps only the
options who's values appear in the search array
```

form_setDocTemplateOptions

```
function form_setDocTemplateOptions(options)
```

Specifies the options used by the document template selection dialog.

Parameters

options

an object that can contain the following options:

default

displays default templates; possible values: `0/1, true/false`

custom

displays custom templates(defined in administration console); possible values: `0/1, true/false`

plugin

displays plugin templates; possible values: `0/1, true/false`

tagNames

filter plugin templates by tag name; value: array of tag names

Return value

None.

Example

```
form_setDocTemplateOptions({ default: true, custom: 0, plugin: 1, tagNames: ['tag1',
'tag2'] });
```

JavaScript functions for customisizing the user interface

flashElement

```
function flashElement(elementID, color, count)
```

Makes the specified element flash. Used to draw the attention of a user.

Parameters

elementID

The ID of the attribute to be flashed.

color

The color used for flashing.

count

Number of flashes.

Return value

None.

Example

form_displayButtonBar

```
function form_displayButtonBar(bDisplay)
```

Shows/hides the toolbar. Useful if the forms contain custom buttons.

Parameters

bDisplay

Visibility indicator (”true” or “false”).

Return value

None.

Example

form_displayDocumentListToggle

This function has been deprecated; it exists only for compatibility reasons and has no effect. The user interface of Gargantua 8 no longer displays the document list at the same time as the task form.

```
function form_displayDocumentListToggle()
```

Shows/hides the container of the process documents list.

Parameters

None

Return value

None.

Example

form_displayDocumentList

This function has been deprecated; it exists only for compatibility reasons and has no effect. The user interface of Gargantua 8 no longer displays the document list at the same time as the task form.

```
function form_displayDocumentList(bDisplay)
```

Shows/hides the process documents list.

Parameters

bDisplay

Visibility indicator (”true” or “false”).

Return value

None.

Example

form_displaySelectorArea

```
function form_displaySelectorArea(bDisplay)
```

Shows/hides the form selection bar. Useful if changing the form is not required.

Parameters

bDisplay

Visibility indicator (”true” or “false”).

Return value

None.

Example

form_displayDefaultButtonBar

```
function form_displayDefaultButtonBar()
```

Displays the default buttons below the form; the default buttons depend on the context; for instance, when the form is displayed for a folder these buttons are different than the buttons displayed in the inbox.

Parameters

None

Return value

None.

Example

form_setCustomButtonEnabled

```
function form_setCustomButtonEnabled(id, enabled)
```

Enables or disables one of the customs buttons previously created by calling `form_displayCustomButtonBar`.

Parameters

id

The ID of the button to show or hide

enabled

The new enabled/disabled status of the button

Return value

None.

Example

form_displayCustomButtonBar

```
function form_displayCustomButtonBar(data)
```

Shows a customized button bar below the form.

Parameters

data

A JSON object describing the button structure to create.

Return value

None.

Example

form_setCustomButtonVisible

```
function form_setCustomButtonVisible(id, visible)
```

Shows or hides one of the customs buttons previously created by calling `form_displayCustomButtonBar`.

Parameters

id

The ID of the button to show or hide

visible

The new visible/hidden status of the button

Return value

None.

Example

form_setCustomButtonHidden

This function has been deprecated; it exists only for compatibility reasons and has no effect.

```
function form_setCustomButtonHidden(id, visible)
```

Hides one of the customs buttons previously created by calling `form_displayCustomButtonBar`.

Parameters

id

The ID of the button to show or hide

visible

The new visible/hidden status of the button

Return value

None.

Example

form_refreshDocumentList

This function has been deprecated; it exists only for compatibility reasons and has no effect. The user interface of Gargantua 8 no longer displays the document list at the same time as the task form.

```
function form_refreshDocumentList()
```

Refreshes the document list.

Parameters

None

Return value

None.

Example

form_displayDocumentListOnReturn

This function has been deprecated; it exists only for compatibility reasons and has no effect. The user interface of Gargantua 8 no longer displays the document list at the same time as the task form.

```
function form_displayDocumentListOnReturn()
```

Parameters

None

Return value

None.

Example

form_processAreaVerticalMaximize

This function has been deprecated; it exists only for compatibility reasons and has no effect. The user interface of Gargantua 8 no longer displays the document list at the same time as the task form.

```
function form_processAreaVerticalMaximize(bMaximize)
```

Expands/shrinks vertically the display area of the process form.

Parameters

bMaximize

Expand/Shrink the form area.

Return value

None.

Example

form_processAreaResizeToggle

This function has been deprecated; it exists only for compatibility reasons and has no effect. The user interface of Gargantua 8 no longer displays the document list at the same time as the task form.

```
function form_processAreaResizeToggle()
```

Extends/shrinks the display area of the form process.

Parameters

None.

Return value

None.

Example

form_processAreaResize

This function has been deprecated; it exists only for compatibility reasons and has no effect. The user interface of Gargantua 8 no longer displays the document list at the same time as the task form.

```
function form_processAreaResize(bResize)
```

Resizes the display area of the form process.

Parameters

bResize

Expand/Shrink the form area.

Return value

None.

Example

form_displayAlert

```
function form_displayAlert(data)
```

Displays an alert for the user to notice.

Parameters

data

A JSON object describing the alert to display.

Return value

The jQuery alert object.

Example

```
form_displayAlert({
    type : 'error',
     message : 'Error alert without a title ...'
});
```

```
form_displayAlert({
     type : 'info',
     title : 'Not neccesary',
     message : 'Info message here ...'
});
```

```
var $alert = form_displayAlert({
     type : 'success',
     keys : {
          message : 'common.edit.success'
     }
});

_.delay(function() {
     form_closeAlert($alert);
}, 500);
```

form_closeAlert

```
function form_closeAlert($alert)
```

Closes an alert previously displayed by a call to `form_displayAlert`.

Parameters

$alert

The jQuery object that represents the alert to close; created by a call to `form_displayAlert`.

Return value

None.

Example

```
var $alert = form_displayAlert({
    type : 'success',
    keys : {
          message : 'common.edit.success'
    }
});

_.delay(function() {
    form_closeAlert($alert);
}, 500);
```

form_displayDialog

```
function form_displayDialog(data)
```

Displays a dialog box for the user to enter data or perform actions.

Parameters

data

A JSON object describing the dialog to display.

Return value

The jQuery dialog object.

Example

```
form_displayDialog({
    type : 'error',
     message : 'Dialog error message here ...',
});
```

```
form_displayDialog({
    type : 'info',
     title : 'Dialog title', // defaults to Gargantua if not specified
     message : 'Dialog info message here ...'
});
```

```
// Button codes :
// var Button = {
//   ESCAPE : 0,
//   OK : 1,
//   CANCEL : 2,
//   YES : 3,
//   NO : 4,
//   CLOSE : 5
// }

// Button combination codes :
// var Buttons = {
//   OK : 1,
//   OK_CANCEL : 2,
//   YES_NO : 3,
//   YES_NO_CANCEL : 4
```

```
//   CANCEL : 5
//   CLOSE : 6
// }

form_displayDialog({
     type : 'question',
     title : 'Survey',
     message : 'What do you want to do ? ...',
     buttons : Buttons.YES_NO_CANCEL,
     callback : function(e, key, btn) {
          console.log('pressed : ' + btn);

          switch (btn)
          {
               case Button.YES:
                    // do something
                    break;
               case Button.NO:
                    // do something
                    break;
               default:
                    break;
          }

     }
});
```

```
var $dialog = form_displayDialog({
     type : 'success',
     keys : {
          message : 'common.edit.success'
     }
});


_.delay(function() {
     form_closeDialog($dialog);
}, 500);
```

```
form_displayDialog({
     type : 'warning',
     keys : {
          title : 'common.edit.title',
          message : 'common.edit.denied'
     }
});
```

form_displayCustomDialog

```
function form_displayCustomDialog(data)
```

Displays a custom dialog box for the user to enter data or take actions.

Parameters

data

A JSON object describing the dialog to display.

Return value

The jQuery dialog object.

Example

```
form_displayCustomDialog({
    title : 'Sélection d\'une fiche de traitement',
     body : function() {
          var htmlTemplate = '<table class="table table-hover">'
               + '<thead class="thead-primary font-weight-bold">'
               + '<tr><th class="w-5"> </th><th
class="w-10">Référence</th><th>Noms et descriptions des fiches</th></tr>'
               + '</thead>'
               + '<tbody>{{#each objects}}'
               + '<tr><td class="text-center"><input type="checkbox"
/></td><td>{{reference}}</td><td>'
               + '<div>{{name}}</div>'
               + '<div class="text-muted">{{description}}</div>'
               + '</td></tr>'
               + '{{/each}}</tbody>'
               + '</table>';

          var template = Handlebars.compile(htmlTemplate);
          var objects = [
               {
                    reference : 'TRT-0001',
                    name : 'Gestion des fiches du personnel',
                    description : 'Gestions des informations sur les fiches'
               },
               {
                    reference : 'TRT-0209',
                    name : 'Traitement pour les souris',
                    description : 'Description'
               },
               {
```

```
                    reference : 'TRT-0213',
                    name : 'Traitement des ânes',
                    description : 'A'
               }
          ];
          return template({
               objects : objects
          });
     },
     buttons : {
          validate : {
               label : 'Valider',
               className : 'btn-primary',
               callback : onValidate
          },
          close : {
               label : 'Fermer'
          }
     }
});
```

```
form_displayCustomDialog({
     id : 'attach-dlg',
     title : 'Ajouter une annexe',
     body : '<form role="form">'
          + '<div class="form-group">'
          + '<label class="form-control-label" for="attach-dlg-
file">Fichier</label>'
          + '<input class="form-control" id="attach-dlg-file" name="attach-dlg-
file" type="file" required="required" data-controltype="file">'
          + '</div>'
          + '<div class="form-group">'
          + '<label class="form-control-label" for="attach-dlg-
name">Libellé</label>'
          + '<input class="form-control" id="attach-dlg-name" name="attach-dlg-
name" type="text">'
          + '</div>'
          + '<div class="form-group form-check">'
          + '<input class="form-check-input" id="attach-dlg-printable"
name="attach-dlg-printable" type="checkbox">'
          + '<label class="form-check-label" for="attach-dlg-
printable">Imprimable</label>'
          + '</div>'
          + '</form>',

     buttons : {
          save : {
               label : 'Sauvegarder',
               className : 'btn-primary',
               // callback : onSave
          },
          cancel : {
               label : 'Annuler
          }
     },
```

```
     callback : function(e, key, type) {
          switch (key)
          {
               case 'save':
                    onSave.call(this);
                    break;
               default:
                    break;
          }
     },
     controls : true
});
```

```
form_displayCustomDialog({
     title : 'Sélection d\'une fiche de traitement',
     body : {
          template : '<table class="table table-hover">'
               + '<thead class="thead-primary font-weight-bold">'
               + '<tr><th class="w-5"> </th><th
class="w-10">Référence</th><th>Noms et descriptions des fiches</th></tr>'
               + '</thead>'
               + '<tbody>{{#each objects}}'
               + '<tr><td class="text-center"><input type="checkbox"
/></td><td>{{reference}}</td><td>'
               + '<div>{{name}}</div>'
               + '<div class="text-muted">{{description}}</div>'
               + '</td></tr>'
               + '{{/each}}</tbody>'
               + '</table>',
          source : function() {
               var objects = [ {
                         reference : 'TRT-0001',
                         name : 'Gestion des fiches du personnel',
                         description : 'Gestions des informations sur les fiches'
                    },  {
                         reference : 'TRT-0209',
                         name : 'Traitement pour les souris',
                         description : 'Description'
                    },  {
                         reference : 'TRT-0213',
                         name : 'Traitement des ânes',
                         description : 'A'
                    } ];

                    return {
                         objects : objects
                    };
               }
          },
          buttons : {
               validate : {
                    label : 'Valider',
                    className : 'btn-primary',
                    callback : onValidate
```

```
            },
            close : {
                label : 'Fermer'
            }
    }
});
```

```
form_displayCustomDialog({
      url : '/app/cmds/document/replace/dialog',
      callback : function(e, key, btn) {
            window.console.log(btn);
      }
});
```

```
// Button codes :
// var Button = {
//   ESCAPE : 0,
//   OK : 1,
//   CANCEL : 2,
//   YES : 3,
//   NO : 4,
//   CLOSE : 5
// }

// Button combination codes :
// var Buttons = {
//   OK : 1,
//   OK_CANCEL : 2,
//   YES_NO : 3,
//   YES_NO_CANCEL : 4
//   CANCEL : 5
//   CLOSE : 6
// }

form_displayCustomDialog({
      title : 'Custom dialog',
      body : function() {
            return '<p>Test body</p>';
      },
      // title : function() {
      //   return 'Custom dialog';
      //},

      // body : '<p>Test body</p>',
      buttons : Buttons.OK,
      callback : function(e, key, btn) {
            window.console.log(btn);
      }
});
```

form_closeDialog

```
function form_closeDialog($dialog)
```

Closes a dialog box previously displayed by a call to `form_displayDialog` or `form_displayCustomDialog`.

Parameters

$dialog

The jQuery object that represents the dialog to close; created by a call to `form_displayDialog` or `form_displayCustomDialog`.

Return value

None.

Example

```
var $dialog = form_displayDialog({
     type : 'success',
     keys : {
          message : 'common.edit.success'
     }
});

_.delay(function() {
     form_closeDialog($dialog);
}, 500);
```

form_getDialogControl

Parameters

...

...

Return value

...

Example

...

form_getDialogElement

Parameters

...

...

Return value

...

Example

...

Button

Buttons

form_loadScript

Parameters

...

...

Return value

...

Example

...

form_loadScripts

Parameters

...

...

Return value

...

Example

...

form_loadStylesheet

Parameters

...

...

Return value

...

Example

...

FORM_LOADSTYLESHEETS

PARAMETERS

...

...

RETURN VALUE

...

EXAMPLE

...

JavaScript functions for triggering actions

resetElement

```
function resetElement(id)
```

Resets the value of the given attribute input control.

Parameters

id

The ID of the attribute whose input control must be reset

Return value

None.

Example

runServerScript

```
function runServerScript(scriptName, parameters, callback, bAsync)
```

Starts the script execution at the server level. The script that is called must begin by the “execute” function:

```
function execute(params) {
    // processing
}
```

Paramètres et valeur retour :

scriptName

The name of the script to execute.

parameters

Parameters for the script. They must include the “databaseId” parameter, that indicates the database in which the script to run is located.

For instance: “databaseId=DATA0008000000000000000000000000000000”.

callback

Function that must be executed upon receiving the response from the server.

bAsync

A boolean that indicates if the request must be made in synchronous or asynchronous mode. The asynchronous mode is preferred, as it allows for a more responsive user interface.

Return value

Server response in the indicated format of the invoked function. In case of an async call the response will be received and processed in the callback function.

This API function has been replaced by the form_runScript function; please use the new function whenever possible.

runServerScript2

```
function runServerScript2(scriptName, functionName, parameters, callback, bAsync)
```

Starts the script execution at the server level.

Paramètres et valeur retour :

scriptName

The name of the script to execute.

functionName

The name of the function to execute.

parameters

Parameters for the script. They must include the “databaseId” parameter, that indicates the database in which the script to run is located.

For instance: “databaseId=DATA000800000000000000000000000000000000”.

callback

Function that must be executed upon receiving the response from the server.

bAsync

A boolean that indicates if the request must be made in synchronous or asynchronous mode. The asynchronous mode is preferred, as it allows for a more responsive user interface.

Return value

Server response in the indicated format of the invoked function. In case of an async call the response will be received and processed in the callback function.

This API function has been replaced by the form_runScript function; please use the new function whenever possible.

FORM_RUNSCRIPT

```
function form_runScript(options)
```

Starts the script execution at the server level.

Parameters and return value:

options

Key-value object that specifies the following options:

- `script` ou `scriptName` - the name of the script to be executed
- `function` ou `functionName` - the name of the function to be executed
- `acync` - boolean indicating if the request should be synchronous (false) or asynchronous (true)
- `data` - object with aditional data to be sent
- `callback` - function to be called when the result is available.

Return value

None.

Example :

```
form_runScript({
    script: 'server.queries',
    'function': 'getValues',
    data: {
        docId: ids[0]
    },
    callback: function (result) {
        try {
           var json = JSON.parse(result);
           form_info(json);
           ....
        }
        catch (e) {
            form_error(e);
        }
    }
});
```

form_apiRequest

```
function form_apiRequest(options)
```

Used to make calls to outside sources (and avoid cross-origin request problems). The call will go through the Gargantua server, who will actually make the request on your behalf.

Parameters and return value:

options

Key-value object that contains the following options:

- `url` - the URL to call
- `method` - HTTP method (defaults to GET)
- `headers` - HTTP headers (as a key-value object)
- `params` - HTTP parameters (as a key-value object)
- `body` - HTTP body (as string or object)
- `contentType` - the content type
- `async` - boolean indicating if the request should be synchronous (false) or asynchronous (true)
- `callback` - function to be called when the result is available

Return value

None.

Example :

```
form_apiRequest({
    url :  'https://thequoteshub.com/api/random-quote',
    method : 'GET',
    callback : function(result) {
        form_info(JSON.parse(result));
    }
});
```

form_scanDocuments

This function has been deprecated; it exists only for compatibility reasons. Please replace it with "form_addDocuments". Currently, this function call defaults to "form_addDocuments"

```
function form_scanDocuments(direct)
```

Launches the scanning operation if the execution context is properly initialized.

Parameters

direct

Indicator for direct scanning. See `form_addDocuments` for more information.

Return value

None.

Example

form_scanDocumentsExt

This function has been deprecated; it exists only for compatibility reasons. Please replace it with "form_addDocuments". Currently, this function call defaults to "form_addDocuments"

```
function form_scanDocumentsExt(direct, label)
```

Launches the scanning operation if the execution context is properly initialized.

Parameters

direct

Indicator for direct scanning. See `form_addDocuments` for more information.

label

The label to use for the newly scanned documents.

Return value

None.

Example

form_scanDocumentsExt2

This function has been deprecated; it exists only for compatibility reasons. Please replace it with "form_addDocuments". Currently, this function call defaults to "form_addDocuments"

```
function form_scanDocumentsExt2(direct, form)
```

Launches the scanning operation if the execution context is properly initialized.

Parameters

direct

Indicator for direct scanning. See `form_addDocuments` for more information.

form

The Form to use for the newly scanned documents.

Return value

None.

Example

form_scanDocumentsExt3

This function has been deprecated; it exists only for compatibility reasons. Please replace it with "form_addDocuments". Currently, this function call defaults to "form_addDocuments"

```
function form_scanDocumentsExt3(direct, form, label)
```

Launches the scanning operation if the execution context is properly initialized.

Parameters

direct

Indicator for direct scanning. See `form_addDocuments` for more information.

form

The Form to use for the newly scanned documents.

label

The label to use for the newly scanned documents.

Return value

None.

Example

form_createDocument

```
function form_createDocument(template, label, form, fulltext, markers)
```

Creates a new document based on a template and text replacements,

Parameters

template

The template ID

label

The label to use for the new document.

form

The form to use for the new document.

fulltext

Boolean to indicate whether the document has to be indexed for fulltext search

markers

A map of (<placeholder>, <value>) elements to replace in the new document

Return value

The new document ID.

Example

form_addDocuments

```
function form_addDocuments(options)
```

Launches the document import operation if the execution context is properly initialized.

Parameters

options

An object describing the addDocument operation options.

direct

Indicator for direct scanning; otherwise a form selector is shown first. Supported values are: **DIRECT_NONE** (0), **DIRECT_SCAN** (1), **DIRECT_SCANWITHPROFILE** (2), **DIRECT_IMPORT** (3), **DIRECT_EMAIL** (4) or **DIRECT_IMPORT_ANY** (5). You can use either the constant or its value.

form

The name of the form to use for the new documents; takes precedence over the formId, if both are specified

formId

The id of the form to use for the new documents.

label

The label for the new documents; takes precedence over the name, if both are specified

name

The name for the new documents; used as a fallback if label is not specified

Return value

None.

Example

form_addDocumentsDirect

This function has been deprecated; it exists only for compatibility reasons. Please replace it with "form_addDocuments". Currently, this function call defaults to "form_addDocuments"

```
function form_addDocumentsDirect(direct)
```

Launches the add document operation if the execution context is properly initialized.

Parameters

direct

Indicator for direct add (no form selector). See `form_addDocuments` for more information.

Return value

None.

Example

form_certifyDocument

```
function form_certifyDocument(params)
```

Shows the document certification user interface elements for the given document.

Parameters

params

An object describing the operation options.

id

The document id to display the certification UI for.

Return value

None.

Example

form_editDocument

```
function form_editDocument(id, create, callback)
```

Triggers the 'edit native' operation for a given document

Parameters

id

The document ID to edit

create

Indicates whether this is a new document or an existing document; for new documents, when the editing operation ends, there will be no dialog box asking for versioning information

callback

The callback used when the operation opens the native application.

Return value

None.

Example

form_viewDocument

```
function form_viewDocument(id, options)
```

Displays the view document user interface elements according to the global layout and settings; has the same effect as the user clicking on the document label or double-clicking on the document line or thumbnail.

Parameters

id

The document ID to display

options

A JSON object specifying the operation options.

The options parameter is a JSON object that needs to follow the following structure; one or more members may be present.

```
{
    viewer : {
        toolbar : {
            close : false,
            zapper : false,
            pinZapper : false,
            editNative : false,
            closeOnEditNative : false,
            switchView : false
        },
        label : true,
        panel : {
            display : 'none',
            fragment : true,
            notes : true,
            ocr : true,
            links : false,
            actions : false,
            sysinfo : true
        }
```

```
    },
    getDisplayData : $.noop,
    onZap : $.noop,
    onClose : $.noop,
    onActionFinished : $.noop
}
```

viewer/toolbar

individual commands for the viewer panel : close button, zapper, zapper for pins, edit native, close viewer on editnative and switch view.

label

boolean to indicate if label should be displayed or not

panel

what the footer panel shoulr display : default display, fragment, notes, ocr, links, actions, system information

getDisplayData, onZap, onClose, onActionFinished

various callbacks for the viewer panel

Return value

None.

Example

form_signDocument

```
function form_signDocument(params)
```

Shows the document signing user interface elements for the given document.

Parameters

params

An object describing the operation options.

id

The document id to display the signing UI for.

Return value

None.

Example

form_viewDocuments

```
function form_viewDocuments()
```

Displays the view documents user interface elements according to the global layout and settings.

This API function call is used exclusively in inbox context.

Parameters

None.

Return value

None.

Example

...

form_downloadDocuments

```
function form_downloadDocuments(ids)
```

Function that downloads the documents identified by the ids array.

Parameters

ids

Array of document ids.

Return value

None

form_editObject

```
function form_editObject(id, options)
```

Function that opens the view/edit attributes dialog for the specified object.

Parameters

id

The ID of the object to edit.

options

Edit options.

Return value

None.

Example

...

form_save

```
function form_save()
```

Saves the form content of the displayed process or task.

Parameters and return value:

No parameter

Return value

None.

FORM_SAVEIMMEDIATE

```
function form_saveImmediate()
```

Saves the form content of the displayed process or task immediately and silently (no success alert will be shown).

Parameters and return value:

No parameter

Return value

None.

FORM_SAVEDRAFT

```
function form_saveDraft()
```

Saves the form of the displayed draft process.

Parameters and return value:

No parameter

Return value

None.

This API function has been replaced by the form_save function; please use the new function whenever possible.

form_saveDraftImmediate

```
function form_saveDraftImmediate()
```

Saves all the form content (all attribute values) immediately and silently.

Parameters and return value:

No parameter

Return value

None.

This API function has been replaced by the form_saveImmediate function; please use the new function whenever possible.

FORM_SAVETASKIMMEDIATE

```
function form_saveTaskImmediate()
```

Saves all the form content (all attribute values) immediately and silently.

Parameters and return value:

No parameter

Return value

None.

This API function has been replaced by the form_saveImmediate function; please use the new function whenever possible.

FORM_SAVETASK

```
function form_saveTask()
```

Saves the form of the displayed task.

Parameters and return value:

No parameter

Return value

None.

This API function has been replaced by the form_save function; please use the new function whenever possible.

FORM_CANCELDRAFT

```
function form_cancelDraft()
```

Cancels the draft and thus the process start operation.

Parameters and return value:

No parameter

Return value

None.

FORM_CANCEL

```
function form_cancel()
```

Cancels the process draft (and by consequence the process start) or the current task if the execution context is properly initialized, depending on the case.

Parameters and return value:

No parameter

Return value

None.

FORM_FINISH

```
function form_finish()
```

Starts a new process if the current process is a draft or sends the current task (validate task - finish task) if the execution context is properly initialized, depending on the case.

Parameters and return value:

No parameter

Return value

None.

form_startProcess

```
function form_startProcess()
```

Starts a new process if the current process is a draft and the execution context is properly initialized.

Parameters and return value:

No parameter

Return value

None.

FORM_CANCELTASK

```
function form_cancelTask()
```

Sends the task (canceled task) if the execution context is properly initialized.

Parameters and return value:

No parameter

Return value

None.

FORM_FINISHTASK

```
function form_finishTask()
```

Sends the task (validate task - finish task) if the execution context is properly initialized.

Parameters and return value:

No parameter

Return value

None.

form_executeSearch

```
function form_executeSearch()
```

Starts the execution of a search if the execution context is properly initialized.

This API function call is used exclusively for forms available for the search context.

Parameters

None.

Return value

None.

Example

FORM_ESCALATETASK

```
function form_escalateTask()
```

Escalates the task according to the rules set by the process definition.

doValidateForm

```
function doValidateForm()
```

Performs the form fields validation test. Disables the send button and highlights in red the background of all the fields that contain errors.

Parameters

None.

Return value

None.

Example

HANDLER_ONSWITCHPAGE

```
function form_switchPage(page)
```

Switch to the given form page.

PARAMETERS

page

the name of the page to display.

RETURN VALUE

None.

EXAMPLE

form_switchForm

```
function form_switchForm(form, page)
```

Switch to the given form and form page.

Parameters

form

the name of the form that contains the page to display.

page

the name of the page to display.

Return value

None.

Example

FORM_GOTOTASKS

```
function form_gotoTasks()
```

Returns to the task list from the form/document display.

Parameters

None

Return value

None

Exceptions

n/a

Example

```
form_gotoTasks();
```

form_invokePlugin

```
function form_invokePlugin(pluginId, command, params, format)
```

Executes the given command from a server plugin, using the provided parameters and output format.

Parameters

pluginId

the ID of the plugin to invoke.

command

the name of the plugin command to execute.

params

A JSON objects containing command-specific parameters.

format

A string specifying the expected return format; can be either xml or text.

Return value

The server response in the requested format, if the plugin can handle it; otherwise, it is a text response.

Example

Advanced JavaScript functions

form_newControl

```
function form_newControl(id, type, options)
```

Creates a new input control based on the given parameters.

Parameters

Return value

Example

form_destroyControl

```
function form_destroyControl(id)
```

Destroys a previously manually created control

Parameters

id

The form element control ID

Return value

None

Example

form_getControl

```
function form_getControl(id)
```

Retrieves the control for the given input ID

Parameters

id

The form element control ID

Return value

The new control object.

Example

form_getElement

```
function form_getElement()
```

Retrieves the jQuery object associated with the given element ID

Parameters

id

The page element ID

Return value

The jQuery object associated with the given element ID

Example

getElementKeyAndValue

```
function getElementKeyAndValue()
```

Retrieves the selected (key, value) pair from the autocomplete control as a JSON object.

Parameters

id

The form element control ID

Return value

A JSON object that represents the (key, value) pair for the given selector.

Example

setElementKeyAndValue

```
function setElementKeyAndValue(id, key, value)
```

Sets the key and value for the given autocomplete control (a list)

Parameters

id

The form element control ID

key

The key to be set

value

The value to be set

Return value

None

Example

form_saveGlobalGrids

```
function form_saveGlobalGrids()
```

Saves the values of all grids in the form.

> ***This API function has been deprecated in Gargantua 8. You may safely remove it when porting existing forms from older Gargantua versions.***

Parameters

None

Return value

None

Example

form_validateLatency

```
function form_validateLatency(timeout)
```

Modifies the form validation latency; this affects how the validation is performed on the form values.

Parameters

timeout

The validation timeout to be used; effective from the function call

Return value

None

Example

form_getValidateLatency

```
function form_getValidateLatency()
```

Retrieves the current form validation latency value.

Parameters

Return value

newMode

Example

form_validateLatencyOffset

```
function form_validateLatencyOffset(offset)
```

Modifies the form validation latency offset; this affects how the validation is performed on the form values.

Parameters

offset

The validation timeout offset to be used; effective from the function call

Return value

None

Example

form_validateMode

```
function form_validateMode(newMode)
```

Changes the validation mode for the form.

Parameters

newMode

The new validation mode for the form; it can be either `VALIDATE_ONSUBMIT` or `VALIDATE_CONTINUOUS`

Return value

None

Example

form_getValidateMode

```
function form_getValidateMode()
```

Retrieves the current validation mode for the form.

Parameters

None

Return value

The current validation mode. It can be either `VALIDATE_ONSUBMIT` or `VALIDATE_CONTINUOUS`

Example

form_getValidationMessage

```
function form_getValidationMessage(failedIds, failedReasons, failedCodes [, options])
```

Returns a standard validation message with the form errors.

Parameters

failedIds

An array with the failed attribute IDs.

failedReasons

An array with the failure reasons.

failedCodes

An array with the failure codes.

options

An object containing the following possible options:

- `multiline` parameter indicating if the different errors should be rendered on separate lines or not. By default this is `false`.

Return value

None.

Example

```
function fun_form_post_validatereponse(failedIds, failedReasons, failedCodes)
{
    var message = form_getValidationMessage(failedIds, failedReasons, failedCodes, {
multiline: true });
    if (message)
        form_displayAlert({ type : 'error', message: message });
}
```

FORM_SETSUBMITONENTER

```
function form_setSubmitOnEnter(bEnable)
```

Enables or disables the use of the **Enter** key for submitting the form.

By default, the "Enter" key is enabled for submitting the form.

Parameters

bEnable

Boolean to enable or disable the use of the **Enter** key for submitting the form..

Return value

None.

Example

```
form_setSubmitOnEnter(false);
```

Controls

General behavior

getValue

```
getValue()
```

Gets the value of the control as a string.

Parameters

None.

Return value

The current value of the control.

Example

```
var control = form_getControl(attrId);
var value = control.getValue();
```

Equivalent to:

```
var value = getElementValue(attrId);
```

setValue

```
setValue(value)
```

Sets the value of the control as a string.

Parameters

value

The value to set.

Return value

None

Example

```
var value = "...";
var control = form_getControl(attrId);
control.setValue(value);
```

Equivalent to:

```
setElementValue(attrId, value);
```

ENABLE

```
enable()
```

Enables the control.

Parameters

None

Return value

None

Example

```
var control = form_getControl(attrId);
control.enable();
```

Equivalent to:

```
setElementEnabled(attrId, true);
```

DISABLE

```
disable()
```

Disables the control.

Parameters

None

Return value

None

Example

```
var control = form_getControl(attrId);
control.disable();
```

Equivalent to:

```
setElementEnabled(attrId, false);
```

isEmpty

```
isEmpty()
```

Checks if the control value is empty.

Parameters

None

Return value

`true` if the value is empty, `false` otherwise

Example

```
var control = form_getControl(attrId);
if (control.isEmpty())
{ ... }
```

RESET

```
reset()
```

Resets the value of the control to the initial value.

Parameters

None

Return value

None

Example

```
var control = form_getControl();
control.reset();
```

isEnabled

```
isEnabled()
```

Checks if the control is enabled.

Parameters

None

Return value

`true` if the control is enabled, `false` otherwise

Example

```
var control = form_getControl(attrId);
if (control.isEnabled())
{ ... }
```

Equivalent to:

```
isElementEnabled(attrId)
```

hasError

```
hasError()
```

Checks if the control value is correct or not (if it respects the control's pattern).

Parameters

None

Return value

`true` if the value is correct, `false` otherwise

Example

```
var control = form_getControl(attrId);
if (control.hasError())
{ ... }
```

Simple input fields

Simple string input

To create a string control:

```
var control = form_newControl(attrId, 'string');
```

Numeric input

To create a numeric control:

```
var control = form_newControl(attrId, 'number', options);
```

The options include:

type

The type of number `[integer|float]`

negative

Boolean indicating if the number can be negative or not

min

The minimum value (any value below it will be considered invalid)

max

The maximum value (any value above it will be considered invalid)

maxUnits

The maximum number of units

maxSubunits

The maximum number of subunits

getNumber

```
getNumber()
```

Gets the value of the control as a number.

Parameters

None

Return value

The value as a number.

Example

```
var control = form_getControl(attrId);
var number = control.getNumber();
```

setNumber

```
setNumber(number)
```

Sets the value of the control as a number.

Parameters

number

The value as a number.

Return value

None

Example

```
var control = form_getControl(attrId);
control.setNumber(250);
```

DATE INPUT

To create a date control:

```
var control = form_newControl(attrId, 'date', options);
```

The options include:

mouseWheel

Boolean indicating if the mouse wheel should or not be used to alter the input's value

getDate

```
getDate()
```

Gets the value of the control as a date (without the hour, minutes, seconds and milliseconds).

Parameters

None

Return value

The value as a date.

Example

```
var control = form_getControl(attrId);
var date = control.getDate();
```

setDate

```
setDate(date)
```

Sets the value of the control as a date (without the hour, minutes, seconds and milliseconds).

Parameters

date

The value as a date.

Return value

None

Example

```
var control = form_getControl(attrId);
control.setDate(new Date());
```

TIME INPUT

To create a time control:

```
var control = form_newControl(attrId, 'time', options);
```

The options include:

mouseWheel

Boolean indicating if the mouse wheel should or not be used to alter the input's value

getTime

```
getTime()
```

Gets the value of the control as an array with 2 elements (hours and minutes).

Parameters

None

Return value

The value as an array with 2 elements (hours and minutes).

Example

```
var control = form_getControl(attrId);
var timeArray = control.getTime();
```

setTime

```
setTime(time)
```

Sets the value of the control as an array with 2 elements (hours and minutes).

Parameters

time

The value as an array with 2 elements (hours and minutes).

Return value

None

Example

```
var control = form_getControl(attrId);
control.setTime([10, 25]);
```

Date & time input

To create a date-time control:

```
var control = form_newControl(attrId, 'datetime', options);
```

The options include:

mouseWheel

Boolean indicating if the mouse wheel should or not be used to alter the input's value

getTimestamp

```
getTimestamp()
```

Gets the value of the control as a date.

Parameters

None

Return value

The value as a date.

Example

```
var control = form_getControl(attrId);
var date = control.getTimestamp();
```

setTimestamp

```
setTimestamp(date)
```

Sets the value of the control as a date.

Parameters

date

The value as a date.

Return value

None

Example

```
var control = form_getControl(attrId);
control.setTimestamp(new Date());
```

Time delay input

To create a time delay control:

```
var control = form_newControl(attrId, 'timedelay', options);
```

The options include:

mouseWheel

Boolean indicating if the mouse wheel should or not be used to alter the input's value

getDelay

```
getDelay()
```

Gets the value of the control as a number (in seconds).

Parameters

None

Return value

The value as a number.

Example

```
var control = form_getControl(attrId);
var delayInSeconds = control.getDelay();
```

setDelay

```
setDelay(delay)
```

Sets the value of the control as a number (in seconds).

Parameters

delay

The value as a date.

Return value

None

Example

```
var control = form_getControl(attrId);
control.setDelay(3600); // 60 minutes
```

EMAIL INPUT

To create an email control:

```
var control = form_newControl(attrId, 'email', options);
```

The options include:

multiple

Boolean indicating if the value can contain multiple emails

source

The source URL

LISTS

AUTOCOMPLETE

To create an autocomplete control:

```
var control = form_newControl(attrId, 'autocomplete', options);
```

The source of the autocomplete is an array of objects. Each object should have at least two properties: a unique value that identifies the object and a display value (for example an id and a name).

The options include:

multiple

Boolean indicating if the control allows multiple values

displayKey

The key identifying the value to display in the control element (for instance the name in the example above)

valueKey

The key that uniquely identifies the selected value (for instance the id in the example above)

freeInput

Boolean indicating if the autocomplete should allow the user to write values that do not exist in the source

source

The source of the autocomplete (can be a URL, an object or a function)

minLength

The minimum number of characters the user has to type before the autocomplete is triggered

prepareSuggestion

A callback function allowing you to customize the display of the autocomplete suggestions

Example

```
form_newControl(attrId, 'autocomplete', {
    displayKey: 'value',
    valueKey: 'key',
    source: {
        values : [
            { key : 'k1', value : 'New York' },
            { key : 'k2', value : 'Chicago' },
            { key : 'k3', value : 'Geneva' },
            { key : 'k4', value : 'London' },
            { key : 'k5', value : 'Helsinki' },
            { key : 'k6', value : 'Boston' },
            { key : 'k7', value : 'Paris (France)' },
            { key : 'k8', value : 'Paris (Texas)' },
            { key : 'k9', value : 'Memphis (Egypt)' },
            { key : 'k10', value : 'Memphis (Tennessee)' },
            { key : 'k11', value : 'Madrid' },
             { key : 'k12', value : 'Vienna' },
             { key : 'k13', value : 'Sydney (Australia)' },
             { key : 'k14', value : 'Sydney (Canada)' }
        ]
    }
});
```

The source should always look like the example above even if it’s a function or an URL, the returned value must follow the same pattern.

getKey

```
getKey()
```

Gets the key of the control.

Parameters

None

Return value

The key of the control.

Example

```
var control = form_getControl(attrId);
var key = control.getKey();
var value = control.getValue();
```

For the example of cities above, if the user selected “Paris (France)”, the key will be `k7` and the value will be `Paris (France)`.

setKey

```
setKey(key)
```

Sets the key of the control.

Parameters

key

The key to set.

Return value

None

Example

```
var control = form_getControl(attrId);
control.setKey('k7');
```

getKeyAndValue

```
getKeyAndValue()
```

Gets an object containing the key and value of the control (the property names will be those configured as displayKey and valueKey).

Parameters

None

Return value

An object containing the key and value of the control.

Example

```
var control = form_getControl(attrId);
var obj = control.getKeyAndValue();
```

For the example of cities above, if the user selected "Paris (France)", the object will be `{ 'key' : 'k7', 'value' : 'Paris (France)' }`.

Equivalent to:

```
getElementKeyAndValue(attrId);
```

setKeyAndValue

```
setKeyAndValue(object)
```

Sets the key and value of the control.

Parameters

object

The key and value to set.

Return value

None

Example

```
var control = form_getControl(attrId);
control.setKeyAndValue({ 'key' : 'k7', 'value' : 'Paris (France)' });
```

Equivalent to:

```
setElementKeyAndValue(attrId, { 'key' : 'k7', 'value' : 'Paris (France)' });
```

SIMPLE LIST

To create a simple list control:

```
var control = form_newControl(attrId, 'list', options);
```

The options include:

flags

`[m|s|p|c]` The flags indicate the type of the list. `m` stands for multiple selection list, `s` stands for simple list, `p` stands for full path list, and `c` stands for category list

source

The source of the list (can be a URL, an object or a function)

closed

Boolean indicating if the list should not allow the user to write values that do not exist in the source

addEmptyOption

Boolean indicating if an empty option should be added to the list

prepareOption

A callback function allowing you to customize the display of the list options

linkId

The id of the master list (if you need to create linked lists; for multi level lists - each list will display it's own level)

Example

```
form_newControl(attrId, 'list', {
    source: {
        values : [
            { key : 'c1', value : 'America', values : [
                { key : 'k1', value : 'New York' },
                { key : 'k2', value : 'Chicago' },
                { key : 'k3', value : 'Boston' }
             ]},
             { key : 'c2', value : 'Europe', values : [
                { key : 'k4', value : 'London' },
                { key : 'k5', value : 'Paris' },
                { key : 'k6', value : 'Vienna' }
             ]}
        ]
    }
});
```

The example above will build a list with 2 levels.

If you add the `flags: 'c'` option the list will transform into a category list in which only the leaf values will be selectable.

By default the flags option is considered `'s'`, in which case all values are selectable. In the case of `'m'` flag you can select multiple values, and in the case of `'p'` flag the selected value will contain the full path (for instance 'Europe/Paris').

getKey

```
getKey()
```

Gets the key of the control.

Parameters

None

Return value

The key of the control.

Example

```
var control = form_getControl(attrId);
var key = control.getKey();
var value = control.getValue();
```

For the example of cities above, if the user selected “Paris (France)”, the key will be `k7` and the value will be `Paris (France)`.

setKey

```
setKey(key)
```

Sets the key of the control.

Parameters

key

The key to set.

Return value

None

Example

```
var control = form_getControl(attrId);
control.setKey('k7');
```

getKeyAndValue

```
getKeyAndValue()
```

Gets an object containing the key and value of the control (the property names will be those configured as displayKey and valueKey).

Parameters

None

Return value

An object containing the key and value of the control.

Example

```
var control = form_getControl(attrId);
var obj = control.getKeyAndValue();
```

For the example of cities above, if the user selected "Paris (France)", the object will be `{ 'key' : 'k7', 'value' : 'Paris (France)' }`.

Equivalent to:

```
getElementKeyAndValue(attrId);
```

setKeyAndValue

```
setKeyAndValue(object)
```

Sets the key and value of the control.

Parameters

object

The key and value to set.

Return value

None

Example

```
var control = form_getControl(attrId);
control.setKeyAndValue({ 'key' : 'k7', 'value' : 'Paris (France)' });
```

Equivalent to:

```
setElementKeyAndValue(attrId, { 'key' : 'k7', 'value' : 'Paris (France)' });
```

MULTI-LINE LIST

To create a multiline list control:

```
var control = form_newControl(attrId, 'multilinelist', options);
```

This will create a multiline select element.

The options include:

source

The source of the list (an object; see simple list example)

addEmptyOption

Boolean indicating if an empty option should be added to the list

Radio button list

To create a radio button list control:

```
var control = form_newControl(attrId, 'radiolist', options);
```

This will create a list of radio elements.

The options include:

source

The source of the list (an object; see simple list example)

Checkbox list

To create a checkbox list control:

```
var control = form_newControl(attrId, 'checklist', options);
```

This will create a list of checkbox elements.

The options include:

source

The source of the list (an object; see simple list example)

Grid

Definition and properties

A grid is a complex type of attribute. In order to be able to use grids in a Gargantua form you must first define a type of grid , then you have to create an attribute of this type which can be used in the form.

A grid presents the data (content) in a table format and it allows the data to be added, viewed, modified in each editable cell. More, the grid can be customized to contain a set of internal features (defined by embedded functions and events handlers) but also to interact externally with other attributes of the form through form scripting. An attribute of grid type can be built assisted by the Grid Designer tool or it is fully scriptable and it can be build in pure Javascript language in a form script.

The grid properties:

Columns

defines the grid's structure

Header

defines the grid heading properties and style

`Toolbar`

provides built-in and user-defined features (actions)

`Functions`

collection of user-defined functions (Javascript source code)

`Events`

your own event handlers for the events exposed by the grid (Javascript source code)

Columns

The column properties:

`id`

the column identification string, it is unique for all columns and must respect the W3C recommandation for HTML ID attribute definition

`width`

numeric, represent the column width in pixels

`type`

the column's type `[string|icon|button|link|checkbox|list|date|time|timestamp|timespan|integer|float|currency|email|`

`label`

the column's name

`tooltip`

the column's tooltip

`defaultValue`

a value used for a corresponding cell when a new grid row is created; for now only columns having type `icon` (use any valid URL for an image or local images such as "${baseURL}/images/controls/grid/icons/grid-xxx.gif" where xxx is any of the values specified for toolbar icon) or `checkbox` (use value '1' or 'true' to set a checkbox as checked) are supported;

`align`

the cell text horizontal alignment `[left|center|right]`

`readonly`

the cell is not editable, but it can be activated and clickable

`editable`
: the cell is not editable, it can not be activated nor clickable

`onload`
: for list types only - callback function to provide the options values

`resetonload`
: for list types only - specify if `onload` shall be always invoked;

Header

The grid heading has the following properties:

`visible`
: toggles the visibility of the header.

`compact`
: specifies if the header is displayed on a single or multiple lines.

Toolbar

The grid toolbar has the following properties:

`visible`
: toggles the visibility of the toolbar.

`align`
: specifies the alignment of the toolbar. `[top|right|bottom|left]`

The toolbar contains three types of buttons: `[standard|custom|separator]`.

Standard buttons:

`gridButtonAddRow`
: add a new row

`gridButtonRemoveRow`
: remove the selected row

`gridButtonMoveRowUp`
: move the selected row one position up

`gridButtonMoveRowDown`
: move the selected row one position down

Button properties:

`type`

`[standard|custom|separator]`

`id`

the button identification string, it is unique for all buttons and must respect the W3C recommandation for HTML ID attribute definition

`visible`

toggle the button visible status (default yes)

`icon`

selected from a limited set of icons (”database”, “delete”, “disk”, “down”, “exclamation”, “eye”, “globe”, “minus”, “pencil”, “plus”, “question”, “star”, “tick-red”, “tick”, “up”)

`onclick`

associate an existing function as click event handler (triggered when the button is clicked)

`title`

tooltip text

`text`

label text

There is a button specific `onclick` event handler and a global `toolbarbuttonclicked` event handler triggered for all toolbar buttons.

The buttons order inside toolbar can be changed with drag and drop in Grid Designer UI.

Functions

The functions can be used to implement an internal feature or to be used as an event handler (ex: `onclick` handler).

A function is callable from another function like this:

```
this.callFunction(functionName, paramsObject);
```

Returned value is optional and shall be specified like this:

```
params.value = '...';
```

Events

Each event handler has two variables already defined: `this` (the grid object) and `params` object (this object is used to provide important parameters depending on event such as

`params.row, params.col, params.from, params.to, params.id`).

`loaded`

triggerred when `load` method (grid definition and data) has been finished;

no params defined

`dataloaded`

triggerred when `loadData` method has been finished;

no params defined

`changed`

triggerred when a cell has been modified;

params defined: `params.row` (number - cell's row), `params.col` (number - cell's column)

`focused`

triggerred when a cell has been focused;

params defined: `params.row` (number - cell's row), `params.col` (number - cell's column)

`clicked`

triggerred when a cell has been clicked;

params defined: `params.row` (number - cell's row), `params.col` (number - cell's column)

`rowmoved`

triggerred when a grid row has been moved up or down;

params defined: `params.from` (number - row initial index), `params.to` (number - row final index)

`toolbarbuttonclicked`

triggerred when a toolbar button has been clicked;

params defined: `params.id` (string - button's id)

Example of event handler for `toolbarbuttonclicked`:

```
var buttonId = params.id;

if (buttonId == "cmdShowValue")
{
        alert( this.getValue());
}
else if (buttonId == "gridButtonAddRow")
{
        var row = this.getRowsCount() - 1;
        var iLink = this.getColumnIndexById("cLink");
        var iIcon = this.getColumnIndexById("cIcon");
```

```
        var iButton = this.getColumnIndexById("cButton");

        this.setCellValue(row, iLink, "This is link-"+row);
        this.setCellValue(row, iButton, "button-"+row);
        var base = "${baseURL}/images/controls/grid/icons/grid-";
        var arrIcons = ["question.gif","exclamation.gif"];
        this.setCellValue(row, iIcon, base + arrIcons[row % 2]);
}
else if (buttonId == "cmdAutoSize")
{
        this.autoSize();
}
```

Scripting the grid

Creating a grid object:

```
var grid = form_newControl(attrId, 'grid', options);
```

The options include:

url

the URL from which to load the grid definition (XML)

definition

the grid definition (XML)

loaded

callback function invoked after the grid is initialized

The url and definition options are exclusive.

Load grid data:

```
grid.load(dataURL, true);
```

Event binding:

```
// bind 2 anonymous functions to the same event

grid. bindEvent('loaded', function() { log('loaded listener-1'); });

grid. bindEvent('loaded', function() { log('loaded listener-2'); });

// bind changed and focused events

grid. bindEvent('changed', function(param) { log('item ('+ param.row + ',' + param.col
+') changed' ); });

grid. bindEvent('focused', function(param) { log('item ('+ param.row + ',' + param.col
+') focused' ); });
```

Functions

addColumn

```
addColumn(props)
```

Adds a column specified by provided props to the grid.

Parameters

props

An object with the column properties

Return value

None

Example

```
var props = {
     'id': 'productName',
     'type': 'string',
     'label': 'Product',
     'tooltip': 'Product name',
     'align': 'left',
     'width': '250px'
};
grid.addColumn(props);
```

addToolbarButton

```
addToolbarButton(props)
```

Adds a button specified by provided props to the grid toolbar.

Parameters

props

An object with the button properties

Return value

None

Example

```
var props = {
    'id': 'open',
    'type': 'custom',
    'title': 'Open document',
    'text': 'Open'
};
grid.addToolbarButton(props);
```

append

```
append(noEdit)
```

Append a row at the end of the grid. It uses defaultValues to fill up the cells.

Parameters

`noEdit`

boolean that specifies if the append operation should not trigger the edit of the first visible cell in the newly added row

Return value

The newly added `tr` DOM element.

Example

```
grid.append();
```

or

```
var tr = grid.append();
```

appendBefore

```
appendBefore(tr, noEdit)
```

Append a row before the given table row. It uses defaultValues to fill up the cells.

Parameters

`tr`

the `tr` DOM element before which to append the new row

`noEdit`

boolean that specifies if the append operation should not trigger the edit of the first visible cell in the newly added row

Return value

The newly added `tr` DOM element.

Example

```
var tr = grid.getRow(2);
var newTr = grid.appendBefore(tr);
```

bindEvent

```
bindEvent(eventname, callback)
```

Bind a callback function to an event.

Parameters

eventName

`loaded | dataloaded | changed | focused | clicked | rowmoved | toolbarbuttonclicked`

The event to bind

callback

callback function assigned to event

Return value

Returns true for success, false if fails.

Example

```
// bind focus event
grid.bindEvent('focused', function(param) { log('item ('+ param.row + ',' + param.col
+') focused' ); });
```

appendAfter

```
appendAfter(tr, noEdit)
```

Append a row after the given table row. It uses defaultValues to fill up the cells.

Parameters

`tr`

the `tr` DOM element after which to append the new row

`noEdit`

boolean that specifies if the append operation should not trigger the edit of the first visible cell in the newly added row

Return value

The newly added `tr` DOM element.

Example

```
var tr = grid.getRow(2);
var newTr = grid.appendAfter(tr);
```

callFunction

```
callFunction(name, params)
```

Call an internal function (created with the Grid Designer UI) specified by `name` and `param` parameters object.

Parameters

`name`

The name of the function to call

`params`

The parameters for the function to call

Return value

None.

Example

editCell

```
editCell(td, options)
```

Activates a cell editor for the specified `td` DOM element.

Parameters

td

The cell to edit

options

An object with the parameters; for now the only available option is `triggerClick` (boolean - if `true`, force an additional click event)

Return value

None

Example

```
grid.editCell(td, {
    'trigger_click': true
});
```

editCheckbox

```
editCheckbox(td)
```

Activates a checkbox type cell for the specified `td` DOM element.

Parameters

td

The cell to edit

Return value

None

Example

```
grid.editCheckbox(td);
```

forceEditCell

```
forceEdit(td)
```

More than editCell, if the grid is read only, it will force the display of the cell control (in read only mode).

Parameters

td

The cell to activate edit for.

Return value

None

Example

```
grid.forceEditCell(td);
```

getCell

```
getCell(row, column)
```

Returns the cell as `td` DOM element specified by `row` index and `column` index.

Parameters

row

The row

column

The column

Return value

Returns the cell as `td` DOM element specified by `row` index and `column` index.

Example

```
var td = grid.getCell(1, 1);
```

getCellData

```
getCellData(row, column, name)
```

Returns the data having `name` key associated to a cell specified by `row` and `column`.

Parameters

row

The row of the cell

column

The column of the cell

name

The name of the property to retrieve

Return value

Returns the data having `name` key associated to a cell specified by `row` and `column`.

Example

```
var value = grid.getCellData(1, 2, 'test');
```

getCellValue

```
getCellValue(row, column)
```

Returns (string) the cell value specified by `row` index and `column` index.

Parameters

row

The row of the cell

column

The column of the cell

Return value

The cell value specified by `row` index and `column` index.

Example

```
var value = grid.getCellValue(2, 3);
```

getColsCount

```
getColsCount()
```

Return (number) the columns count.

Parameters

None

Return value

The columns count.

Example

```
var columns = grid.getColsCount();
```

getColumnIndexById

```
getColumnIndexById(columnId)
```

Returns the numeric index of a column specified by `columnId`, or -1 if not found.

Parameters

columnId

The column identifier

Return value

The numeric index of a column specified by `columnId`.

Example

```
var column = grid.getColumnIndexById('col1');
```

getColumnProps

```
getColumnProps(column)
```

Get the properties of a grid column specified by `column` index.

Parameters

column

The column index.

Return value

The properties (object) of a grid column.

Example

```
var props = grid.getColumnProps(1);
```

getRow

```
getRow(row)
```

Returns the row as `tr` DOM element specified by `row` index.

Parameters

row

The row index to retrieve

Return value

The row as `tr` DOM element specified by `row` index.

Example

```
var tr = grid.getRow(2);
```

getRowIndex

```
getRowIndex(tr)
```

Returns the numeric row index.

Parameters

tr

The DOM element row to retrieve the index for.

Return value

The numeric row index.

Example

```
var row = grid.getRowIndex(tr);
```

getRowsCount

```
getRowsCount()
```

Return (number) the rows count.

Parameters

None

Return value

The rows count.

Example

```
var rows = grid.getRowsCount();
```

getSelectedRow

```
getSelectedRow()
```

Returns the selected row `tr` DOM element or `null` if not found.

Parameters

None

Return value

Returns the selected row `tr` DOM element or `null` if not found.

Example

```
var tr = grid.getSelectedRow();
```

getToolbarButtonProps

```
getToolbarButtonProps(id)
```

Get the properties of the toolbar button identified by the given `id`.

Parameters

`id`

The button identifier

Return value

The properties (object) of a grid toolbar button.

Example

```
var props = grid.getToolbarButtonProps('open');
```

getValue

```
getValue()
```

Returns the grid value as JSON string.

Parameters

None

Return value

The grid data as JSON string.

Example

```
var data = grid.getValue();
```

hasError

```
hasError()
```

Returns `true` if the grid has at least one row or one cell marked as invalid or if the grid is marked as required but has not rows.

Parameters

None

Return value

`true` if the grid has at least one row or one cell marked as invalid or is empty while being required.

Example

```
if (grid.hasError())
{
    ...
}
```

groupRowsByColumn

```
groupRowsByColumn(column)
```

Scan the cells of the grid column specified by `column` index from top to bottom and merge the cells having the same value.

Parameters

column

The column index to scan

Return value

None

Example

isCellEditable

```
isCellEditable(row, column)
```

Returns true if a specified cell is editable.

Parameters

row

The row that contains the cell

column

The column that contains the cell

Return value

`true` if the specified cell is editable.

Example

```
var isEditable = grid.isCellEditable(5, 4);
```

isCellReadOnly

```
isCellReadOnly(row, column)
```

Returns true if a specified cell is read only.

Parameters

row

The row that contains the cell

column

The column that contains the cell

Return value

`true` if the specified cell is read only.

Example

```
var isReadOnly = grid.isCellReadOnly(5, 4);
```

isFunction

```
isFunction(name)
```

Returns `true` if a function specified by `name` exists (created with the Grid Designer UI).

Parameters

name

The function name to check for

Return value

`true` if the function specified by `name` exists.

Example

```
grid.isFunction('internal_fxn');
```

hasRowErrors

```
hasRowErrors(row)
```

Returns `true` if the specified row has any invalid cell.

Parameters

row

The row to check

Return value

`true` if a specified row has any invalid cell.

Example

isCellInvalid

```
isCellInvalid(row, column)
```

Returns true if the cell specified by `row` and `column` is marked as invalid (has error).

Parameters

row

The row that contains the cell

column

The column that contains the cell

Return value

`true` if the cell specified by `row` and `column` is marked as invalid.

Example

isEmpty

```
isEmpty()
```

Returns `true` if grid is empty.

Parameters

None

Return value

`true` if grid is empty.

Example

isReadOnly

```
isReadOnly()
```

Returns `true` if the grid is read only.

Alias for `isEnabled()`.

Parameters

None

Return value

`true` if the grid is read only.

Example

```
if (!grid.isReadOnly())
{
   ...
}
```

layout

```
layout()
```

Refresh the grid layout, called after a UI option has been changed.

This is called internally after updating the header, toolbar or toolbar buttons.

Parameters

None

Return value

None

Example

```
grid.layout();
```

moveup

```
moveup(tr)
```

Move the row specified by `tr` DOM element one position up. If tr not specified, move the selected row.

Parameters

tr

The row to move

Return value

None

Example

```
var tr = grid.getSelectedRow();
grid.moveup(tr);
```

movedown

```
movedown(tr)
```

Move the row specified by `tr` DOM element one position down. If tr not specified, move the selected row.

Parameters

tr

The row to move

Return value

None

Example

```
var tr = grid.getSelectedRow();
grid.movedown(tr);
```

remove

```
remove(tr)
```

Delete the row specified by `tr`. If tr not specified, delete the selected row.

Parameters

tr

The row to remove

Return value

None

Example

```
var tr = grid.getSelectedRow();
grid.remove(tr);
```

loadData

```
loadData(url, data)
```

Loads the grid content from a specified url. It triggers `dataloaded` when finished.

Parameters

url

The endpoint for data

data

The parameters to pass with the request

Return value

None

Example

removeAll

```
removeAll()
```

Delete all rows.

Parameters

None

Return value

None

Example

```
grid.removeAll();
```

setCellColor

```
setCellColor(row, column, color)
```

Set a cell background color. The color is a CSS string.

Parameters

row

The row that contains the cell

column

The column of the cell

color

The color to set; the color is a CSS string.

Return value

None

Example

```
grid.setCellColor(0, 0, 'red');
grid.setCellColor(0, 0, '#888');
```

selectRow

```
selectRow(tr)
```

Selects the row specified by the `tr` DOM element.

Parameters

tr

The row to select

Return value

None

Example

```
grid.selectRow(grid.getRow(0)); // select first row!
```

setCellData

```
setCellData(row, column, name, value)
```

Set a data `value` of any type, having the `name` key to a cell specified by `row` and `column`.

Parameters

row

The row of the cell

column

The column of the cell

name

The name of the property to set

value

The value to set

Return value

None

Example

```
var pt = { x:100, y:200 };
grid.setCellData( 0, 0, 'point', pt);
grid.setCellData( 0, 0, 'hasPoint', true);
```

setCellEditable

```
setCellEditable(row, column, status)
```

Set/Unset cell editable.

Parameters

row

The row of the cell

column

The column of the cell

status

Boolean to indicate editable status

Return value

None

Example

```
grid.setCellEditable(0, 0, false);
```

setCellInvalid

```
setCellInvalid(row, column, value)
```

Mark a cell as invalid if `value` is true.

Parameters

row

The row of the cell

column

The column of the cell

value

Cell validity indicator

Return value

None

Example

```
grid.setInvalidCell(0, 0, true);  // invalid
grid.setInvalidCell(0, 1, false); // valid
```

setCellReadOnly

```
setCellReadOnly(row, column, status)
```

Set/Unset cell specified by `row` index, `column` index as read only.

Parameters

row

The row of the cell

column

The column of the cell

status

Boolean to indicate read-only status

Return value

None

Example

```
grid.setCellReadOnly(0, 0, true);
```

setCellValue

```
setCellValue(row, column, value)
```

Set the cell value.

Parameters

row

The row of the cell

column

The column of the cell

value

The value to set

Return value

None

Example

```
grid.getCellValue(2, 3, 'John Doe');
```

setColumnReadOnly

```
setColumnReadOnly(column, status)
```

Set/Unset cells in a column specified by column `index` as read only.

Parameters

column

The column of the cell

status

Boolean to indicate read-only status

Return value

None

Example

```
grid.setColumnReadOnly(1, true);
```

setReadOnly

```
setReadOnly(readonly)
```

Set/Unset grid readonly.

Aliases for `disable()`, `enable()`.

Parameters

readonly

The read-only status for the grid

Return value

None

Example

```
grid.setReadOnly(true);
```

setRowInvalid

```
setRowInvalid(row, value)
```

Mark a row as invalid.

Parameters

row

The row to set as invalid.

value

Row validity indicator

Return value

None

Example

```
grid.setRowInvalid(1, true);
```

setRowReadOnly

```
setRowReadOnly(row, status)
```

Set/Unset a grid row read only.

Parameters

row

The row to set.

value

Row read-only indicator

Return value

None

Example

```
grid.setRowReadOnly(1, true);
```

setValue

```
setValue(value)
```

Sets the value of the grid. Value must be a stringified JSON.

Parameters

value

The new grid content

Return value

None

Example

```
grid.setValue('{"col1":[1,2,3],"col2":["John","Maria","Alice"]}');
```

unbindEvent

```
unbindEvent(eventname)
```

Unbind all binded event handlers for an event.

Parameters

eventName

The name of the event to unbind handlers for

Return value

Returns `true` for success, `false` if fails.

Example

```
grid.unbindEvent('focused');
```

updateEditCell

```
updateEditCell(hide)
```

Saves the value from the editor into grid cell; if `hide` boolean is true then close the editing input.

Parameters

hide

Flag to signal that input field should be destroyed, too.

Return value

None

Example

```
grid.updateEditCell(true);
```

updateToolbarButton

```
updateToolbarButton(id, props)
```

Update the properties of the toolbar button identified by the given `id`.

Parameters

`id`

The id of the toolbar button

`props`

The button properties to change

Return value

None

Example

```
grid.updateToolbarButton('btn1', { 'visible': true, 'text': 'New button' });
```

updateToolbar

```
updateToolbar(props)
```

Update the properties of the grid toolbar.

Parameters

`props`

The toolbar properties to change

Return value

None

Example

```
grid.updateToolbar({ 'visible': true, 'align': 'top' });
```

updateHeader

```
updateHeader(props)
```

Update the properties of the grid header.

Parameters

props

The header properties to change

Return value

None

Example

```
grid.updateHeader({ 'visible': true, 'compact': false });
```

Other input fields

File input

To create a file control:

```
var control = form_newControl(attrId, 'file', options);
```

The options include:

dnd

Boolean indicating if the control should be rendered as a drag-and-drop area or as a simple input file

Phone input

To create a phone control:

```
var control = form_newControl(attrId, 'phone', options);
```

The options include:

initialCountry

The ISO code of the default country to select

placeholderNumberType

The number type to use for the placeholder `[MOBILE|FIXED_LINE|FIXED_LINE_OR_MOBILE]`

autoPlaceholder

Set the input's placeholder to an example number for the selected country, and update it if the country changes `[polite|aggressive]`

Form callback functions

FUN_FORM_PRE_LOAD

```
fun_form_pre_load()
```

Callback function called before creating the form controls and filling the form values.

Parameters

None.

Return value

None.

Example

```
function fun_form_pre_load() {
    try {
        // request necesary data
        var result = form_invokePlugin({
            pluginId: 'publicis',
            command: 'NomLists',
            data: {}
        });
        var json = JSON.parse(result);

        // cache data for later use
        // ...
   }
   catch (e) {
       form_error(e);
   }
}
```

FUN_FORM_LOAD

```
fun_form_load()
```

Callback function called after the form controls have been initialized and the form values have been set.

Parameters

None.

Return value

None.

Example

```
function fun_form_load() {
    // init form values with defaults
    setElementKey(gfxGetAttrIdByLabel('userId'), getElementValue('username'));


    // init custom buttons
    form_displayCustomButtonBar({ ... });
}
```

FUN_FORM_POST_LOAD

```
fun_form_post_load()
```

Callback function called at the very end of the form load sequence.

Parameters

None.

Return value

None.

Example

FUN_FORM_DISPLAYCONTROLS

```
fun_form_displayControls(displayMode)
```

Callback function called after enabling/disabling the controls based on the display mode.

PARAMETERS

displayMode

The current display mode of the form, which can be one of the following values (both symbolic and numeric constants can be used):

```
0 = DISPLAY_NORMAL
1 = DISPLAY_READONLY
2 = DISPLAY_DISABLEDINPUTS
4 = DISPLAY_DISABLEDBUTTONS
```

RETURN VALUE

None.

EXAMPLE

FUN_FORM_PRE_LOADATTRS

```
fun_form_pre_loadattrs(data)
```

Callback function called before setting the attribute values inside the form.

Parameters

data

An object containing two objects: attributes and rights. The attributes object contains the values to set (each object consists of an attributeId-attributeValue pairing). The rights object contains information on weather the user can view and/or modify the values.

Return value

None.

Example

FUN_FORM_POST_LOADATTRS

```
fun_form_post_loadattrs(data)
```

Callback function called after setting the attribute values inside the form.

Parameters

data

An object containing two objects: attributes and rights. The attributes object contains the values to set (each object consists of an attributeId-attributeValue pairing). The rights object contains information on weather the user can view and/or modify the values.

Return value

None.

Example

FUN_FORM_PENDING_VALIDATE

```
fun_form_pending_validate()
```

Callback function called before actually starting the validation process.

Parameters

None.

Return value

None.

Example

FUN_FORM_PRE_VALIDATEATTRS

```
fun_form_pre_validateattrs(data)
```

Callback function called before starting the validation process.

Parameters

data

An object containing attributeId-attributeValue pairings that will be sent on the server for validation.

Return value

None.

Example

FUN_FORM_PRE_VALIDATEREPONSE

```
fun_form_pre_validatereponse(failedIds, failedReasons, failedCodes)
```

Callback function called immediately after receiving the validation response but before marking the failed values. Allows you to modify the attributes that will be marked as failed by adding or removing from the `failed*` arrays.

The arrays must be kept in sync, so if you add something to the `"failedIds"` you have to also add to the other two arrays as well.

PARAMETERS

failedIds

Array of attribute ids that have failed validation.

failedReasons

Array of reasons that describe the failure.

failedCodes

Array of failure codes.

RETURN VALUE

None.

EXAMPLE

```
function fun_form_pre_validatereponse(failedIds, failedReasons, failedCodes) {
            var date = gfxVal('date');
            var startDate = gfxVal('startDate');

            if (startDate.trim() !== '' && date.trim() !== '') {
                var sdate = form_parseDate(startDate);
                sdate.setHours(0);
                sdate.setMinutes(0);
                sdate.setSeconds(0);
                sdate.setMilliseconds(0);

                var adate = form_parseDate(date);
```

```
                adate.setHours(0);
                adate.setMinutes(0);
                adate.setSeconds(0);
                adate.setMilliseconds(0);


                if (sdate.getTime() < adate.getTime()) {
                    failedIds.push(gfxGetAttrIdByLabel('startDate'));
                    failedCodes.push('invalid');
                    failedReasons.push('Start date cannot be lower than the date.');
                }
            }
}
```

FUN_FORM_VALIDATE

```
fun_form_validate(result)
```

Callback function called immediately after after marking the failed values.

Parameters

result

An object containing the validation outcome.

The object looks like this:

```
{
    validation : {
        enable : true | false,
        outcome : {
            labeled : true | false,
            anonymous : true | false,
            format : true | false, // true if there are no format failures
            mandatory : true | false, true if there are no mandatory failures
            regex : true | false, // true if there are no regex failures
            formatRegex : true | false, // true if there are no format and regex
failures
            validate : true | false, // true if validation succeeded
            label : true | false, // true if the label is filled
        },
        fulltext : true | false,
    }
}
```

Return value

None.

Example

FUN_FORM_POST_VALIDATEREPONSE

```
fun_form_post_validatereponse(failedIds, failedReasons, failedCodes)
```

Callback function called immediately after after marking the failed values. You might use it to enable/disable the form buttons based on the validation state.

Parameters

failedIds

Array of attribute ids that have failed validation.

failedReasons

Array of reasons that describe the failure.

failedCodes

Array of failure codes.

Return value

None.

Example

```
function fun_form_post_validatereponse(failedIds, failedReasons, failedCodes) {
      // disable buttons if form not valid
      form_setCustomButtonEnabled('submit', failedIds.length === 0);
}
```

FUN_FORM_GET_VALIDATIONMESSAGE

```
fun_form_get_validationmessage(id, label, reason, code)
```

Callback function that offers the possibility to customize the validation messages that appear as tooltip on invalid controls.

Parameters

id

The invalid attribute's ID.

label

The invalid attribute's label.

reason

The default validation message.

code

The validation code.

Return value

The custom message. If `null`, `empty` or `undefined` is returned then the default validation message will be used.

Example

```
function fun_form_get_validationmessage(id, label, reason, code)
{
  form_info(id + ", " + label);

  switch (code)
  {
    case 'mandatory': // This field is mandatory. You must fill it.
      return 'Custom mandatory message';

    case 'length': // The value entered in this field is too long.
      if (label === 'myattribute')
        return 'The value exceeds the accepted size.';
      else
        return null;
```

```
    // case 'format': // Please check the validity of the data.

    // case 'listsize': // You have selected too many elements from this multiple
selection list.

    // case 'regex': // The value does not match the regular expression specified
for validation.

    // case 'emailsize': // You have selected too many e-mails.

    // case 'mandatorycolumn': // This grid contains mandatory cells. You must fill
them.


    default:

      return null;

  }

}
```

FUN_FORM_PRE_SUBMITFORM

```
fun_form_pre_submitform(formId, parentId, objectId)
```

Callback function called before submitting the form. Can stop the form submit if returned value is `false`.

Parameters

formId

The id of the current form

parentId

The id of the parent object

objectId

The id of the current object

Return value

The return value should be a boolean. If `false` then the form submit is stopped, otherwise the submit is allowed to continue.

Example

```
function fun_form_pre_submitform(formid, parentid, objectid) {
    generateDocumentNumber();
    return true;
}
```

In the example above a function that generates a number is called before allowing the form to submit.

You can also make certain validations and allow or stop the form submit accordingly.

FUN_FORM_CONFIRM_SUBMITFORM

```
fun_form_confirm_submitform(options)
```

Callback function called after `fun_form_pre_submit` and before actually submitting the form. Allows the display of a confirmation message before submitting the form.

Parameters

options

An object containing the command name, the context and subcontext of the call.

Return value

An object containing the title, message and buttons to display in the confirmation dialog. You can also specify a callback function to be called when one of the buttons is clicked.

Example

```
function fun_form_confirm_submitform(options) {
    switch(options.cmd) {
        case 'save':
        case 'saveClose':
        case 'zap':
            return {
                title : 'Save',
                message : 'Are you sure you want to save this ?',
                callback : function(e, key, btn) {
                    console.log('pressed : ' + btn);

                    switch (btn)
                    {
                        case Button.YES:
                            // do something
                            break;

                        case Button.NO:
```

```
                            // do something
                            break;

                        default:
                            break;
                    }
                }
            };

         default:
            break;
     }
}
```

```
function fun_form_confirm_submitform(options) {
    return {
        title : 'Save',
        message : 'Are you sure you want to save this ?',
        buttons : {
                yes : {
                    label : 'Valider',
                    className : 'btn-default',
                    callback : function(e) {
                        console.log('validate');
                     }
                },
                no : {
                     label : 'Fermer'
                }
            }
    };
}
```

FUN_FORM_POST_SUBMITFORM

```
fun_form_post_submitform(formId, parentId, objectId, success)
```

Callback function called after submitting the form.

Parameters

formId

The id of the current form

parentId

The id of the parent object

objectId

The id of the current object

success

Boolean indicating if the submit was successful or not

Return value

None.

Example

```
function fun_form_post_submitform(formId, parentId, objectId, success) {
    form_displayAlert({
        type : success ? 'success' : 'error',
        message : success ? 'Success message ...' : 'Error message ...'
    });
}
```

FUN_FORM_PRE_SENDTASK

```
fun_form_pre_sendtask(formId, parentId, objectId)
```

Callback function called before finishing a task. Can stop finish task if returned value is `false`.

This callback is called only for forms displayed in inbox.

Parameters

formId

The id of the current form

parentId

The id of the parent object

objectId

The id of the current object

Return value

The return value should be a boolean. If `false` then the task finish is stopped, otherwise the finish operation is allowed to continue.

Example

FUN_FORM_GET_EDITNATIVE_OPTIONS

```
fun_form_get_editnative_options(id)
```

Callback function called when an edit native operation is requested in order to get the overwrite options for it.

Parameters

id

The id of the document being edited.

Return value

Return a JSON object containing the version property with one of the following values: `overwrite | major | minor`.

Example

```
function fun_form_get_editnative_options(id) {
    // add check if the doc's parent id is the current process and compute edit native
options
    return {
        version : 'major' // minor or overwrite
    };
}
```

FUN_FORM_REFRESH_DOCS

```
fun_form_refresh_docs()
```

Callback function called when documents were added, edited, deleted, pinned, unpinned, pasted, rolled back or splitted.

Parameters

None.

Return value

None.

Example

```
function fun_form_refresh_docs() {
    // check existence of a certain document using form_hasDocuments
or form_getDocuments methods
}
```

FUN_FORM_PRE_ESCALATETASK

```
fun_form_pre_escalatetask(formId, parentId, objectId)
```

Callback function called before escalating a task. Can stop escalate task if returned value is `false`.

This callback is called only for forms displayed in inbox.

Parameters

formId

The id of the current form

parentId

The id of the parent object

objectId

The id of the current object

Return value

The return value should be a boolean. If `false` then the task escalate is stopped, otherwise the escalate operation is allowed to continue.

Example

FUN_FORM_GET_OCR_ATTRIBUTES

```
fun_form_get_ocr_attributes()
```

Callback function called in order to retrieve the list of attribute ids that can be used when performing OCR.

Parameters

None.

Return value

The function should return an array of valid attribute ids.

Example

```
var ocrAttrs = ['label', 'alfa', 'text']; // or ids


function fun_form_get_ocr_attributes() {
    var ids = [];
    for (var i = 0; i < ocrAttrs.length; i++)
        ids.push(gfxGetAttrIdByLabel(ocrAttrs[i]));


    return ids;
}
```

FUN_FORM_SET_OCR_ATTRIBUTES

```
fun_form_set_ocr_attributes(attributes)
```

Callback function called after an OCR text to attribute association (after setting the values that were retrieved through OCR).

Parameters

attributes

Array of objects containing attribute id to value pairs.

Return value

None.

Example

```
function fun_form_set_ocr_attributes(attributes) {
    form_info(attributes);
    // perform necessary actions post OCR value set
}
```

FUN_FORM_POST_SIGN_DOCUMENTS

```
fun_form_post_sign_documents(data)
```

Callback function called after a document sign performed in the sign viewer (according to the task configuration). This allows you to finish the current task after the user has signed all documents.

PARAMETERS

data

An object containing :

- `ids` - array of signed document ids
- `count` - number of documents still left to sign

RETURN VALUE

None.

EXAMPLE

```
function fun_form_post_sign_documents(data) {
    form_info('Signed : ');
    form_info(data.ids);
    form_info('No. documents left to sign : ' + data.count);


    if (data.count == 0)
        form_finish();
}
```

FUN_FORM_RESET

```
fun_form_reset()
```

Callback function called after a form rest was performed.

Parameters

None.

Return value

None.

Example

Services available through AJAX requests

ajax_syncServerRequestXML

```
function ajax_syncServerRequestXML(strServerUrl)
```

Invokes a service identified by the given URL.

Parameters

strServerUrl

Identifier for the invoked service.

Return value

Response in XML format from the invoked service.

Example

/API/SEARCH/ATTR?REQUEST

Request format

```
<Request> ::= <ConditionIds> & <Criteria>
<ConditionIds> ::= <IdAttr> "=" <valueAttr> | <IdAttr> "=" <valueAttr> "&"
<ConditionIds>
<Criteria> ::= <CriteriaObject> | <CriteriaObject> "&" <Criteria>
<CriteriaObject> ::= "processes=1" | "folders=1" | "documents=1" | "fiches=1" |
"pins=1"
<valueAttr> ::= attribute value
<IdAttr> ::= the ID of the attribute prefixed with DFAT
```

Example:

```
var aReq = "DFAT00010000001000000000000000000000000=siatel";
aReq += "&";
aReq += "folders=1&documents=1";
var xmlResponse = ajax_syncServerRequestXML( '/api/search/attr?' + aReq );
```

Response format

Returns the IDs of the objects (files, documents, processes, etc.)

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<search>
    <results>
       <result>
          <id>DOCF000100000005000000000000000100000000<id>
       </result>
       <result>
          <id> FLDR00010000000500000000000000000000000</id>
       </result>
    </results>
</search>
```

/API/GET?REQUEST

Request format

```
<request> ::= <param1> & <param2>
<param1> ::= "id=" <idObj>
<param2> ::= "mode=" <valueMode>
<valueMode> ::= "simple" | "full"
<idObj> ::= object identifier preceded by an object prefix
```

Example:

```
var aReq = "id=DOCF000100000005000000000000000100000000";
aReq += "&";
aReq += "mode=simple";
var xmlResponse = ajax_syncServerRequestXML( '/api/get?' + aReq );
```

Response format

Returns information about an object.

The full mode is currently not implemented.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<object>
    <objectid> DOCF000100000005000000000000000100000000
    </objectid>
    <objectlabel> test document </objectlabel>
    <objectdesc></objectdesc>
    <objectattrs>
       <objectattr>
          <attrid>DFAT000100000010000000000000000000000000</attrid>
          <attrvalue>Example</attrvalue>
       </objectattr>
    </objectattrs>
</object>
```

Information accessible from the form

Information accessible from the form :

userid

ID of the currently connected user.

username

User name of the currently connected user.

userrealname

Real name of the currently connected user.

search

Indicates whether the form is used in search mode.

multi

Indicates whether the form is used in multi-mode.

ext_taskName

Task name.

ext_processName

Process name.

ext_processDefName

Process definition name.

Example

```
var taskName = document.getElementById( "ext_taskName" ).value;
```

Deprecated APIs

This section lists the deprecated functions for forms; these should be replaced with Gargantua 8 APIs.

form_executeURL, form_refreshDocumentList, form_displayDocumentListToggle, form_displayDocumentList, form_displayDocumentListOnReturn, form_processAreaVerticalMaximize, form_processAreaResizeToggle, form_processAreaResize, form_setCustomButtonHidden