

GARGANTUA 8 PHP API

TABLE OF CONTENTS

GLOSSARY OF TERMS	5
INTRODUCTION	6
Object types	7
Object identification	8
GARGANTUA 7 REST API BASICS	13
Communicating with the Gargantua server	13
REST API access URL	13
REST API services.....	14
Object keys	17
GARGANTUA 7 PHP API	21
Low-level access	21
High-level access	22
Error handling.....	22
GARGANTUA 7 PHP API REFERENCE	23
The G7API() function	25
Parameters	25
Return value	25
Error codes	25
Example.....	25
Low-level PHP API.....	27
doRequest	27
doFileRequest.....	29
High-level PHP API.....	30
hasConnection	30
getConnection	31
doLogin.....	32
doLogout	33
checkAuthorization	34
executeServerScript	35
getThumbnail	36

OBJECT REFERENCE	37
Package com.siatel.defs.....	39
Interfaces	39
Package com.siatel.domain	77
Interfaces	77
Classes.....	108
 EXAMPLES	 200
Basic G7API example	200
Execute a simple search	202

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

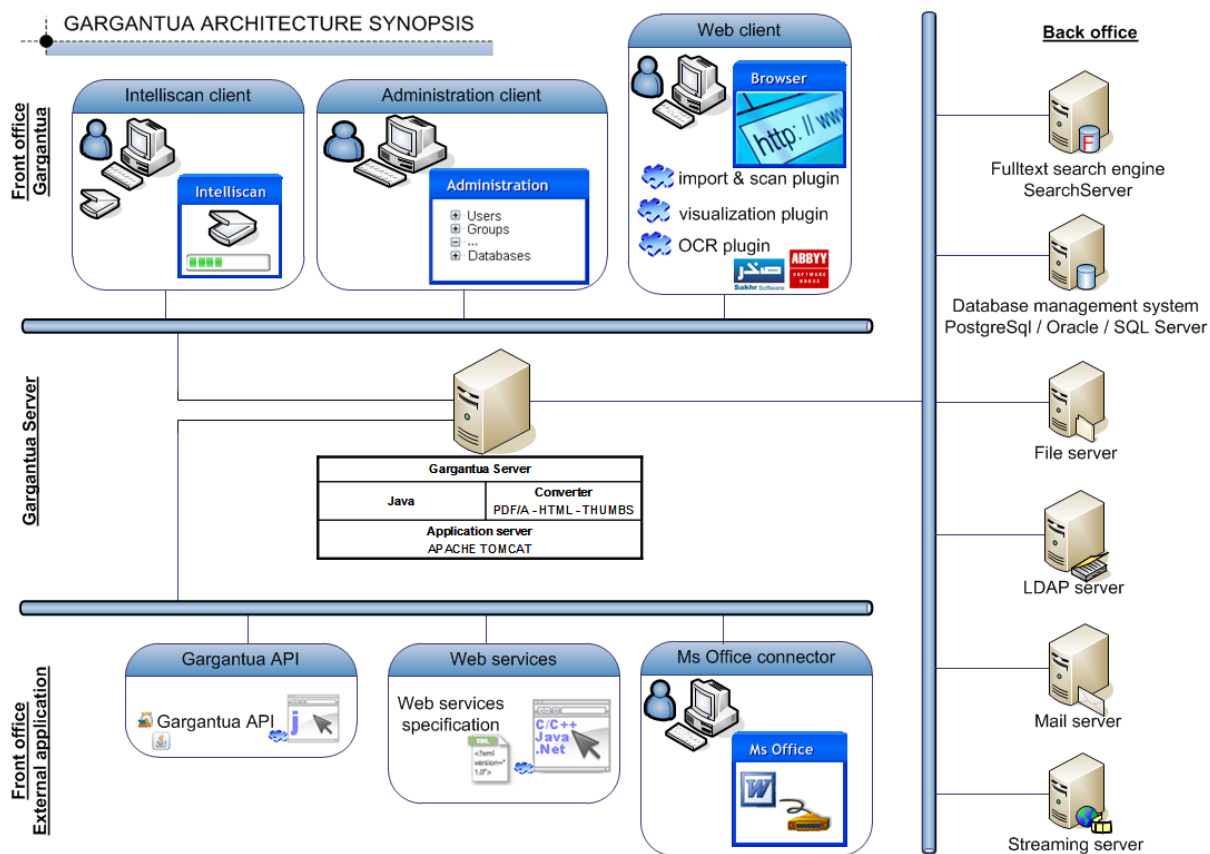
A data structure that is the result of executing a Gargantua server command.

Data transfer object

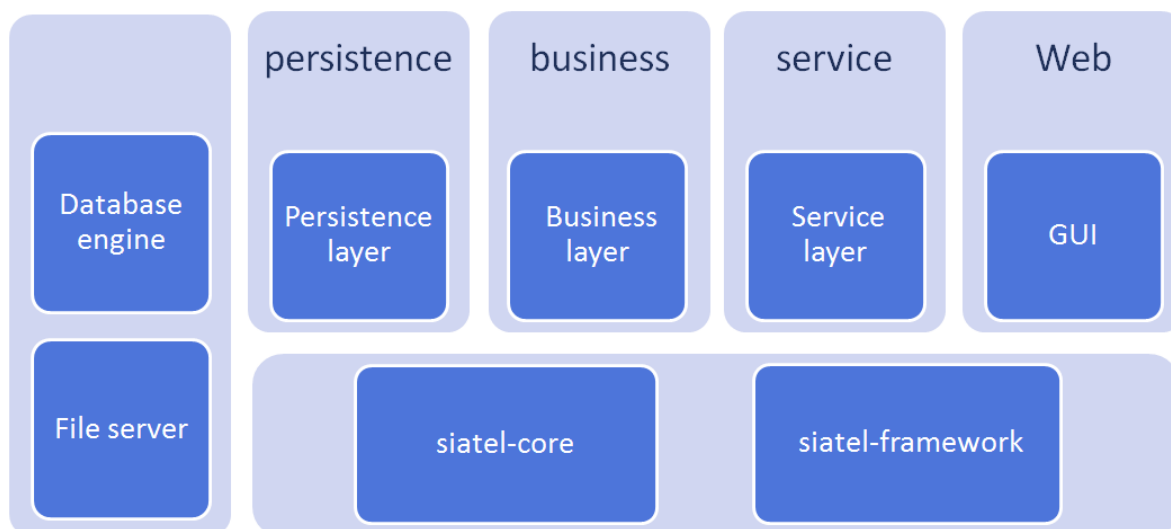
A map data structure used to pass parameters towards the Gargantua server or retrieve the result of a command.

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.

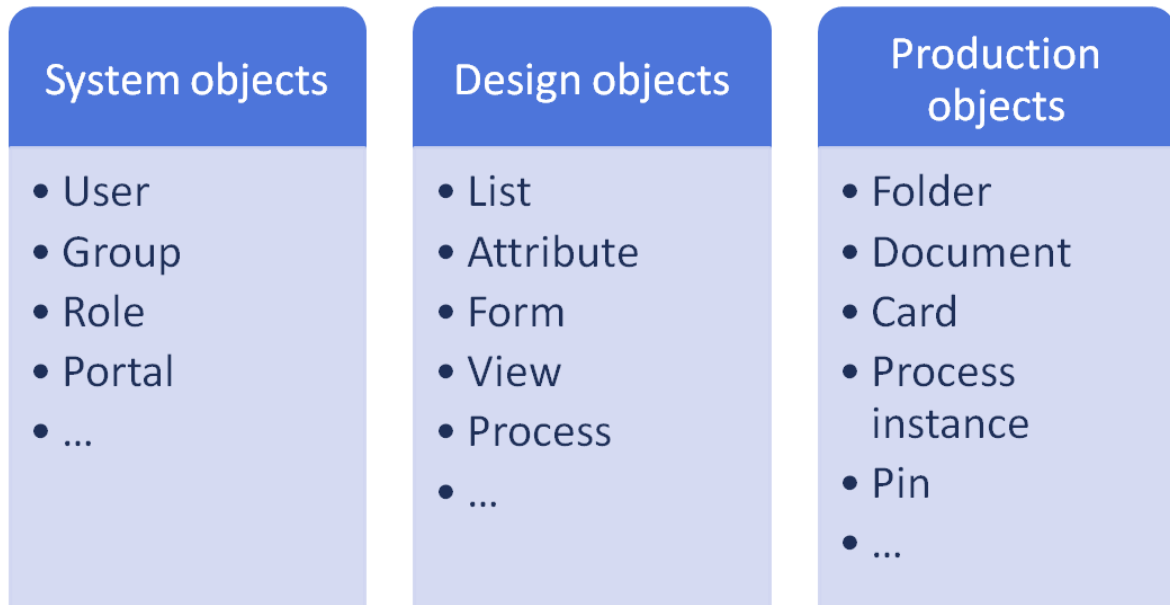
Design objects

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 assigned 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	Design objects	Production objects
• User - USER • Group - GROU • Role - ROLE • Portal - PORT • ...	• List - DFTY • Attribute - DFAT • Form - DFFR • View - DFVW • Process - DFPR • ...	• 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 be 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.

GARGANTUA 7 REST API BASICS

GARGANTUA 7 REST API BASICS

This document describes the REST API of the Gargantua server.

COMMUNICATING WITH THE GARGANTUA SERVER

The Gargantua server handles requests. A request comes in form of a command with an input data structure and yields an output data structure upon completion.

Gargantua server commands fall in several different categories, as described in the table below

Administration

Commands that will handle administration objects; these are objects that are server-wide scoped, like users, groups, databases, etc.

Design

Commands that will handle design objects; these objects are defined on a per-database schema, and are used for defining runtime behavior of the server, like attributes, forms, workflow processes, etc.

Production

Production commands handle all content stored within a Gargantua server. These can furthermore be divided into document management, workflow and search.

REST API ACCESS URL

The REST API is accessible on the `/api/rest` path relative to the base URL of the Gargantua web application.

Example: if the web application is accessible at `http://example.domain.com/siatel-web/` then the REST API is accessible at the `http://example.domain.com/siatel-web/api/rest`.

The generic form of an access URL is

```
http://example.domain.com:port/siatel-web/api/rest/<service>/<command>/<object>/<extra>
```

<service>

The name of the service that must handle the request; available services are : `auth`, `admin`, `design`, `ged`, `flow`, `search`, `security`, `storage` and `misc`; refer to the next section for details for each service.

<command>

The name of the command to be executed, normally a verb; the command set is service and object dependent, and are described individually; here are some examples : `login`, `logout`, `create`, `update`, `delete`, etc.

<object>

The target object type of the call command request; these, too are service dependent; for instance, the ged service will handle object types folder, document, etc.

<extra>

An extra element for commands that require it; this extra element is documented for commands that require it.



Depending on the service and command involved, the object type may be optional, as well as the extra element.

Below are a couple of example of URLs :

```
http://example.domain.com:port/siatel-web/api/rest/auth/login/
```

```
http://example.domain.com:port/siatel-web/api/rest/ged/create/folder
```

```
http://example.domain.com:port/siatel-web/api/rest/ged/read/document
```

REST API SERVICES

The list below briefly describes the available services for the REST API

auth

handles all requests related to a user session (authentication, termination, etc.)

admin

handles all request on administration objects

design

handles all requests on design objects

ged

handles document management related requests; please note that documents within workflow processes are also handled by this service

flow

handles all task and process related requests; process content is handled by the ged service

search

handles search requests; this include all types of search available on the server

security

handles security management requests for all Gargantua objects

storage

handles file storage requests, most notably storing and retrieveing files

misc

handles various commands that do not fall in any of the above categories, such as sending an email or retrieving pdf renditions of a document

Each request sent to a Gargantua server must include a connection token in the HTTP header, using field name 'conn', except for the authentication command that does not require this token, but returns it.

The request body is the input data structure, in JSON format; for commands that require it, a file can be attached to the request (requires the POST method).

The request outcome is the response body, also in JSON format.

Here are some examples (for complete details, see each commands individually)

This section describes the input and output JSON-formatted data structures

INPUT DATA STRUCTURE

The input data structure uses the following keys; the combimation of these keys are command dependent and are described for each command individually.

flags

`long`; specifies the command flags; see each command for details.

offset

`int`; an offset for the starting element of an enumeration command's list of results; usually optional, defaults to `0`.

count

`int`; the maximum number of elements to be returned by an enumeration command; usually optional, defaults to `-1` which represents 'no limit'.

criteriaIds

`array`; an array of filter criteria for an enumeration; see each command for details and possible filtering criteria.

object

`object`; a Gargantua object.

objectId

`string`; the identifier (id) of a Gargantua object.

objectLabel

`string`; the name (label) of a Gargantua object.

parentId

string; the identifier of the parent object when working with nested objects; for example when creating a document the **parentId** specifies a folder or a process.

targetId

string; the identifier of a destination for a move or copy-paste command.

afterId

string; the identifier of a position reference within a container for a reposition command.

user

string; the login name of the user when opening a session (login command).

pass

string; the user password when opening a session (login command).

For example, a read user command will send a request like the one below:

```
{
  "flags":1080933179801481247,
  "object": {
    "prefix":"USER",
    "id":"USER00000000000D0000000000000000000000"
  }
}
```

OUTPUT DATA STRUCTURE

The output data structure has the following keys

result

string; the outcome of a command call, **"ok"** in case of success, **"fail"** otherwise, in which case the **"error"** key provides additional details.

dtoOut

object; the result(s) of a command call; this is actually the payload of the server response to the command request; see each command for details.

error

string; details provided in case the command call did not succeed; usually a java exception stacktrace.

When an object or list of objects is returned, each JSON object mapped to a Gargantua object will use the keys described in [this section](#).

For example the response of a read attribute command would look like this:

```
{
  "result": "ok",
  "dtoOut": {
    "object": {
      "prefix": "DFAT",
      "id": "DFAT00020000007600000000000000000000000000",
      "typeId": "DFTY000200000001000000000000000000000000",
      "value": null,
      "label": "string attribute",
      "ownerId": "USER000000000001000000000000000000000000",
      "flags": 0,
      "dateC": 1493106793024,
      "dateM": 1493106793024,
      "description": "",
      "data": null,
      "packages": []
    }
  }
}
```

DATA TYPES

Data types are restricted to the data types allowed by the JSON format; the following conversions must be applied to higher level language data types

timestamps

Long (64 bit) values that represent a Java time, also known as Unix time (01/01/1970 00:00:00 UTC based). For example the `dateC` and `dateM` fields of most Gargantua objects.

flags

Long (64 bit) values representing bitwise combinations on command options or Gargantua object properties depending on the command context; the `Flag` java class documentation (javadoc) details the significance for each bit in the 0-62 range.

strings

String values in UTF-8 encoding.

OBJECT KEYS

prefix

`string`; first 4 characters of a Gargantua object id; indicates the object type; used to indicate to a command the object type when an object id is not available (for example retrieving an object by label); always the first key inside an `"object"` key.

id

string; the object identifier.

label

string; the name of the object; limited to 256 characters.

dateC

long; object creation timestamp.

dateM

long; object last modification timestamp.

flags

long; object flags

ownerId

string; id of the user who is the owner of the object.

formId

string; the identifier of the object form (for folders, pins, documents, processes).

attributes

array; the array of user defined attributes of an object (for folders, pins, documents, processes); each element in this array is a subsequent Gargantua 'attribute' object and consequently bears some of the keys described in this section.

typeId

string; the type identifier of an attribute

value

string, **number**, **boolean** or **null**; the value of an attribute; for date/timestamp attributes, the value (if non-null) is numeric and represents a **timestamp**.

fileId

string; a stored file's identifier

description

string; a description field for administration and design objects.

packages

array; an array of Gargantua "package" objects to which administration and design objects may be assigned to.

data

string; a field to hold object custom data for some administration and design objects; used by **attribute** and **assignment** objects.

GARGANTUA 7 PHP API

GARGANTUA 7 PHP API

The Gargantua 7 PHP API relies on the more complex REST API and provides a high-level access as well as low-level access directly through the REST API commands.

The Gargantua 7 PHP API provides some common functionalities as callable methods, which rely on the low-level REST API, and is implemented as a simple PHP class, named G7API.

LOW-LEVEL ACCESS

Low-level access is provided via the `doRequest` method.

The `doRequest` method takes a variable number of parameters; the first parameters which may be up to four are :

the service name

Maps to the service name used by the REST API

the command name

Maps to the command name used by the REST API

the object type

Maps to the object type used by the REST API

the secondary command

Maps to the extra element used by the REST API

The last parameter is always the command payload, or parameters; this is always provided as a PHP array which will be translated to JSON format for use by the server.

Below is an example of such request.

```
$G7API = G7API(
    $G7_url,
    $G7_user,
    $G7_pass);

$search = $G7API->doRequest(
    'search',
    'advanced',
    array(
        'flags' => 'SEARCH_DOCUMENTS | SEARCH_FOLDERS | SEARCH_GED',
```

```
'databaseLabel' => 'test database',  
'count' => 1,  
'expression' => '\some attribute\' = \'some value\'',  
'attributes' => array (  
    'object.label', 'object.fileId'  
)  
);
```

HIGH-LEVEL ACCESS

The Gargantua 7 PHP API also provides some high-level access functionalities, that are detailed in the following sections.

Please note that all high-level functionalities are executed as a sequence of low-level REST API calls to the Gargantua 7 server; for instance, to retrieve the thumbnail for a given document, first a check is made to ensure that the thumbnail exists; if it does not, it is created, then it is downloaded and returned to the caller.

ERROR HANDLING

GARGANTUA 7 PHP API REFERENCE

THE G7API() FUNCTION

```
function G7API($serverUrl [, $userName, $password])
```

The G7API function creates a G7API object for later use; it will either create a new one or bind to an existing connection previously opened by a similar call.

PARAMETERS

serverUrl

Required.

The address of the target Gargantua 7 server

userName

Optional.

The user name for authentication.

password

Optional.

The password for the user.

When creating the initial object, all three parameters are required.

If binding to an existing connection (connection token is saved for the active session), only the first parameter is used; if provided the last two are ignored

RETURN VALUE

The G7API object that can be used to communicate with the Gargantua 7 server.

ERROR CODES

Errors are always thrown as exceptions. The G7API function will throw the same exceptions as the [doLogin](#) method.

EXAMPLE

Content of file `config.php`

```
<?php
    $G7_url = 'example.domain.org';
    $G7_user = 'my-user';
    $G7_pass = 'my-password';

?>
```

Content of file [test.php](#)

```
<html>
    <head>
    </head>
    <body>
        <?php
            include_once(dirname(__FILE__) . '/config.php');
            include_once(dirname(__FILE__) . '/modules/gargantua.class.php');

            $G7API = G7API(
                $G7_url,
                $G7_user,
                $G7_pass);

            if ($G7API->hasConnection())
            {
                ?> <span>Connection successfull.</span> <?php
            }
            else
            {
                ?> <span>Connection failure </span> <?php
            }
        ?>
    </body>
</html>
```

LOW-LEVEL PHP API

The Gargantua 7 low-level PHP API provides two methods that directly access the REST API; one that retrieves a server response JSON, and the other that retrieves a file.

DOREQUEST

```
mixed doRequest(string $service, string $command [, string $object [, string $extra]], array $parameters)
```

Requests the execution of a command on the Gargantua 7 server.

PARAMETERS

service

Required.

The name of the service that must handle the request; available services are : auth, admin, design, ged, flow, search, security, storage and misc; refer to the Gargantua 7 REST API section for details on each service

command

Required.

The name of the command to be executed, normally a verb; the command set is service- and object-dependent, and are described individually; here are some examples : [login](#), [logout](#), [create](#), [update](#), [delete](#), etc.

object

Optional.

The target object type of the call command request; these, too are service dependent; for instance, the ged service will handle object types folder, document, etc.

extra

Optional.

An extra element for commands that require it; this extra element is documented for commands that require it.

parameters

Optional.

An array that is translated to a JSON object that conforms to the DTO input structure of every Gargantua 7

REST API command.

RETURN VALUE

A PHP object, usually an array that has been produced by parsing the JSON response sent by the server using `json_decode`.

ERROR CODES

Errors are always thrown as exceptions. See the Gargantua 7 REST API for specific error codes for each command.

DOFILEREQUEST

```
mixed doFileRequest(string $service, string $command [, string $object  
[, string $extra]], array $parameters)
```

Requests the execution of a command on the Gargantua 7 server and returns a binary stream representing content (for instance a document's file or thumbnail).

PARAMETERS

RETURN VALUE

A binary stream as a result of the given command.

For instance, this may be a document's file, a thumbnail or the PDF version for a given document.

ERROR CODES

Errors are always thrown as exceptions. See the Gargantua 7 REST API for specific error codes for each command.

HIGH-LEVEL PHP API

HASCONNECTION

```
boolean hasConnection()
```

Checks if the G7API object is connected to a Gargantua 7 server.

PARAMETERS

None.

RETURN VALUE

`true` if the G7API object is bound to a Gargantua 7 server.

ERROR CODES

Errors are always thrown as exceptions. See the Gargantua 7 REST API for specific error codes for each command.

GETCONNECTION

```
string getConnection()
```

Retrieves the connection token for the G7API object.

PARAMETERS

None.

RETURN VALUE

The connection token for the given G7API object, or the empty string if the G7API object is not bound.

ERROR CODES

Errors are always thrown as exceptions. See the Gargantua 7 REST API for specific error codes for each command.

DOLOGIN

```
void doLogin()
```

Performs a login using the credentials provided when instantiating the G7API object.

PARAMETERS

None.

RETURN VALUE

None.

ERROR CODES

Errors are always thrown as exceptions. See the Gargantua 7 REST API for specific error codes for each command.

DOLOGOUT

```
void doLogout()
```

Performs a logout operation on the given G7API object. To be used again, a new [doLogin](#) method call is required.

PARAMETERS

None.

RETURN VALUE

None.

ERROR CODES

Errors are always thrown as exceptions. See the Gargantua 7 REST API for specific error codes for each command.

CHECKAUTHORIZATION

```
void checkAuthorization($userName, $password)
```

Checks that the username and password combination provided is valid. This method does not return anything, instead, it throws an exception if the username and password combination is invalid.



Please note that this method will not give a detailed error code; this is by design.

PARAMETERS

userName

Required.

The username to check.

password

Required.

The password to check.

RETURN VALUE

None.

ERROR CODES

Errors are always thrown as exceptions. See the Gargantua 7 REST API for specific error codes for each command.

EXECUTESERVERSCRIPT

```
mixed executeServerScript($databaseLabel, $scriptLabel, $scriptFunction, $parameters)
```

Executes a server script, then retrieves the JSON result.

PARAMETERS

databaseLabel

Required.

The name of the database where the script is located.

scriptLabel

Required.

The name of the script to execute.

scriptFunction

Required.

The name of the function to execute.

parameters

Required.

The script parameters.

RETURN VALUE

The `result` object from the response DTO from server.

ERROR CODES

Errors are always thrown as exceptions. See the Gargantua 7 REST API for specific error codes for each command.

GETTHUMBNAIL

```
binary getThumbnail($fileId, $sizeX, $sizeY)
```

Retrieves the thumbnail for given file.



Please note that the id provided as a parameter is a file ID and not an object ID.

PARAMETERS

fileId

Required.

The identifier of the file for which to retrieve the thumbnail.

sizeX

Required.

Horizontal size of the thumbnail.

sizeY

Optional.

Vertical size of the thumbnail.

RETURN VALUE

The thumbnail file itself, as a binary stream.

ERROR CODES

Errors are always thrown as exceptions. See the Gargantua 7 REST API for specific error codes for each command.

OBJECT 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
```

DatedEx 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 to 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 they 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 by examining the object identification string. These string identifiers have always the same character length, thus enabling consistent handling, and always hold at 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 that 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"](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 formatted 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 attribute 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 , CMTF	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 extended 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:

DFVE000100000001**00000000**000000000000000000 is the parent id for root view elements of the DFVW000100000001000000000000000000000000 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 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 of 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 hierarchical structure 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 its 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 documents 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 documents 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	<u>ICON</u>
public static final long	<u>DATE_S</u>
public static final long	<u>DATE_E</u>
public static final long	<u>DATE_D</u>
public static final long	<u>DATE_X</u>
public static final long	<u>OVERDUE</u>
public static final long	<u>EXPIRED</u>
public static final long	<u>REFCOUNT</u>
public static final long	<u>PRIORITY</u>
public static final long	<u>SECURITY_PROFILE</u>
public static final long	<u>RENDER_SECURITY</u>
public static final long	<u>UUID</u>
public static final long	<u>TAGS</u>
public static final long	<u>DATE_AR</u>
public static final long	<u>EMAIL</u>
public static final long	<u>REAL_NAME</u>
public static final long	<u>DIRECTORY</u>
public static final long	<u>PASSWORD</u>

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

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

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

long	
public static final long	<u>BY LABEL ASC</u>
public static final long	<u>BY LABEL DESC</u>
public static final long	<u>BY DATEC ASC</u>
public static final long	<u>BY DATEC DESC</u>
public static final long	<u>BY DATEM ASC</u>
public static final long	<u>BY DATEM DESC</u>
public static final long	<u>BY DATES ASC</u>
public static final long	<u>BY DATES DESC</u>
public static final long	<u>BY DATED ASC</u>
public static final long	<u>BY DATED DESC</u>
public static final long	<u>BY PRIO ASC</u>
public static final long	<u>BY PRIO DESC</u>
public static final long	<u>COMMENT CONTENT</u>
public static final long	<u>OWNER NAME</u>
public static final long	<u>HIGHER ID</u>
public static final long	<u>USER ALL</u>
public static final long	<u>USER UPDATE ALL</u>
public static final long	<u>GROUP ALL</u>

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

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

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

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

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

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

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

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

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

long	
public static final long	<u>REORDER_LABEL_DESC</u>
public static final long	<u>EMAIL_NOTIFICATION</u>
public static final long	<u>CERT_ISSUED_BY</u>
public static final long	<u>CERT_SERVER</u>
public static final long	<u>CERT_FLAG_VALID_2</u>
public static final long	<u>CERT_IDENTITY_PREFIX</u>
public static final long	<u>CERT_IDENTITY_NO</u>
public static final long	<u>CERT_IDENTITY</u>
public static final long	<u>CERT_GEN_CACERT</u>
public static final long	<u>CERTIFICATE_ALL</u>
public static final long	<u>IS_JOB_SCAN</u>
public static final long	<u>IS_JOB_IMPORT</u>
public static final long	<u>IS_JOB_EMAIL</u>
public static final long	<u>IS_JOB_LAD_RAD</u>
public static final long	<u>IS_JOB_SIGN</u>
public static final long	<u>IS_JOB_SIGN_MANDATORY</u>
public static final long	<u>SIMPLE_DEBUG</u>
public static final long	<u>ENABLE_DOCUMENT_ACCESS</u>

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

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 = 1080933179801481247L
```

```
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 = 864716417222582298L
```

```
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 = 34235184362047L
```

```
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 = 2071656151656496847L
```

```
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 ($1 \ll 1$), database has simple security support

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

[Database#getFlags\(\)](#): Security mode 4 ($2 \ll 1$), database has medium security support

```
public static final long SECURITY_MODE_COMPLEX = 6L
```

[Database#getFlags\(\)](#): Security mode 6 ($3 \ll 1$), 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 default 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	<u>DATABASE</u>
public static final java.lang.String	<u>GROUP_DATABASE</u>
public static final java.lang.String	<u>USER_DATABASE</u>
public static final java.lang.String	<u>ROLE_DATABASE</u>
public static final java.lang.String	<u>STORAGE_AREA</u>
public static final java.lang.String	<u>DATABASE_STORAGE_RULE</u>
public static final java.lang.String	<u>GROUP_PORTAL</u>
public static final java.lang.String	<u>USER_PORTAL</u>
public static final java.lang.String	<u>ROLE_PORTAL</u>
public static final java.lang.String	<u>PLANNER</u>
public static final java.lang.String	<u>PLANNER_DAY</u>
public static final java.lang.String	<u>RESOURCE</u>
public static final java.lang.String	<u>SCRIPT</u>
public static final java.lang.String	<u>SECURITY_PROFILE</u>
public static final java.lang.String	<u>MAIL_TEMPLATE</u>
public static final java.lang.String	<u>DOCUMENT_TEMPLATE</u>
public static final java.lang.String	<u>FOLDER_PROFILE</u>
public static final java.lang.String	<u>COMPOSITION_RULE</u>
public static final	<u>SEARCH_TEMPLATE</u>

java.lang.String	
public static final java.lang.String	<u>USER_SEARCHTEMPLATE</u>
public static final java.lang.String	<u>GROUP_SEARCHTEMPLATE</u>
public static final java.lang.String	<u>ROLE_SEARCHTEMPLATE</u>
public static final java.lang.String	<u>ASYNC_JOB</u>
public static final java.lang.String	<u>EXTENDED_PROPERTY</u>
public static final java.lang.String	<u>OBJECT_EXTENDED_PROPERTY</u>
public static final java.lang.String	<u>CERTIFICATE</u>
public static final java.lang.String	<u>TYPE</u>
public static final java.lang.String	<u>GRID</u>
public static final java.lang.String	<u>ATTRIBUTE</u>
public static final java.lang.String	<u>FORM</u>
public static final java.lang.String	<u>FORM_ATTRIBUTE</u>
public static final java.lang.String	<u>VIEW</u>
public static final java.lang.String	<u>VIEW_ELEMENT</u>
public static final java.lang.String	<u>GROUP_VIEW</u>
public static final java.lang.String	<u>USER_VIEW</u>
public static final java.lang.String	<u>ROLE_VIEW</u>
public static final java.lang.String	<u>PROCESS_DEF</u>

public static final java.lang.String	<u>SCRIPT2</u>
public static final java.lang.String	<u>SCRIPT2_VERSION</u>
public static final java.lang.String	<u>COMPOSITION_RULE_VERSION</u>
public static final java.lang.String	<u>FORM_VERSION</u>
public static final java.lang.String	<u>PROCESS_DEF_VERSION</u>
public static final java.lang.String	<u>IS_JOB_VERSION</u>
public static final java.lang.String	<u>MODEL_OCR_VERSION</u>
public static final java.lang.String	<u>VIEW_VERSION</u>
public static final java.lang.String	<u>VIEW_ELEMENT_VERSION</u>
public static final java.lang.String	<u>IS_JOB</u>
public static final java.lang.String	<u>MODEL_OCR</u>
public static final java.lang.String	<u>GROUP_ISJOB</u>
public static final java.lang.String	<u>ROLE_ISJOB</u>
public static final java.lang.String	<u>USER_ISJOB</u>
public static final java.lang.String	<u>CLASS_ELEMENT</u>
public static final java.lang.String	<u>PROCESS</u>
public static final java.lang.String	<u>TASK</u>
public static final java.lang.String	<u>FOLDER</u>
public static final	<u>FOLDER_VALIDATIONCYCLE</u>

java.lang.String	
public static final java.lang.String	<u>FOLDER_MYDOCS</u>
public static final java.lang.String	<u>FOLDER_WORKSPACE</u>
public static final java.lang.String	<u>FOLDER_IN_WORKSPACE</u>
public static final java.lang.String	<u>PIN_DOCF</u>
public static final java.lang.String	<u>PIN_DOCP</u>
public static final java.lang.String	<u>PIN_DOCW</u>
public static final java.lang.String	<u>FILE_DOCF</u>
public static final java.lang.String	<u>FILE_DOCP</u>
public static final java.lang.String	<u>FICHE_DOCF</u>
public static final java.lang.String	<u>FICHE_DOCD</u>
public static final java.lang.String	<u>FICHE_DOCV</u>
public static final java.lang.String	<u>FICHE_DOCW</u>
public static final java.lang.String	<u>FILE_DOCD</u>
public static final java.lang.String	<u>FILE_DOCV</u>
public static final java.lang.String	<u>FILE_DOCW</u>
public static final java.lang.String	<u>FICHE_DOCP</u>
public static final java.lang.String	<u>REF_DOCF</u>
public static final java.lang.String	<u>REF_DOCP</u>

public static final java.lang.String	<u>REF_FICHEF</u>
public static final java.lang.String	<u>REF_FICHEP</u>
public static final java.lang.String	<u>FOLDER_NOTE</u>
public static final java.lang.String	<u>PROCESS_NOTE</u>
public static final java.lang.String	<u>DOC_NOTE_BASE</u>
public static final java.lang.String	<u>FICHEF_NOTE</u>
public static final java.lang.String	<u>DOCP_NOTE</u>
public static final java.lang.String	<u>FICHEP_NOTE</u>
public static final java.lang.String	<u>DOCF_VERSION</u>
public static final java.lang.String	<u>DOCP_VERSION</u>
public static final java.lang.String	<u>TASK_DEF</u>
public static final java.lang.String	<u>DELEGATION</u>
public static final java.lang.String	<u>ARCHIVE_FOLDER</u>
public static final java.lang.String	<u>ARCHIVE_PIN</u>
public static final java.lang.String	<u>ARCHIVE_FILE</u>
public static final java.lang.String	<u>INTERVENTION_REPORT</u>
public static final java.lang.String	<u>SERVER_REPORT</u>
public static final java.lang.String	<u>FOLDER_COMMENT</u>
public static final	<u>FOLDER_COMMENT_ARCH</u>

java.lang.String	
public static final java.lang.String	<u>DOCF COMMENT</u>
public static final java.lang.String	<u>DOCF COMMENT ARCH</u>
public static final java.lang.String	<u>PROCESS COMMENT</u>
public static final java.lang.String	<u>DOCP COMMENT</u>
public static final java.lang.String	<u>VALIDCYCLE TEMPLATE</u>
public static final java.lang.String	<u>VALIDCYCLE</u>
public static final java.lang.String	<u>VALIDCYCLE STAGE TEMPLATE</u>
public static final java.lang.String	<u>VALIDCYCLE STAGE</u>
public static final java.lang.String	<u>VALIDCYCLE STAGE DATA TEMPLATE</u>
public static final java.lang.String	<u>VALIDCYCLE STAGE DATA</u>
public static final java.lang.String	<u>VALIDCYCLE STAGE TEMPLATE NOTIFICATION</u>
public static final java.lang.String	<u>VALIDCYCLE STAGE NOTIFICATION</u>
public static final java.lang.String	<u>EXPIRABLE DOWNLOAD LINK</u>
public static final java.lang.String	<u>NOTIFICATION DEFINITION</u>
public static final java.lang.String	<u>NOTIFICATION</u>
public static final java.lang.String	<u>FAVORITE</u>
public static final java.lang.String	<u>OBJECT LINK</u>
public static final java.lang.String	<u>PACKAGE</u>

```
public static final
java.lang.String
```

[JUKE](#)

Field detail :

```
public static final String FILE = "FILE"
```

File id prefix.

Id structure is

FILEDDDDXXXXXXXXXXXXXXXXAAAAAABBBBBBBBCCCCCCCC

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

USER0000XXXXXXXX0000000000000000000000000000

where XXXXXXXX is the hexadecimal representation of the user numeric id.

```
public static final String USER_DELEGATION = "DLGT"
```

User object prefix.

Id structure is

DLGT0000XXXXXXXX0000000000000000000000000000

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

USPR0000XXXXXXXX0000000000000000000000000000

where XXXXXXXX is the hexadecimal representation of the user profile numeric id.

```
public static final String GROUP = "GROU"
```

Group object prefix.

Id structure is

```
GROU0000XXXXXXXX000000000000000000000000
```

where XXXXXXXX is the hexadecimal representation of the group numeric id.

```
public static final String ROLE = "ROLE"
```

Role object prefix.

Id structure is

```
USER0000XXXXXXXX000000000000000000000000
```

where XXXXXXXX is the hexadecimal representation of the role numeric id.

```
public static final String MANAGER = "MGRR"
```

Manager object prefix.

Id structure is

```
MGRR0000XXXXXXXX000000000000000000000000
```

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:

```
ASUDDDDDDXXXXXXXXYYYYYYYY0000000000000000
```

,

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

```
ASGDDDDDDXXXXXXXX0000000000000000000000
```

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

```
ASUDDDDDDXXXXXXXXX000000000000000000000000
```

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

```
ASRDDDDDDXXXXXXXXX000000000000000000000000
```

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

```
AREA0000XXXXXXXXX000000000000000000000000
```

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

```
RULE0000XXXXXXXXX000000000000000000000000
```

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

```
ASUP0000XXXXXXXXXXYYYYYYYYY0000000000000000
```

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

```
PLAN0000XXXXXXXX000000000000000000000000
```

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

```
RSRC0000XXXXXXXX000000000000000000000000
```

where XXXXXXXX is the hexadecimal representation of the resource numeric id.

```
public static final String SCRIPT = "SCPT"
```

Script object prefix.

Id structure is

```
SCPT0000XXXXXXXX000000000000000000000000
```

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

```
SECPDDDDXXXXXXXXX000000000000000000000000
```

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

```
MTPLDDDDXXXXXXXXX000000000000000000000000
```

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

```
DTPLDDDDXXXXXXXXX000000000000000000000000
```

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

```
FLPRDDDDXXXXXXXXX000000000000000000000000
```

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

```
CPRUDDDDXXXXXXXXX000000000000000000000000
```

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

```
SRCHDDDDXXXXXXXXX000000000000000000000000
```

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

```
JASY0000XXXXXXXX000000000000000000000000
```

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

```
DFATDDDDXXXXXXXXX000000000000000000000000
```

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

```
DFFRDDDDXXXXXXXXX000000000000000000000000
```

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

```
ASGVDDDDXXXXXXXXXYYYYYYYYY0000000000000000
```

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

```
DFVWDDDDXXXXXXXXX000000000000000000000000
```

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

```
DFVEDDDDDXXXXXXXXXYYYYYYYYY0000000000000000
```

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

```
ASGVDDDDXXXXXXXXXXXXXXXXXXXX0000000000000000
```

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

```
ASUVDDDDXXXXXXXXXXXXXXXXXXXX0000000000000000
```

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

```
ASRVDDDDXXXXXXXXXXXXXXXXXXXX0000000000000000
```

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

```
DFPRDDDDXXXXXXXXXXXX000000000000000000000000
```

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.

[illegible]

```
public static final String SCRIPT2_VERSION = "VDSC"
```

VDSCDDDDXXXXXXXXXXYYYYYYY0000000000000000

```
public static final String COMPOSITION_RULE_VERSION = "VDCR"
```

VDCRDDDDXXXXXXXXXXYYYYYYY0000000000000000

```
public static final String FORM_VERSION = "VDFR"
```

VDFRDDDDXXXXXXXXXXYYYYYYYY0000000000000000

```
public static final String PROCESS_DEF_VERSION = "VDPR"
```

VDPRDDDDXXXXXXXXXXYYYYYYYY0000000000000000

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

```
VDISDDDDXXXXXXXXXXXXXXXXXXXX0000000000000000
```

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

```
VDMODDDDDXXXXXXXXXXXXXXXXXXXX0000000000000000
```

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

```
VDVWDDDDXXXXXXXXXXXXXXXXXXXX0000000000000000
```

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

```
VDVEDDDDDXXXXXXXXXXXXXXXXXXXX0000000000000000
```

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

```
ISJBDDDDXXXXXXXXX000000000000000000000000
```

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

```
MOCRDDDDXXXXXXXXX000000000000000000000000
```

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

```
ASGIDDDDDXXXXXXXXYYYYYYYY0000000000000000
```

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

```
ASRIDDDDDXXXXXXXXYYYYYYYY0000000000000000
```

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

```
ASUIDDDDDXXXXXXXXYYYYYYYY0000000000000000
```

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

```
CELMDDDDXXXXXXXXXXXXXXXXXXXXZZZZZZZZ00000000
```

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

```
PROCDDDDXXXXXXXX000000000000000000000000
```

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

```
TASKDDDDXXXXXXXXXXXXXXXX0000000000000000
```

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

```
FLDRDDDDXXXXXXXX000000000000000000000000
```

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

```
FLVCDDDDXXXXXXXXXXXXXXXXZZZZZZZZWWWWWWW
```

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),

WWWWWWW 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

FLMYDDDDXXXXXXXXX000000000000000000000000

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

FLWSDDDDXXXXXXXXX000000000000000000000000

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

FLWRDDDDXXXXXXXXX000000000000000000000000

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

PINFDDDDXXXXXXXXXXXXYYYYYY0000000000000000

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_DOCF = "PINP"
```

Pin contained in process object prefix.

Id structure is

```
PINPDDDDXXXXXXXXXXXXXXXXXXXX0000000000000000
```

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

```
PINWDDDDXXXXXXXXXXXXXXXXXXXX0000000000000000
```

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

```
DOCFDDDDXXXXXXXXXXXXXXXXXXXXZZZZZZZZ00000000
```

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

```
DOCPDDDDXXXXXXXXXXXXXXXXXXXXZZZZZZZZ00000000
```

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

```
FCHFDDDDXXXXXXXXXXXXXXXXZZZZZZZZ00000000
```

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_DOCB = "FCHD"
```

File-less document (index fiche) contained in distribution list (modified copy of original) /.

Id structure is

```
FCHDDDDDDXXXXXXXXXXXXXXXXZZZZZZZZ00000000
```

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

```
FCHVDDDDXXXXXXXXXXXXXXXXZZZZZZZZ00000000
```

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

```
FCHWDDDDXXXXXXXXXXXXXXXXZZZZZZZZ00000000
```

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_DOCB = "DOCB"
```

File document contained in distribution list /.

Id structure is

```
DOCBDDDDXXXXXXXXXXXXXXXXZZZZZZZZ00000000
```

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

```
DOCVDDDDXXXXXXXXXXXXXXXXZZZZZZZZ00000000
```

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

```
DOCWDDDDXXXXXXXXXXXXXXXXZZZZZZZZ00000000
```

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

```
DOCVDDDDXXXXXXXXXXXXXXXXZZZZZZZZ00000000
```

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_DOCP = "RFDP"
```

Folder contained reference to a file document object prefix.

Id structure is

```
RFDFDDDDXXXXXXXXXXXXXXXXZZZZZZZZ00000000
```

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

```
RFDPDDDDXXXXXXXXXXXXXXXXZZZZZZZZ00000000
```

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

```
RFFFDDDDXXXXXXXXXXXXXXXXZZZZZZZZ00000000
```

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

```
RFFPDDDDXXXXXXXXXXXXXXXXZZZZZZZZ00000000
```

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

```
NTPRDDDDXXXXXXXXX0000000000000000YYYYYYYYY
```

where DDDD is the database numeric id, XXXXXXXXX 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

```
NTDFDDDDXXXXXXXXXYYYYYYYYYZZZZZZZZWWWWWWW
```

where DDDD is the database numeric id, XXXXXXXXX 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

```
NTDFDDDDXXXXXXXXXYYYYYYYYYZZZZZZZZWWWWWWW
```

where DDDD is the database numeric id, XXXXXXXXX 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

```
NTDFDDDDXXXXXXXXXYYYYYYYYYZZZZZZZZWWWWWWW
```

where DDDD is the database numeric id, XXXXXXXXX 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

NTDFDDDDXXXXXXXXXXXXXXXXZZZZZZZZWWWWWWW

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

VNDFDDDDXXXXXXXXXXXXXXXXZZZZZZZZWWWWWWW

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

VNDFDDDDXXXXXXXXXXXXXXXXZZZZZZZZWWWWWWW

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

TKDFDDDDXXXXXXXXXXXXXXXX0000000000000000

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

USER0000XXXXXXXX000000000000000000000000

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

```
ARFLDDDDXXXXXXXXX000000000000000000000000
```

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

```
APINDDDDDXXXXXXXXXXXXYYYYYYYYY0000000000000000
```

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

```
ADOCDDDDXXXXXXXXXXXXYYYYYYYYYZZZZZZZZ00000000
```

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

```
INRE0000XXXXXXXXX000000000000000000000000
```

where XXXXXXXX is the hexadecimal representation of the intervention report numeric id.

```
public static final String SERVER_REPORT = "SVRE"
```

Server report.

Id structure is

```
SVRE0000XXXXXXXXX000000000000000000000000
```

where XXXXXXXX is the hexadecimal representation of the server report numeric id.

```
public static final String FOLDER_COMMENT = "CMTF"
```

A folder comment.

```
CMTFDDDDXXXXXXXXX0000000000000000CCCCCCCC
```

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.

```
ACMFDDDDXXXXXXXXX0000000000000000CCCCCCCC
```

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.

```
CMDFDDDDXXXXXXXXXYYYYYYYZZZZZZZZCCCCCCCC
```

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.

```
ACDFDDDDXXXXXXXXXYYYYYYYZZZZZZZZCCCCCCCC
```

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 = "CMTF"
```

A processes comment.

```
CMTFDDDDXXXXXXXXX0000000000000000CCCCCCCC
```

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"
```



```
public static final String VALIDCYCLE_STAGE_DATA_TEMPLATE = "VCDT"
```

validation cycle template stage data.

Id structure is

```
VCDTDDDDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

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

```
VCDRDDDDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

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

```
VCNTDDDDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

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

```
VCNRDDDDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

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

```
EXDLDDDDXXXXXXXXX000000000000000000000000
```

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

```
NODFDDDDXXXXXXXXX000000000000000000000000
```

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

```
NOTFDDDDXXXXXXXXX000000000000000000000000
```

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

```
FAVODDDDDXXXXXXXXX000000000000000000000000
```

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

```
OLNKDDDDXXXXXXXXX000000000000000000000000
```

where DDDD is the database numeric id, XXXXXXXX is the hexadecimal representation of the link id.

```
public static final String PACKAGE = "PKG"
```

Package link

Id structure is

PKGDDDDXXXXXXXX000000000000000000000000

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

JUKEDDDDDXXXXXXXX000000000000000000000000

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 abstract 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 :

<pre>public Siatel_1_4 ()</pre>
<pre>public Siatel_1_4 (String id)</pre>
<pre>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)</pre>

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 unique 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 ()
```

EXAMPLES

BASIC G7API EXAMPLE

Content of file `config.php`

```
<?php
    $G7_url = 'example.domain.org';
    $G7_user = 'my-user';
    $G7_pass = 'my-password';
?>
```

Content of file `test.php`

```
<html>
    <head>
    </head>
    <body>
        <?php
            include_once(dirname(__FILE__) . '/config.php');
            include_once(dirname(__FILE__) . '/modules/gargantua.class.php');

            $G7API = G7API(
                $G7_url,
                $G7_user,
                $G7_pass);

            if ($G7API->hasConnection())
            {
                ?> <span>Connection successfull.</span> <?php
            }
            else
            {
                ?> <span>Connection failure </span> <?php
            }
        ?>
    </body>
</html>
```

```
</body>  
</html>
```



```

        'count' => 1,
        'offset' => 0,
        'expression' => '\attribute\' = \'value\'',
        'attributes' => array (
            'object.label', 'object.fileId')
    )
);
?>

<div class="container">
    <div class="row" id="results">
        <?php
            foreach ($search->{'results'} as $searchResult)
            {
                ?>

                <div class="col-md-2" style="height:350px">

                    <a href="download.php?id=?php echo $searchResult-
>{'object.id'}; ?>">

                        <div style="height:7em;"><h3><?php echo
$searchResult->{'object.label'}; ?></h3></div>

                                </a>

                            </div>

                        <?php
                            }
                        ?>

                    </div>

                </div>

            </div>

        <br/>
        <br/>

        <input id="more" type="button" value="More" >

        <br/>
        <br/>

        <script>

            $("#more").click(function() {

```

```

        $.ajax({
            url : "more.php",
            data: { context : '<?php echo $search->{'context'} ?>' },
        }) .done(function(response) {
            $(response).appendTo("#results");
        }) .fail(function() {
            alert( "error" );
        });
    });
</script>
</body>
</html>

```

Content of file [more.php](#)

```

<?php
    include_once(dirname(__FILE__).'/config.php');
    include_once(dirname(__FILE__).'/modules/gargantua.class.php');

    // attach to existing instance
    $G7 = G7API($G7_url);

    $search = $G7->doRequest(
        'search',
        'more',
        array(
            'context' => $_GET['context'],
            'count' => 1 )
    );

    foreach ($search->{'results'} as $searchResult)
    {
        ?>

        <div class="col-md-2" style="height:350px">

            <a href="download.php?id=<?php echo $searchResult-
            >{'object.id'}; ?>">

                <div style="height:7em"><h3><?php echo $searchResult-
                >{'object.label'}; ?></h3></div>

```

```

                </a>

            </div>

        <?php
    }

?>

```

Content of file [thumb.php](#)

```

<?php

    include_once(dirname(__FILE__).'/config.php');
    include_once(dirname(__FILE__).'/modules/gargantua.class.php');

    // attach to existing instance
    $G7 = G7API($G7_url);

    header('Content-Type: image/jpeg');
    header('Content-Disposition: inline; filename="thumb.jpg"');
    header('Cache-Control: private, max-age=0, must-revalidate');
    header('Pragma: public');

    ini_set('zlib.output_compression', '0');
    die($G7->getThumbnail($_GET['id'], 200, 300));

?>

```