
Workflows API

Release 15.4.1.3

CONTACT Software

Oct 29, 2018

Contents

| | | |
|----------|---|-----------|
| 1 | Workflow Object | 1 |
| 2 | Workflow Task | 4 |
| 3 | Workflow Form | 6 |
| 4 | Briefcase | 7 |
| 4.1 | Sort Order | 8 |
| 4.2 | Briefcase Link | 8 |
| 5 | Workflow Designer Catalog Conditions | 9 |
| | Index | 12 |

Workflow Object

The APIs to manage a workflow object are defined in the standard object class `cs.workflow.processes.Process` (page 1) and an interface class `cs.workflow.designer.wfinterface.Process`, which enable some advanced APIs on the standard object class to handle workflow programmatically.

```
class cs.workflow.processes.Process (**values)
```

AddAttachment (*object_id*)

Add an object to the attachment briefcase of this process.

RemoveAttachment (*object_id*)

Remove an object from the attachment briefcase of this process.

AddConstraint (*rule_name*, *invert_rule=0*, *task_id=None*, *briefcase_id=None*)

Add a constraint to the process.

Parameters

- **rule_name** – the name of a cdb object rule.
- **invert_rule** (*either 0 or 1.*) – if negative, the object rule is inverted.
- **task_id** – the id of a task in the process (optional).
- **briefcase_id** – the id of a briefcase in the process (optional); if a *task_id* is specified, the briefcase should be assigned to the given task or be a global briefcase.

CreateTask (*ttype*, *title*, *relative_position=None*, *other_task=None*, ***kwargs*)

Add a new task to the workflow.

If the workflow has no task group, a parallel task group is created. Otherwise it is possible to add a task, specifying the relative position (before, after, parallel) of the new task with regard to another task (*other_task*)

The workflow should have no tasks and at most one task group. Otherwise an exception is raised.

Parameters

- **ttype** (*one of ['execution', 'approval', 'examination', 'system']*) – type of the new task
- **title** (*string*) – title of the new task

The following parameters are optional

Parameters

- **subject_id** (*string*) – ID of the responsible for the new task
- **subject_type** (*one of ['Person', 'Common Role', 'PCS Role']*) – type of the responsible for the new task
- **deadline** (*string formatted as in "14.09.1983"*) – deadline for the new task
- **description** (*string*) – description of the new task
- **finish_option** (*integer*) – flag to close the process early in case of success
- **task_definition_id** (*a cdb_object_id*) – cdb_object_id of the system task definition for the new tag, only for system tasks
- **parameters** (*dictionary*) – parameters of the system task

Raises cdb.ue.Exception

CreateBriefcase (*name*)

Interface method to create a briefcase. Checks permissions and executes user exits.

SaveAsTemplate (*self*)

Save the current workflow as template. A new database object is created.

Returns the new template as a cs.workflow.processes.Process object.

ModifyProcess (***kwargs*)

Interface method to modify the current process. Checks permissions and executes user exits.

classmethod CreateProcess (***kwargs*)

Interface method to create a process. Checks permissions and executes user exits. All the parameters are optional

Parameters

- **title** (*string*) – The title of the workflow. If not given a standard title will be used. This is defined in the CDB-label cdbwf_standard_workflow_title.
- **subject_id** (*string*) – The subject_id of the workflow responsible. If not given, the id of the current user is used.
- **subject_type** (*string*) – The subject_type of the workflow responsible. If not given 'Person' is used.
- **categ_id** (*string*) – The id of the workflow category
- **cdb_project_id** (*string*) – The id of the project associated to the workflow.
- **start_date** (*string*) – Start date of the workflow.
- **deadline** (*string*) – A deadline for the workflow.
- **max_duration** (*integer*) – Maximum duration of the workflow in days.
- **org_id** (*string*) – Id of an organization, to which the workflow belongs.

Returns The new process

SimplifyStructure ()

Simplify the task structure under the current component

activate_process ()

Start the current workflow. The tasks would be set to ready if possible.

cancel_process (*comment=""*)

Cancel the current workflows. The completion tasks would be activated if exist.

close_process()

Attempts to close the workflow. The completion tasks would be activated if exist.

dismiss_process()

Dismiss the current workflow. The completion tasks would be cancelled.

isActivatable()

Manually check whether a process can be started(activated) or not.

isDismissable()

Manually check whether a process can be discarded or not.

Workflow Task

The APIs to manage a workflow object are defined in the standard object class `cs.workflow.tasks.Task` (page 4) and an interface class `cs.workflow.designer.wfinterface.Task`, which enable some advanced APIs on the standard object class to handle workflow task programmatically. The task type specific APIs are defined in different derived classes.

class `cs.workflow.tasks.Task` (***values*)

Base class for both interactive task and system task types. Defines some common APIs to handle workflow task.

ModifyTask (***kwargs*)

Interface method to modify the current task. Checks permissions and executes user exits.

If the process is already running, the current task must not be running or closed. Otherwise an exception is raised.

Raises `cdb.ue.Exception`

DeleteTask ()

Delete the task from the database and from the workflow. If needed rearrange the task groups.

Checks permissions and executes user exits

If the process is already running, the current task must not be running or closed. Otherwise an exception is raised.

Raises `cdb.ue.Exception`

Returns the most derived unchanged component

CloseTask (***kwargs*)

Interface method to close the current task. Checks permissions and executes user exits.

ForwardTask (***kwargs*)

Interface method to forward the current task. Checks permissions and executes user exits.

RefuseTask (***kwargs*)

Interface method to refuse the current task. Checks permissions and executes user exits.

activate_task ()

Attempts to set the current task ready. If the constraints check fails, the task would be cancelled.

cancel_task (*comment=""*)

Attempts to cancel the current task.

requiresComment ()

Manually check whether the comment is required while closing or canceling a task.

class `cs.workflow.tasks.InteractiveTask` (***values*)

getExtensionObject ()

Get the task extension object instance, if the current task is extended.

classmethod **getPossibleExtensions** ()

Get the possible extension types for the current task type.

Returns a list of dictionary with keys `label` and `classname` of each available extension type.

classmethod **get_running_tasks_by_briefcase** (*briefcase*)

Find out which running tasks are linked with a given briefcase.

Returns a list of tasks

The APIs to manage a workflow form object are defined in the standard object class `cs.workflow.forms.Form` (page 6).

```
class cs.workflow.forms.Form(**values)
```

```
    classmethod InitializeForm(task, form_template)
```

1. Initializes a form using FormTemplate object `form_template`
2. Creates a new local briefcase for given Task object `task` named like the form template
3. Adds the form to the briefcase's contents
4. Returns the new form

```
    get_empty_mandatory_fields()
```

Returns a set of all mandatory field names containing NULL values.

```
    preset_data()
```

Presets form data using all non-form objects in the same briefcase. Already present data is not overwritten.

Caller is responsible for checking access, especially if the briefcase is an edit briefcase.

```
    read_data()
```

Returns persisted form data as dict.

```
    write_data(json_data)
```

Persist given `json_data`, which must be a JSON-serializable dict.

Raises a `ValueError` if `json_data` is not a dict. Raises a `TypeError` if `json_data` is not JSON-serializable.

Briefcase

The APIs to manage a briefcase are defined in the standard object class `cs.workflow.briefcases.Briefcase` (page 7) and an interface class `cs.workflow.designer.wfinterface.Briefcase`, which enable some advanced APIs on the standard object class to handle briefcase programmatically.

class `cs.workflow.briefcases.Briefcase` (***values*)

ModifyBriefcase (***kwargs*)

Interface method to modify the current briefcase. Checks permissions and executes user exits.

DeleteBriefcase ()

Interface method to delete a briefcase. Checks permissions and executes user exits.

SetGlobalMeaning (*meaning, extends_rights=None*)

Set the global meaning of the briefcase for the current process. Overwrites an already existing global meaning.

There should be no task specific meaning for any task in the process. Otherwise an exception is raised.

Raises `cdb.ue.Exception`

SetTaskMeaning (*meaning, task, extends_rights*)

Set the meaning of the briefcase for a specific task. Overwrites an already existing meaning for the given task.

There should be no global meaning for the given briefcase in the process. Otherwise an exception is raised.

Raises `cdb.ue.Exception`

RemoveTaskMeaning (*task*)

Interface method to remove meaning of the briefcase for a specific task.. Checks permissions and executes user exits.

AddObject (*object_id*)

Add an object to the briefcase

Parameters `object_id` (*a cdb_object_id*) – the `cdb_object_id` of an existing object

RemoveObject (*object_id*)

Remove an object from the briefcase

Parameters `object_id` (*a cdb_object_id*) – the `cdb_object_id` of an existing object

classmethod ByContent (*obj_or_obj_id*)

Return all briefcases which contain given object or objects with given cdb_object_id

4.1 Sort Order

When displayed in the Workflow Designer or *CONTACT Tasks*, briefcases are usually sorted by their name. You can customize their sort order by overwriting the default `key` function for the `sorted` call in PowerScript:

Example Sort briefcases by ID

```
from cs.workflow.briefcases import WithBriefcase
WithBriefcase.__briefcase_sorting_key__ = lambda self, x: x.briefcase_id
```

4.2 Briefcase Link

The meaning of a briefcase for the workflow or a task, is carrying on a `cs.workflow.briefcases.BriefcaseLink` (page 8) object.

class `cs.workflow.briefcases.BriefcaseLink` (**values)

DeleteBriefcaseLink ()

Interface method to delete a briefcase link. Checks permissions and executes user exits.

Workflow Designer Catalog Conditions

class `cs.workflow.designer.catalogs.CatalogWithCustomizableCondition`

Catalogs of this class allow for customizing the default search conditions which are not accessible to the user.

Provide custom conditions by connecting to the `cdb.sig.signal` `cs.workflow.designer.catalogs.CATALOG_CONDITION`. Your functions will receive a single parameter containing the fully-qualified Python name of the calling catalog instance and is expected to return a list or set.

If you provide any custom conditions for a catalog, the default conditions will be ignored. Include the default conditions in your custom conditions if you want to keep them.

Usage examples:

```
from cdb import sig
from cs.workflow.designer.catalogs import CATALOG_CONDITION

@sig.connect(CATALOG_CONDITION)
def custom_catalog_condition(fqpyname):
    if fqpyname == "cs.workflow.designer.catalogs.ConstraintsCatalog":
        return ["cdb_module_id='my.constrs'", "name LIKE 'wf-designer:%'"]

    if fqpyname == "cs.workflow.designer.catalogs.FormTemplateCatalog":
        return ["status = 20"]

    if fqpyname == "cs.workflow.designer.catalogs.OperationCatalog":
        return ["inactive=0 OR inactive IS NULL"]

@sig.connect(CATALOG_CONDITION)
def another_condition(fqpyname):
    if fqpyname == "cs.workflow.designer.catalogs.FormTemplateCatalog":
        return ["mask_name != 'forbidden'"]
```

List of Figures

List of Tables

A

activate_process() (cs.workflow.processes.Process method), 2
 activate_task() (cs.workflow.tasks.Task method), 4
 AddAttachment() (cs.workflow.processes.Process method), 1
 AddConstraint() (cs.workflow.processes.Process method), 1
 AddObject() (cs.workflow.briefcases.Briefcase method), 7

B

Briefcase (class in cs.workflow.briefcases), 7
 BriefcaseLink (class in cs.workflow.briefcases), 8
 ByContent() (cs.workflow.briefcases.Briefcase class method), 7

C

cancel_process() (cs.workflow.processes.Process method), 2
 cancel_task() (cs.workflow.tasks.Task method), 4
 CatalogWithCustomizableCondition (class in cs.workflow.designer.catalogs), 9
 close_process() (cs.workflow.processes.Process method), 2
 CloseTask() (cs.workflow.tasks.Task method), 4
 CreateBriefcase() (cs.workflow.processes.Process method), 2
 CreateProcess() (cs.workflow.processes.Process class method), 2
 CreateTask() (cs.workflow.processes.Process method), 1

D

DeleteBriefcase() (cs.workflow.briefcases.Briefcase method), 7
 DeleteBriefcaseLink() (cs.workflow.briefcases.BriefcaseLink method), 8
 DeleteTask() (cs.workflow.tasks.Task method), 4
 dismiss_process() (cs.workflow.processes.Process method), 3

F

Form (class in cs.workflow.forms), 6

ForwardTask() (cs.workflow.tasks.Task method), 4

G

get_empty_mandatory_fields() (cs.workflow.forms.Form method), 6
 get_running_tasks_by_briefcase() (cs.workflow.tasks.InteractiveTask class method), 5
 getExtensionObject() (cs.workflow.tasks.InteractiveTask method), 5
 getPossibleExtensions() (cs.workflow.tasks.InteractiveTask class method), 5

I

InitializeForm() (cs.workflow.forms.Form class method), 6
 InteractiveTask (class in cs.workflow.tasks), 5
 isActivatable() (cs.workflow.processes.Process method), 3
 isDismissable() (cs.workflow.processes.Process method), 3

M

ModifyBriefcase() (cs.workflow.briefcases.Briefcase method), 7
 ModifyProcess() (cs.workflow.processes.Process method), 2
 ModifyTask() (cs.workflow.tasks.Task method), 4

P

preset_data() (cs.workflow.forms.Form method), 6
 Process (class in cs.workflow.processes), 1

R

read_data() (cs.workflow.forms.Form method), 6
 RefuseTask() (cs.workflow.tasks.Task method), 4
 RemoveAttachment() (cs.workflow.processes.Process method), 1
 RemoveObject() (cs.workflow.briefcases.Briefcase method), 7
 RemoveTaskMeaning() (cs.workflow.briefcases.Briefcase method), 7

[requiresComment\(\)](#) (cs.workflow.tasks.Task method), [4](#)

S

[SaveAsTemplate\(\)](#) (cs.workflow.processes.Process method), [2](#)

[SetGlobalMeaning\(\)](#) (cs.workflow.briefcases.Briefcase method), [7](#)

[SetTaskMeaning\(\)](#) (cs.workflow.briefcases.Briefcase method), [7](#)

[SimplifyStructure\(\)](#) (cs.workflow.processes.Process method), [2](#)

T

[Task](#) (class in cs.workflow.tasks), [4](#)

W

[write_data\(\)](#) (cs.workflow.forms.Form method), [6](#)