Basics

Release 15.4.2.4

CONTACT Software

Aug 15, 2018

Contents

1 Introduction 1

2 Calendar 2
2.1 Introduction . . . 2
2.2 Calendar profiles . . . 2
2.3 Calendar profile exceptions . . . 3
2.4 Calendar entries . . . 3
2.5 Day type . . . 4

3 Currencies and exchange rates 5

4 Semantic links 6
4.1 Introduction . . . 6
4.2 Link Types . . . 6
4.3 Link Graph . . . 7

5 Audit Trail 9
5.1 Introduction . . . 9
5.2 Configuration . . . 9

CHAPTER 1

Introduction

CONTACT Basics provides carefully selected cross-application functions such as calendars, currencies, batch operations, semantic links and the Audit Trail.

CHAPTER 2

Calendar

2.1 Introduction

A certain *calendar profile* is assigned to each employee by the administrator (see *Calendar profiles* (page 2)). Through the combination of the *Calendar profile*, *Calendar profile exceptions*, *Calendar entries* and *Day type* operations, the administrator has suitable tools for mapping the working time models as well as presences and absences of the individual employees. Based on a week rule, each *calendar profile* specifies which days of the week will be worked on, and which will be free. The *calendar profile exceptions* can be used to define exceptions to the *calendar profile* (e.g. official holidays, company holidays, etc.).

With the assignment of a *calendar profile*, each employee gets a personal *calendar* in which he can record his own *calendar entries*, e.g. to indicate that he will be absent for a certain period for a certain reason (holiday, illness, weekend, official holiday, short-time work, etc.) or will make less capacity available.

2.2 Calendar profiles

The administrator defines a generally valid calendar, in which the normal working time is defined for the entire company, as *Calendar profile*. The days of the week when work will be done or not done are specified here.

If a company has different working time models, a separate calendar can be created for each location. This way regionally different working day structures can be mapped for the company (examples: Location 1 has Monday through Friday as workdays, location 2 has Monday through Saturday as workdays, etc.).

Likewise, multiple calendars can be defined for one location, each for an exactly specified period. However, the periods must not overlap.

Then the calendar applicable to each employee is assigned to him or her. If the *Resource* option for the employee is enabled (for details on this, refer to *CONTACT Elements Client Reference*), the assigned calendar is applied as the basis for determining the resource capacity.

2.2.1 Creating a calendar profile

In the navigation area, the administrator goes to the entry *Administration/Configuration* → *Administration* → *Calendar* → *Calendar profiles*. By right-clicking *Calendar profiles* he or she can call up the *New…* operation in the context menu.

The administrator defines a new calendar in the *Calendar profile (New)* dialog.

Name Here, the administrator has to specify a name for the calendar.

Valid from Here, the administrator has to specify the beginning of the validity range for the calendar. The entered day is the first day of validity.

Valid until Here, the administrator has to specify the end of the validity range for the calendar. The entered day is the last day of validity.

Monday* to *Sunday The administrator has to specify the *Day type* (page 4) for each individual day of the week.

Description Here, the administrator can describe particularities for the *calendar profile*.

Click the `New` button to create the *calendar profile*.

2.3 Calendar profile exceptions

Calendar profile exceptions define exceptions to a calendar profile, which then apply for all persons with this calendar profile. A possible *calendar profile exception* could be, for example, an all-day company outing or regional holidays. The *calendar profile exception* has to be linked with exactly one *calendar profile* (see *Calendar profiles* (page 2)). Skilled combination of calendar profiles and calendar profile exceptions enables coupling of regional holidays with company-specific calendar profiles.

2.3.1 Importing calendar profile exceptions from an .ics file

Please search under *Administration/Configuration* → *Administration* → *Calendar* → *Calendar Profiles*, at first for the calendar profile for which you would like to import multiple exceptions (particularly official holidays) all at once from an .ics file. Select the calendar profile in the result list and click the *Import holidays from .ics file* operation in the context menu of the selected calendar profile. In the subsequent dialog, select the path for the .ics file to be imported and confirm the dialog. Then the exceptions are automatically created for the selected calendar profile.

2.3.2 Manually creating a calendar profile exception

Please search under *Administration/Configuration* → *Administration* → *Calendar* → *Calendar profiles*, at first for the calendar profile for which you would like to define an exception. Then open the data sheet of the calendar profile and call up the *New...* operation in the *Calendar profile exceptions* tab.

Define a new calendar profile exception in the *Calendar exception (New)* dialog.

Calendar profile Here, the administrator has to select a *calendar profile* (page 2), for which the *calendar profile exception* applies. It is selected from a list of options of all defined *calendar profiles*.

Day Here, the administrator has to select a day for which the *calendar profile exception* applies.

Day type Here, the administrator has to specify a *Day type* (page 4). It is selected from a list of options of all defined day types.

Description Here, the administrator can describe particularities for the *calendar profile exception*.

Click the `New` button to finish creating the new *calendar profile exception*.

2.4 Calendar entries

With the *Calendar entries* operation the employees of the company can manage their presences and absences in a more targeted manner. This operation primarily manages the annual holidays of the employees. Furthermore, the *Calendar entries* operation can be used to show, for example, if an employee will be going on parental leave, or has expressed willingness to work overtime for a certain period. For each of these cases, he or she can use a *calendar entry* to adjust his or her *calendar profile* according to personal considerations.

2.4.1 Create a calendar entry

In the navigation area, go to the entry *Organizations* → *Persons*. By right-clicking *Persons* you can call up the *Search...* operation in the context menu.

With the *Person (Search)* dialog you can search for your personal data. Double-click the entry with your username to open the data sheet with your personal data. In the personal data sheet, enable the *Calendar entries* tab (for details on this, refer to the *CONTACT Elements Client Reference*). Right-click the *Calendar entries* field range to get to the *New...* operation in the context menu.

Define a new calendar entry in the *Calendar entry (New)* dialog.

From Specify the beginning of the validity range for the *calendar entry* here.

Until Specify the end of the validity range for the *calendar entry* here.

Person The system enters the name of the logged-in user here. The field cannot be edited if the dialog was opened as a tab in the personal data sheet. The field can be edited if the *Calendar entry (New)* dialog was called up via the navigation area *Administration/Configuration* → *Administration* → *Calendar* → *Calendar entries*.

Day type Specify a *Day type* (page 4) here. It is selected from a list of options of all defined day types.

Capacity Here, the system automatically enters the value that is specified according to the selection in the *Day type* (page 4) field. If an expansion of capacity (for example, overtime) is to be defined, the *Workday* value is selected in the *Day type* field and the value for the total capacity is entered here, thus the normal working time plus the anticipated overtime.

Description Here, the user can describe particularities for the *calendar entry*.

Click the `New` button to create the *calendar entry*.

2.5 Day type

The *day type* specifies the type of presence or absence concerned for the *calendar entry*, *calendar profile exceptions*, or individual weekdays of the *calendar profile*.

2.5.1 Creating a day type

In the navigation area, the administrator goes to the entry *Administration/Configuration* → *Administration* → *Calendar* → *Day type*. By right-clicking *Day type* he or she can call up the *New...* operation in the pop-up menu.

In the *Day type (New)* dialog the administrator defines a new *Day type*.

Day type ID Here, the administrator has to specify a numerical value for an ID. The selection of the numerical value is arbitrary. However, each numerical value may occur only once.

Is day off With this option field the administrator specifies whether the *day type* concerns a presence or an absence.

Name (de) Here, the administrator has to define a German name for the *day type*.

Name (en) Here, the administrator can define an English name for the *day type*.

Click the `New` button to create the *day type*.

CHAPTER 3

Currencies and exchange rates

You can manage currencies and their exchange rates via the *Administration/Configuration –-> Catalog Administration → General -> Currencies* menu item in the navigation tree. One of the defined currencies can be specified here as the reference currency.

The exchange rates between all defined currencies can be calculated by using the reference currency, if the exchange rate with respect to the reference currency was defined for each currency. Exchange rates can also be established for a specific project.

CHAPTER 4

Semantic links

4.1 Introduction

Semantic Links are a universal tool for the networking of business objects. In the following chapters, you will learn how to configure this tool to meet your individual needs.

4.2 Link Types

In a semantic link between two business objects, one object basically represents the *source* and the other the *target*, the *semantic* then expresses the meaning of the link between source and target, e.g.:

> **Requirement X** (subject) **specifies** (verb) **Part Y** (object)

The configuration, which business objects can be linked to each other with which semantics, is done via so-called *Link Types*. You can maintain these via the menu access *Administration/Configuration* → *Configuration* → *Semantic Links* → *Semantic Link Types*. You can define a link type by using the following master data:

- ***Source*** Select the class that should represent the source in the link.
- ***Semantic*** Here you can freely enter the semantics of the link between source and target (usually in the form of a verb).
- ***Target*** Select the class that should represent the target in the link.
- ***Create Mirror Link*** Using this field, you can define that with the creation of this type, a mirror link selected via the field :guilabel:‘Mirror Link Type’ should be generated automatically, in which source and target are exchanged and the semantic (passive or active) verb form is changed accordingly. In addition, deleting a link automatically deletes the mirror link, if applicable.
- ***Mirror Link Type*** Select an already defined link type that represents the *Mirror Link* of the currently edited type, e.g.:

  Requirement X specificies Part Y **mirrors** Part Y is specified by Requirement X
- ***Copy Link Type*** Check this box if the link type defines the semantics *is copy of*. Then the system automatically creates a semantic link of this type when you copy an object.
- ***Invalid*** Set a Link type to *Invalid* if you no longer want to be able to create a new link.

- ***Module*** Here, the system automatically enters the module in which the type is defined when creating a new module (this is your customer module in the case of customer-specific configuration).

Note: If you define a new link type in which a class is used for the first time as a *source* or *destination*, you must also have to define a relationshio *Semantic Links*, in which the class is entered as referer and the standard class: guilabel:`cdb_semantic_link` as link class and reference (see as an example the definition of the standard link: guilabel:`cdbrqm_req2slink`). Only when this link is defined, the semantic links of the relevant business objects become accessible to the user (via the tab of the same name in the data sheet of the technical objects). Use this tab to create, change and delete semantic shortcuts.

4.3 Link Graph

4.3.1 Introduction

The: guilabel:`Link Graph` is a universal Elink application for visualizing and navigating semantic networks between any technical objects, e.g. between requirements, articles and project tasks in requirements management.

Below you will find out how to configure customer-specific which business objects a graph should contain and what you have to do to call the *Link Graph* with a defined configuration from the context of a linked business object.

4.3.2 Configuration

The configuration of which business objects a certain graph should contain and how it should be displayed is done first of all via so-called *Link Graph Configurations*. You can maintain these via the menu access *Administration/Configuration* → *Configuration* → *Semantic Links* → *Link Graph Configuration*. The definition of a *Link Graph Configuration* is done via the following master data:

- ***Name*** Give the configuration an understandable name.
- ***Graphviz call-up*** GraphViz is used for visualizing graphs (see http://www.graphviz.org/). You can enter the transfer parameters for the GraphViz call-up here. These allow you to configure the representation of the graph.
- ***Font Name*** Here, you can define which font is to be used for the edge and node labels.
- ***Font Size*** Here, you can define the font size in which the edge and node labels should be represented.
- ***View*** Here you can define a view that is taken into account when creating the Link Graph. Therein, other *Semantic Links* between objects can be created via SQL, which are not listed in the *Link Types*.
- ***Module*** Here, the system automatically enters the module in which the type is defined (this is your customer module).

4.3.3 Configuration Call

In order to open the *Link Graph* for the objects of a certain class, you first have to configure an operation (e.g. the operation *link_graph*) and implement the *Link Graph* call in Python via this operation, e.g. as follows for the class *Requirement* and the configuration *RQMSemanticLinkGraph*:

```
#!/usr/bin/env powerscript
# -*- python -*- coding: UTF-8 -*-
from cs.tools.semanticlinks import LinkGraphConfig

def on_link_graph_now(self, ctx):
    config = LinkGraphConfig.KeywordQuery(name="RQMSemanticLinkGraph")[0].cdb_
↪object_id
    ctx.url("/cs-tools-semanticlinks-linkgraph/?config_id=%s&root=%s&radius=1" %␣
↪(config,self.cdb_object_id))
```

CHAPTER 5

Audit Trail

5.1 Introduction

Audit Trail is a component for capturing and comparing changes. By default, only changes that are made during creation, modification or deletion are recorded. Further types of changes can be configured and added in the course of PowerScript customizing. In the following chapters you will learn how you can configure this tool to suit your individual requirements.

5.2 Configuration

5.2.1 Introduction

The *Audit Trail Configuration* consists of several parts:

- ***Audit Trail Configuration*** This determines which subject objects are to be monitored by the *Audit Trail*.
- ***Audit Trail Types*** Here the different types and icons for the *Audit Trail* are determined.
- ***PowerScript Customizations*** For business objects to receive an *Audit Trail*, classes in PowerScript must inherit from the *WithAuditTrail* class *cs.audittrail*.
- ***Relationship*** To use the :guilabel‘:`Audit Trail`, a relationship must be created.
- ***Mask*** Here the necessary mask adjustments for the display of the *Audit Trail* are explained.

Note that the class to be extended contains a *cdb_object_id*.

5.2.2 Audit Trail Configuration

The configuration of which subject objects are to be monitored by the *Audit Trail* is done via the *Audit Trail Configuration*. You can maintain this via the menu access *Administration/Configuration* → *Configuration* → *Audit Trail* → *Audit Trail Configuration*. The definition of an *Audit Trail configuration* is done via the following master data:

- ***Name*** Give the configuration an understandable name.
- ***Class*** Indicates for which class the *Audit Trail* records modifcations, creations, and so on.

- ***Indexable?*** Specifies whether business objects in the class can be indexed. This is not a must, but it affects the data that is drawn to the *Audit Trail* when the information is output. During activation, not only the individual business object is considered, but all versions. This enables changes to be displayed and compared across versions.
- ***Ident Field*** If business objects of the class can be indexed, an ident field must be specified here. For documents this would be the z_number.
- ***Index / Version / Revision*** If business objects of the class can be indexed, the index field must be specified here. For documents this would be e.g. the z_index.
- ***Module*** Here, the system automatically enters the module in which the type is defined (this is your customer module).

After creating the *Audit Trail Configuration*, the monitored fields have to be specified (i.e. if a change is made to these, an Audit Trail entry will be made):

- ***Name*** Specify the attribute that is to be recorded in the audit trail.
- ***Module*** Here, the system automatically enters the module in which the type is defined (this is your customer module).

Note: Without these field *configurations*, no value changes are recorded.

5.2.3 Audit Trail Types

If additional types are required in connection with customer customizing, they can be entered for output in the *Audit Trail* via the menu access *Administration/Configuration* → *Configuration* → *Audit Trail* → *Audit Trail Types*. The definition of an *Audit Trail Type* is carried out using the following master data:

- ***Type*** Use this to name the type. This is required to identify the *Audit Trail* entry.
- ***Name*** Give this type an understandable name.
- ***Icon ID*** Select the *Audit Trail* icon to be displayed.

5.2.4 *PowerScript Customizations*

To capture the *Audit Trail* for the business objects of a specific class, you must inherit for your class from the *cs.audittrail* class and implement this in Python, e.g. as follows for the custom class *cDocument*:

```
#!/usr/bin/env powerscript
# -*- python -*- coding: UTF-8 -*-
from cs.audittrail import WithAuditTrail
from cs.documents import Document

class cDocument(Document, WithAuditTrail)
    ...
```

Furthermore, overwriting the *referencedAuditTrailObjects* method allows you to extend the display from your own business object to underlying business objects or to aggregate with other business objects. For example, you can display the changes to project tasks in the *Audit Trail* of the project. Of course only under the condition that both classes have been extended by *WithAuditTrail* in advance.

5.2.5 Relationship

To display the *Audit Trail* of business objects, a relation *Audit Trail* is required in which the class is recorded as Referer, the standard class *cdb_audittrail_objects* as Link Class and the standard class *cdb_audittrail_view* as Reference. In the Referer Keymap *cdb_object_id=object_id* and in the Reference Keymap *audittrail_object_id=audittrail_id* must be entered. Since the display is via the mask, it is recommended to hide the relationship.

5.2.6 Mask

To display on business objects, an additional mask is required for each business object, which must be entered in the mask register. The remaining name and the role name of the pre-generated relationship must be entered in the URL of the eLINKControl. The mask *cs_audittrail_view_reg* can be used as an example. The outlet *cs-audittrail-web-AuditTrailOutlet* is available to display the *Audit Trail* in the web client, where the relationship must also be entered.

List of Figures

List of Tables