Documents (cs.documents)

Release 15.2.3.7

CONTACT Software

Oct 29, 2018

Contents

CHAPTER 1

Release Notes for 15.2.3

The package is designed to work with *CONTACT Elements Platform* 15.3.

1.1 Checklist

Before you activate this version of `cs.documents` you have to check if the document operations work with the *Elements UI* (page 1).

You should also ensure that the new definition of the predicate `cs.documents:  Drawing Documents` is correct (see *Document Categories* (page 2)).

1.2 Elements UI

1.2.1 Operations

Create from template is now available with the Elements UI. A lot of operations has been activated for use with the Elements UI. You have to check if theses operations work with your installation. There are some customizing restrictions that are pointed out in the Elements UI documentation. You should at least check this topics:

- The standard customizing uses the mask `dokument_mask_webui` for the standard operations. You have to check if this mask is suitable for your installation.
- No `post_mask` and `dialogitem_change` user exits are called with the Elements UI. The standard solution has been adapted to use dialog hooks or to make checks at the `pre` Call. You have to check the calls in your customizing.
- Elements-UI does not support follow-Up operations.
- *File Properties* are not supplied.

If there is no time to adapt your customizing you have to deactivate the *Available with Web UI* Checkbox in the operation configuration.

1.2.2 Last Modified Documents - Widget

This widget replaces the *My Documents* widget. The documents are now sorted by the modification date and the UI has changed to display the information more compact.

1.3 Access System

The *Access Control Domain* `cs.documents: Documents - Unprotected` has been added which grants access to the non derived files of a document to the `public` role. The standard defines every document of the class `document` as unprotected. There is no change for documents of derived classes like `model`.

1.4 Document Categories

The document categories have been revised. There are a lot of new categories. Categories are not updated automatically so the change will only take place in new installations.

The attribute `parent_name` has been added to the data dictionary class `cdb_doc_categ` to display the name of the parent category. This also allows to fill the parent_name by regular catalog configuration - because of this the class `cs.documents.web.category_browser.CategoryBrowserCatalog` is no longer necessary and has been removed.

The predicate `cs.documents: Drawing Documents` has been changed to match the new document categories. In contrast to the predicate that is updated automatically unless it is not used in an *Access Control Domain* the document categories are not updated automatically. This can cause the problem that the register *Drawing* will not be displayed in some dialogs. You have to decide if you want to use the new category *Drawings / Mechanics* with its subcategories or you have to redefine the predicate to match your category system. The standad version of the predicate has used the term `z_categ2=177` in older versions.

1.4.1 Decomposition

The decomposition `DOC` is used to provide access with a predefined document category from the navigation tree. It is now possible to define the attribute that is used to sort the entries. The attribute is defined in the default setting with the id `decomp_sort_attribute` and the section `DOC`.

1.5 Template information

The attribute `source_oid` has been added to remember the template that has been used to create a document. It is filled automatically by the framework if a document is created using the *Copy* or the *Create From Template* operation.

1.6 Data Dictionary

The attribute `z_bereich` is now a predefined attribute (`cdb_person.abt_nummer`) because it is filled by the result of `auth.get_department`.

The attribute `source_oid` has been added (see *Template information* (page 2)).

1.7 Change History

The configuration of the table that displays the change history (cdbecm_doc_ecn) has been changed to sort the entries using the lenght of the index and the index itself. If this does not match your indexing scheme you should adapt the configuration.

1.8 Sharing

The standard recipient group *Reviewer* has been added to share a document with the reviewer.

1.9 Breaking a lock

If a user calls *Unlock* on a file locked by a different user the system now automatically generates a sharing to inform the person that has initiated the lock.

1.10 PowerScript

1.10.1 Presetting attributes for a document copy or version

The functions GetInitialCopyValues and GetInitialIndexValues have been added to cs.documents.Document to define the attributes that has to be resetted automatically when creating a new document version or a copy of a document.

The function GetDefaultErzSystem is used to preset the attribute erzeug_system if no integration is active. The value is read from the personal setting cs.documents.default_cad. This mechanism replaces the mask default $default_cad.

1.10.2 Presetting attributes for a document created from a template

The functions get_template_preset_attributes has been moved from the standard solution to the base class. The function is called automatically to determine the attributes that have to be copied from a template if you use the *Create from template* operation. The standard implementation returns the document category attributes.

1.10.3 Exclude states from interactive selection

The function IsValidInteractiveTargetStatus has been added to cs.documents.Document to exclude status from interactive use.

CHAPTER 2

Release Notes for 15.2.2

The package is designed to work with *CONTACT Elements Platform* 15.2.

2.1 Elements UI

The application has been adapted to the new design of Elements UI. The standard configuration allows reading access to all documents.

2.2 PowerScript

2.2.1 Object Life Cycle Calculation

The functions `CalculateOLC`, `GetOLCRelevantAttributes` and `GetReadonlyAttrsIfOLCStarted` has been added with Service Level 7 to `cs.documents.Document`. You might overwrite this functions if you use a copy of the standard solution.

2.2.2 Presetting attributes for a new document

The function `GetInitialCreateValues` has been added with Service Level 7 to `cs.documents.Document` to provide attributes that has to be set automatically when creating a new document.

2.3 Access system

The access right `save` has been added to the Access Conrol Domain `cs.documents: Documents (RELEASED)` in Service Level 7 with the *exclusive* flag set to prevent that a released document can be changed by anyone even if the access is granted elsewhere. If your access system is updated automatically (see `cdbpkg_sync.conf`) you have to ensure that this change is appropiate for your installation.

CHAPTER 3

Release Notes for 15.2.1

The package is designed to work with *CONTACT Elements Platform* 15.1.

3.1 Elements UI

The application has been adapted to the new Elements UI. The standard configuration allows reading access to all documents.

CHAPTER 4

Release Notes for 15.2.0

4.1 Internationalization

CONTACT Elements 15 and its applications contain many newly translated and updated terms. The following paragraphs list all affected elements in this application, while details concerning the actual conversion (called "internationalization" for simplicity) can be found in each product's release notes.

4.1.1 Translated Relationship Access Profiles

The following Relationship Access Profiles have been renamed:

Änderungsmitteilung Renamed to `Change Notification`

Please rename any usages in custom relationships accordingly.

4.1.2 Translated Catalogs

The following catalogs previously not supporting multiple languages are updated as detailed in the product release notes:

- Change Cause (`cdb_mod_event`)
- Change Source (`cdb_mod_source`)
- Change Type (`cdb_mod_kind`)

The catalogs are used in these classes, which are also updated accordingly:

- Change Notification (Document) (`aenderung`)
- Engineering Change (`cdbecm_ec`) (part of cs.ec)
- Change Notification (Part) (`cdbecm_item_ecn`) (part of cs.vp.ec_items)

4.1.3 Internationalized Catalogs

The catalog class Document Categories (`cdb_doc_categ`) has been internationalized. Affected classes:

- Document (`document`)

- Folder (cdb_folder)

4.1.4 Translated Status Names

Status names have been internationalized for the classes:

- Document (document)
- Status Protocol Document (cdb_z_statiprot)

4.1.5 Translated Catalog Values

The following catalog's values have been newly translated:

- Change Type (cdb_mod_kind)
- Change Cause (cdb_mod_event)
- Change Source (cdb_mod_source)

Keys remain unchanged.

The new translations will be automatically updated.

Note: This does only apply to catalog values not changed by any custom module.

4.2 Changes to Document Categories

Due to internationalization document categories are now referenced by id, instead of by value. This means that from now on the attributes z_categ1 and z_categ2 of the class document must contain the categ_id of the referenced category (so far they contained the german name of the category).

As long as a category is referenced by some document, it cannot be deleted. Categories now have an *obsolete* flag. If the flag is set to true the category will be ignored when generating the DOC decomposition (i.e. it will not appear in the navigation tree).

Other references to document categories from other packages have been also converted to ids. These include:

- The classes cdbpcs_cl2doctmpl, cdbpcs_cli2doctmpl, cdbpcs_prj2doctmpl, cdbpcs_task2doctmpl and cdb_folder from cs.pcs
- The object rule 2D Drawings from cs.documents
- The object rules mBOM Manager: 2D documents and mBOM Manager: 3D documents from cs.vp

Furthermore the masks and the tables of the classes document and model have been modified accordingly.

The following browsers have also been modified and return now different values:

- cdb_doc_categ_brows
- cdb_doc_categ_brows2

If you use them in your masks, please check your configuration.

On updating the exiting data must be migrated. This is done automatically by the update task cs.documents.updates.v15_2_0_2.MigrateCategories. The task works as follows:

- The existing documents are checked against the configured categories. The value of the attribute z_categ* is checked against the german name of the category.

- If a unique category can be determined for a document, the value in z_categ* is replaced by the id of the category. Otherwise a new category is generated and marked as obsolete.
- In standard there are only two level of categories, however customers can introduce more level through customizing. This scripts will migrate all the levels. Some assumptions are made:
  - The attributes in the class documents must be named according to the schema z_categ<level> (e.g. z_categ1, z_categ2, z_categ3, etc.)
- Favorites are also migrated. Searches for documents with the keys z_categ<level>=<search term> are migrated into searches with the keys category<level>_name=<search term>.

Note: Only favorites for the classes document and model and for the operations query and CDB_Search are migrated. Other classnames and operations, which may be present in your customizing, are not automatically migrated.

4.3 Change Notifications

All classes related to Document Change Notifications have been relocated from *CONTACT Engineering Changes* to *CONTACT Documents*:

- Change Notification (Document) (aenderung)
- Change Type (cdb_mod_kind)
- Change Cause (cdb_mod_event)
- Change Source (cdb_mod_source)

This is also true for the Powerscript class DocumentECN, which was relocated from cs.ec_documents to cs.documents.

4.4 Purging of file history entries on status change

After a successful status change to Released, the file history entries for all files attached to the document will be deleted. If this behaviour is not desired for an installation, the method Document.purgeFileHistoriesAfterRelease can be overwritten.

List of Figures

List of Tables

Index