

d.velop

DSR 8.1.0 (Documentation d.3
Server API 8.1.0)

Table of Contents

1. d.3 server scripting api (jpl)	4
1.1. Introduction	4
1.1.1. Conventions	4
1.1.2. About the d.3 Server API	4
1.1.3. Prerequisites	5
1.2. General information on the d.3 server API	5
1.2.1. Overview	5
1.2.2. Function call	8
1.2.3. Logging	8
1.2.4. Examples	9
1.3. Global variables	12
1.3.1. Document context	12
1.3.2. User context	13
1.3.3. Object context	14
1.3.4. Workflow context	14
1.3.5. Mailbox context	15
1.3.6. Lock context	15
1.3.7. Application context	15
1.3.8. Other variables	16
1.3.9. Error Handling	17
1.4. Functions	17
1.4.1. Overview of the functions	17
1.4.2. Document	22
1.4.3. Folder	63
1.4.4. Links	65
1.4.5. Directories	74
1.4.6. Status	79
1.4.7. Notes	90
1.4.8. Resubmission	93
1.4.9. Workflow	104
1.4.10. Document type	109
1.4.11. Datasets	113
1.4.12. User	116
1.4.13. Groups	131
1.4.14. Roles	137
1.4.15. Restriction sets	144
1.4.16. TIFF-functions	147
1.4.17. File access	150
1.4.18. String conversion	159
1.4.19. Miscellaneous	173
1.4.20. d3fc-calls	188
1.4.21. JAVA	201
1.4.22. Creating async jobs	203
1.5. JPL-Styleguide	204
1.5.1. File location and name	204
1.5.2. Comments	204
1.5.3. Header	205
1.5.4. Variables	207
1.5.5. Debugging	208
1.5.6. Loops / branches	209
1.6. Useful functions	210
1.6.1. WriteNotizEntry	210
1.6.2. AddPrefixAndZeroes	211

1.6.3. CutPrefixAndZeros	211
1.6.4. OpenSessionConnection	211
1.6.5. GroupCheckedIn	213
1.6.6. UserCheckedIn	213
1.6.7. UserIsGroup	214
1.6.8. DetermineUser	214
1.6.9. fillWmFromDbField	215
1.6.10. getValueFromDBField	216
1.6.11. hasDtsReposId	217
1.6.12. AvailableValues	218
1.7. Webservice functions	220
1.8. Additional information sources and imprint	221

1. d.3 server scripting api (jpl)

1.1. Introduction

1.1.1. Conventions

The fonts and symbols used in this document have the following meaning:

Font / Symbol	Meaning
Settings Options	Menu option or button Sub-menus are separated by the "pipe" character.
Extras Options [Miscellaneous]	The last sub-menu option in square [brackets] represents a tab to be selected.
<ALT>-<CTRL>-<X>	Key or key combination In a key combination the individual keys are separated by a hyphen. The initial keys must remain down while the last one is pressed.

Note

Sections marked with this symbol require special attention, present significant hints or exceptions.

Warning

Sections marked with this symbol help to avoid mistakes that could, in the worst case, lead to erroneous behavior of the software or loss of data.

Note

Sections marked with this symbol contain sample data for the described features or functions.

Note

Sections marked with this symbol contain additional details to certain points and provide further background to the topic under discussion.

Note

Paths to a directory, path or file name are represented in this font.

Note

Section marked like this provide a hyperlink to a video related to the subject, if available.

1.1.2. About the d.3 Server API

With the d.3 server API you can directly access to the d.3 server functions. This includes the following points:

- document-related data (properties, links ...)

- document type-related data (advanced properties ...)
- user-related data (group membership ...)

The functions mentioned above can be read and edited.

The developers can interact via the interface long-time stable functions which save the access to the d.3-data. Furthermore the API provides more auxiliary functions to allow a simplified implementation.

This documentation is available online for development partners of d.velop AG in the Service Portal. Passing this documentation or parts of it on to someone else is not permitted. For requests regarding the development partnership only the online documentation applies.

Please note that your software accesses stored and configured data in d.3ecm owned by your customers via this interface and also affects the procedures in the d.3ecm system. Please handle it with care and make sure that your application is part of an existing collaboration of several applications. The improper use of this interface can lead to changed application procedures and the loss of data.

Software development with this programming interface is an individual development. The program code generated by you is not subject to the maintenance and support conditions of the products of d.velop AG. Our support will be pleased to help you, but your requests are subject to additional charges, if it is not lead back to an error in our products.

For any questions about the prerequisites and the software development with d.3ecm, please contact the Technology Partners Management of d.velop AG.

1.1.3. Prerequisites

Warning

This manual is targeted at d.3 application developers.

Basic knowledge of programming and the d.3 system are required to understand this documentation.

1.2. General information on the d.3 server API

1.2.1. Overview

Various contexts are defined to achieve a best possible transmission of values, so there is a set of global variables, which can be set / read.

Before calling a function, e.g. the document ID can be set in form of the variable `api_doc_id` and therefore with the document context. All the functions below work with the context of this document ID. If now operations like property changes will be done, they refer always to this document ID.

The same applies for the user. He can be configured global, so that he must not be specified before each function call. If no user context is selected, the hook will be executed with the rights of the default user. The default user is dependent of the configuration parameter `SET_API_USER_MASTER`. If this variable is "1", then the default user is used as "Master", that means that the hook has all rights, if there is no other user explicit selected. If this variable is "0", then the user is used, who executed the d.3 API call, which again triggered the hook.

For the hook-users there are the following functions:

- `api_hook_user_set (user_name)`
- `api_hook_user_reset ()`
- `api_hook_user_get ()`

- `api_hook_user_master()`

Alternatively for setting the context this entries can be done immediately by the function call, so that the current context for this function call will be ignored.

To exchange the values between the global `dok_dat_feld`-fields and the `api_doc_fields` there are the following functions:

- `api_set_dok_dat_feld ()`
- `api_set_api_doc_field ()`
- `api_clear_document_context()`
- `api_clear_user_context()`

Note

From d.3 version 8.0 in the d.3 admin and the d.3 applications it is no longer spoken of attributes but of properties. The API calls referring to attributes/properties and described in this documentation were not renamed.

api_hook_user_set (user_name)

This function sets an user as executive user. Afterwards a `api_hook_user_reset` should be executed.

api_hook_user_set (user_name)	
Parameters	<code>user_name</code> Name of the user, on whose name will be worked on.
Return	0 error free -1 <code>user_name</code> was not specified -2 <code>user_name</code> is an invalid d.3 user name

api_hook_user_reset ()

This function sets the default user as executive user again. This is always called after a `api_hook_user_set`.

api_hook_user_reset ()	
Parameters	none
Return	Name of the user, who was used as user before.

api_hook_user_get ()

This function defines the executive user.

api_hook_user_get ()	
Parameters	none
Return	Name of the executing user

Note

```

proc setzeNeuenUser ()
{
  vars h_UserName
  h_UserName = api_hook_user_get ()
  if (h_UserName == "user2")
  {
    if (api_hook_user_set ("user1") < 0)
    {
      return -1
    }
  }
  // Im Folgenden arbeitet der Hook im Namen des Benutzers
  user1,
  // falls zuvor user2 Benutzer war
  call api_hook_user_reset ()
  return 0
} // end of setzeNeuenUser

```

api_hook_user_master()

This function sets the “Master” as executive user.

api_hook_user_master()	
Parameters	none
Return	none

Note

since d.3 version 6.0 HF2

api_set_dok_dat_feld ()

Sets the global dok_dat_felder with the values of the api_doc_fields (e.g. for attribute_get_valid_values)

api_set_dok_dat_feld ()	
Parameters	none
Return	0 error free

api_set_api_doc_field ()

Sets the api_doc_fields with the values of the global dok_dat_felder (e.g. for attribute_get_valid_values)

api_set_api_doc_field ()	
Parameters	none
Return	0 error free

api_clear_document_context()

clears all document related fields

api_clear_document_context()	
Parameters	none

api_clear_document_context()

Return 0 error free

api_clear_user_context()

clears all user related fields

api_clear_user_context()

Parameters none

Return 0 error free

1.2.2. Function call

To use a server-API function, the following function needs to be called:

api_function(name, param1, param2, param3, param4 ...)Parameters name: name of the function to be executed
param1..n: parameters, which are needed for the functionReturn 0: everything OK, no errors
>0 Database error
-1: Panther syntax error, details see logviewer
-2: function was called incorrect
3: undefined error, contact d.velop support**Note**

else: see the description of the respective function

There are defined information, too, which can provide the global error variables, next to the transfer parameters, which always contain these. The values recline to the API.

Transfer / Import parameterapi_error_text Error text
api_error_var[1..n] Variable name
api_error_value[1..n] Value of the variable
n is defined in the variable MAX_ERROR_VAR**1.2.3. Logging**In general for the logging the function `api_log` is available.**api_log(log_level, log_message)**Parameters log_level: Log level, typical d.3 log levels 0..9
undefined constants:
LOG_DEBUG 9
LOG_INFO 6
LOG_ERROR 0
log_message the text to be written
Return 0 error free

For certain loglevel are further functions defined, which can be used instead and which lead to the above mentioned general log function:

Log function	
<code>api_log_debug(log_message)</code>	Debug-Logging
<code>api_log_info(log_message)</code>	Information-Logging
<code>api_log_error(log_message)</code>	Error-Logging

1.2.4. Examples

Example 1

The example hook sends a resubmission to an user after a successful storage of a document.

Note

```
//-----
proc InsertExit_30 (h_DocID, h_FileDest, h_ImportOK, h_User, \
                  h_DocTypeShort)
{
  call api_log (9, \
    "+++++++ START InsertExit_30( :h_User, \
      :h_FileDest, :h_ImportOK, :h_User, :h_DocTypeShort ) \
    ++++++")

  // Schritt 1: Einschränkung auf
  // eine bzw. n Dokumentart(en)
  if (h_DocTypeShort == "ERECH")
  {
    // Schritt 2: Ermittlung eines Sachbearbeiters
    // Es wird erst einmal ein fester SB vorgegeben!
    vars h_TmpSB          = "dvelop"
    vars h_ReturnValue    = 0
    vars h_TmpRecipient   = h_TmpSB

    vars h_Subject = "Rechnung mit der Rchnungs-NR.: \
      : (dok_dat_feld[1])"
    vars h_Date    = "" // Optional!
    vars h_Sender  = "" // Falls nicht gesetzt wird der
                      // User aus dem Kontext verwendet!
                      // In diesem Fall der System-User

    // Schritt 3: Versenden einer Wiedervorlage
    // mittels der d.3-Server-API
    h_ReturnValue = api_function ("hold_file_send",
      h_TmpRecipient, \
      h_Subject, h_DokuID, \
      h_Date, "", \
      0, 1, \
      aktuelles_datum_deu, \
      "", h_Sender)

    if (h_ReturnValue != 0)
    {
      call api_log (9, "Die Wiedervorlage für das Dokument mit
der \
      Dokument-ID: :doku_id \
      konnte nicht versendet werden \
      [:h_ReturnValue]!")
    }
  }
  call api_log (0, "+++++++ ENDE InsertExit_30( :h_User, \
    :h_FileDest, :h_ImportOK, :h_User, \
    :h_DocTypeShort ) ++++++")
} // end of InsertExit 30
```

Example 2

This hook-function writes the content of `h_Worte` in the field 63 during the storage of the document.

Note

```
proc InsertEntry_10 (h_User, h_DocTypeShort)
{
  // Variablendeklaration
  vars h_ReturnValue          // Fehlercode
  vars h_Count1, h_Count2     // Zählervariablen für Schleife
  vars h_Worte = "Test1#Test2#Test3"

  //übergebenen String aufsplitten
  h_ReturnValue = api_function ("string_split", h_Worte, "#")
  if (h_ReturnValue < 0)
  {
    return 8002
  }

  // Feld 63 setzen
  h_Count2 = 1
  for h_Count1 = 1 while (h_Count1 &lt;= h_ReturnValue) step 1
  {
    if (apiSplittedString[h_Count1] != "")
    {
      dok_dat_feld_63[h_Count2] =
apiSplittedString[h_Count1]
      h_Count2 = h_Count2 + 1
    }
  }
} // end of InsertEntry_10
```

Note

```

proc InsertEntry_10 (h_User, h_DocTypeShort)
{
    // Variablendeklaration
    vars h_ReturnValue          // Fehlercode
    vars h_Count1, h_Count2     // Zählervariablen für Schleife
    vars h_Worte = "Test1#Test2#Test3"

    //übergebenen String aufsplitten
    h_ReturnValue = api_function ("string_split", h_Worte, "#")
    if (h_ReturnValue < 0)
    {
        return 8002
    }

    // Feld 63 setzen
    h_Count2 = 1
    for h_Count1 = 1 while (h_Count1 <= h_ReturnValue) step 1
    {
        if (api_splitted_string[h_Count1] != "")
        {
            dok_dat_feld_63[h_Count2] =
api_splitted_string[h_Count1]
            h_Count2 = h_Count2 + 1
        }
    }
}
} // end of InsertEntry_10

```

1.3. Global variables

1.3.1. Document context

Variable	Description
api_doc_id	Document ID
api_doc_texts[1..4]	Comment text of the single rows
api_doc_type_short	Document type short name
api_doc_field [1..89]	company specific fields
api_doc_field_60..69[1..2000]	company-specific multi-value property fields
api_doc_date_import	Import date
api_doc_last_alter_no	Latest alteration number
api_doc_publ_alter_no	Alteration number (Release)
api_doc_status	Status={"Processing", "Verification", "Release", "Archive"} (in version 5.5.1.x: doc_state)
api_doc_status_info	Additional information to the status for doc_status="Verification" = { 0 = unverified, 1 = verified}, for doc_status="Release" = { 0 = blocked, 1 = released}
api_doc_editor	Editor
api_doc_no	Document Number
api_doc_var_no	Variant number

Variable	Description
api_doc_text	Comment text Warning The global field <code>api_doc_text</code> is no longer supported. On none-unicode-systems this field can be used as before (backward compatibility) but on unicode-systems the access to this field is either reading nor writing possible. Instead, use the variable <code>api_doc_texts[1..4]</code>.
api_doc_ids[MAX_DOC_IDS]	Document ID-array
api_doc_owner	Owner
api_doc_file_name	File name
api_doc_file_ext	File extension
api_doc_encrypt	Encryption (0/1)
api_doc_cnt_access	Count access
api_doc_date_last_access	Date of the last access
api_doc_user_last_access	User of the last access
api_doc_last_update_file	Timestamp of the last change
api_doc_web_published	web published (0/1)
api_doc_workflow	in a workflow (0/1)
api_links[MAX_LINKS]	parent/child documents
api_link_creators[MAX_LINKS]	Creator of the link (a: folder scheme, m: manually, l: a00-file) (for <code>link_get_children</code> and <code>link_get_parents</code>)
api_link_tstamps[MAX_LINKS]	Time stamp, when the link was created (for <code>link_get_children</code> and <code>link_get_parents</code>)
api_doc_types[MAX_DOC_TYPES]	Document types
api_classes[MAX_CLASSES]	Array with all assigned classes
api_class_right[MAX_CLASSES]	Array with the assigned rights for the class with the same index in <code>api_classes</code> (0=denied, 1=granted, 2=not specified) Position 1: Read Position 2: Write Position 3: Validate Position 4: Release
api_color_code	Color code of the document 0: not set 1-24: Color code Note New variable from version 7.2

1.3.2. User context

Variable	Description
api_users[MAX_USERS]	User names
api_groups[MAX_GROUPS]	Group name
api_rolls[MAX_ROLLS]	User names
api_user_name	User name

Variable	Description
api_user_right_level	Rights level (1..5)
api_user_right_archive	Right to view of archive version
api_user_right_link	Right to link folders
api_user_right_flags	Right flags Position 1: Creating links Position 2: Web access permitted Position 3: User disabled Position 4: User is blocked
api_user_field[10]	user related fields
api_user_checkout_flag	User ist absent (0/1: no/yes)
api_user_checkout_text	Absence text of the user
api_user_realname	Name
api_user_email	e-mail address
api_user_phone	Phone number
api_user_plant	Plant
api_user_department	Department
api_user_tstamp_current_login	Time stamp of current login
api_user_tstamp_last_login	Time stamp of last login
api_user_tstamp_hf_info	Time stamp of the information about new resubmissions
api_user_ldap_dn	LDAP name
api_user_change_count	Alteration number
api_user_pw_valid_until	Date up to which the current password is valid
api_user_tstamp_login_failed	Time stamp of the login attempt with wrong password
api_user_count_login_failed	Count of login attemps with wrong password
api_roll_id	ID of the role
api_roll_name	Name of the role

1.3.3. Object context

Variable	Description
api_property_name	Property name
api_property_caption	Short description of the property
api_property_description	Detailed description of the property
api_property_value	Value of the property
api_object_id	Object ID of the property (e.g. user name, group name, ...)

1.3.4. Workflow context

Variable	Description
api_wfl_header_id	ID of the workflow
api_wfl_step_header_id	ID of the workflow step
api_wfl_step_exit_string	wfl_step_exit_string
api_wfl_step_tstamp_entry	Time stamp by reaching this step
api_wfl_header_name	Workflow name
api_wfl_step_recipient	Step recipient
api_wfl_step_type	Step type
api_wfl_step_action_type	Action type
api_wfl_step_action_string_1	Option1 of the action

Variable	Description
api_wfl_step_action_string_2	Option2 of the action
api_wfl_path_initiator	Path initiator of the workflow

1.3.5. Mailbox context

Variable	Description
api_hold_file_date	Date of the resubmission
api_hold_file_recipient	Recipient
api_hold_file_notice	Subject text
api_hold_file_informed	Recipient notified (0/1)
api_hold_file_tstamp_sent	Time stamp "sent"
api_hold_file_tstamp_ack	Time stamp "acknowledged"
api_hold_file_date_ack_until	Date "acknowledge until"
api_hold_file_sender	Sender
api_hold_file_date_remember	Date "remind on"
api_hold_file_deleted_sender	Deleted by sender (0/1)
api_hold_file_chain_id	Chain ID
api_hold_file_type	Type (W=Workflow)

1.3.6. Lock context

Variable	Description
api_lock_token	If a new lock has been successfully created, the respective token will be returned; it can be used to prolong or remove the lock.
api_lock_ttl	Actual period of time which the lock will be valid for (in minutes)
api_lock_object_info	If the creation of the lock failed because the object was already locked elsewhere, the info text created along with the lock will be returned here.
api_lock_user	If the creation of the lock failed because the object was already locked elsewhere, the user who created the lock will be returned here.

1.3.7. Application context

In the global variable `d3server_letzte_api_applikat` the application ID of the application will be written, which triggered the hook.

Though, the first three characters are important. A function for determination of the application ID could be look like this:

Note

```
string proc d3GetLetzteApiApplikation( h_FullFlag )
{
    ars h_TmpLetzteApiApplikation = ""
    if ( h_FullFlag != 1 )
        h_TmpLetzteApiApplikation = d3server_letzte_api_applikat(1,3)
    else
        h_TmpLetzteApiApplikation = d3server_letzte_api_applikat
    return h_TmpLetzteApiApplikation
}
```

You can react in a hook by a IF-command. If you initiate a hook from the d.3 smart explorer, then the value is "200" and by the d.3 import "220".

Note

```
if(applikationsid==200) // d.xplorer

{

}

if(applikationsid==220)// d.3 Import

{

}
```

The list below provides you an overview of the usual application ID's of the calling applications:

Application	ID
d.3 smart explorer	200
d.3 import	220
d.3 smart start	495
d.3 view	100
d.3 folder browser	499
d.capture batch	430
d.capture batch import	433
d.capture dialog	437
d.classify	501
d.cold	550
d.ecs forms	908
d.3 virtual user	253
d.3 smart suite CE	020
d.3 presentation server	901
d.3 web webservice	247
d.3 web	260
d.3 image service	483
d.exchange	460
d.link for microsoft outlook	465

1.3.8. Other variables

Other variables	Description
api_converted_string	includes the converted string of functions string_...
api splitted_string [MAX_SPLIT]	contains the splitted strings of the function string_split
api_single_info	contains a single return string (e.g. single property)
api_data_type	Data type of the dataset
api_list[]	Array for inclusion of the entries of directory_list from d.3 version 6.0.1 hotfix 1
MAX_DOC_IDS	Size of the Doc ID array
RIGHT_ANY	right of any kind
RIGHT_READ	Right to read
RIGHT_WRITE	Right to write

Other variables	Description
RIGHT_VERIFY	Right to verify
RIGHT_RELEASE	Right to release

1.3.9. Error Handling

Variable	Description
MAX_ERROR_VAR	Size of error arrays
api_error_text	Error description
api_error_var[MAX_ERROR_VAR]	Error variables array
api_error_value[MAX_ERROR_VAR]	Error variables array
LOG_ERROR = 0	Error
LOG_INFO = 6	Info
LOG_DEBUG = 9	Debug
additional_info_text	If the hook is called by a d.3 server process (not hostimp or async), then this text is passed as an additional export parameter to the client with the current API call.

1.4. Functions

1.4.1. Overview of the functions

- Document
- Folder
- Links
- Directories
- Status
- Notes
- Resubmission
- Workflow
- Document type
- Datasets
- User
- Groups
- Roles
- TIFF-functions
- File access
- String conversion
- D3fc-calls
- Miscellaneous

Note

The meaning of the return values of all d.3 server API functions can be viewed in the files `fehler.txt` and `errors.txt`. These are located on the d.3 server host in the directory `...\d3server.prg`.

Document (overview)

Document functions	Description
attribute_get_single	single attribute
attribute_get_all	all about context
attribute_update_single	single attribute

Document functions	Description
<code>attribute_update_all</code>	all about context
<code>attribute_get_valid_values</code>	Function for query allowed attribute data
<code>attribute_test_value</code>	Validate value for validity
<code>attribute_update_begin</code>	writes the current document attributes in the <code>api_doc_fields</code>
<code>attribute_update_execute</code>	changes the <code>api_doc_fields</code> after call of <code>attribute_update_begin</code>
<code>document_change_type</code>	Change document type
<code>document_get_valid_values</code>	get allowed property data
<code>document_delete</code>	Delete document
<code>document_find_ids</code>	Search document to context
<code>document_get_permission</code>	Rights of a user to a document
<code>document_publish_for_web</code>	marks a document as (not-) web-published
<code>document_dependent_add</code>	storages an dependent document or overwrites an existing document
<code>document_dependent_delete</code>	deletes a dependent document
<code>document_get_file_path</code>	determines to the predefined parameters the complete path in the d.3 document tree under which the document should stand
<code>document_start_lifetime</code>	sets the lifetime for a document
<code>document_set_cache_days</code>	sets the count of days - beginning today - , which the document stays in the document tree (fileserv); the count of days is only relevant, if the document is written to the secondary storage
<code>document_send_to_dsearch</code>	sends OCR-information of a document version to d.3 search
<code>add_inherit_doc_rights</code>	Function for the inheritance of rights to a document
<code>remove_inherit_doc_rights</code>	Function for the deletion of inherited rights to a document

Folder (overview)

Folder functions	Description
<code>folder_create</code>	Create folder

Links (overview)

Link functions	Description
<code>link_get_parents</code>	get parent documents
<code>link_get_children</code>	get child documents
<code>link_documents</code>	link two documents
<code>link_delete</code>	Delete link
<code>link_exists</code>	Test, if two documents are linked

Directory overview

Directory functions	Description
<code>directory_create</code>	Create directory
<code>directory_delete</code>	Delete directory
<code>directory_exist</code>	determines, if a directory exists
<code>directory_list</code>	List the content of a directory (max. 100 entries)
<code>directory_count_files</code>	Determine the count of files in a directory by search pattern; subfolders can be intergrated, too

Status (overview)

Status function	Description
document_block	Block document
document_verify	Verifying a document
document_transfer	Execute status transfer

Notes (overview)

Note functions	Description
note_add_file	Attach file content to a note file
note_add_string	Attach text to a note file

Mailbox (overview)

Mailbox functions	Description
hold_file_send	Send document to mailbox (resubmission)
hold_file_find	Find documents in mailbox (resubmission)
hold_file_delete	Delete document from mailbox (resubmission)
hold_file_get_document_info	Information on a resubmission

Workflow (overview)

Workflow features	Description
workpath_get_document_info	Get information to current workpath
workpath_end_document	Remove document from workpath
workpath_start_document	Set document in workpath
workpath_go_to_next_step	Move document to next step in workpath

Document types (overview)

Document type functions	Description
document_type_get_long	Define document type long text
document_type_get_short	Define document type short name

Datasets (overview)

Datasets functions	Description
predefined_values_get	Get dataset
predefined_values_test	Validate values to datasets

Users (overview)

User functions	Description
user_get	all user names
user_is_system	This function tests, if a specified user name is a system-user.
user_get_groups	Groups the user shall be a member of
user_get_verifier_groups	Groups the user shall be a verifier of
user_get_rolls	Rolls assigned to a user
user_get_data	Load data in the global variables
user_get_doc_types	Number of document types to which the user has access rights

User functions	Description
<code>user_validate_password</code>	Tests, if an user with this password is known in d.3 (not LDAP)
<code>user_get_favourites</code>	returns the document ID of the favorites of the user in the array <code>api_doc_ids</code>
<code>user_get_sid_rights</code>	determines the rights and properties for a user
<code>user_checkout</code>	Function for reporting the presence or absence of d.3 users
<code>user_get_superior</code>	Function determines the superior of a user

Groups (overview)

Group functions	Description
<code>group_get</code>	Get group name
<code>group_get_members</code>	Get members of a group
<code>group_get_verifiers</code>	Get verifiers of a group
<code>group_get_doc_types</code>	Get document types of a group

Rolls (overview)

Roll functions	Description
<code>roll_get</code>	Get all roll IDs
<code>roll_get_names</code>	Get all roll names
<code>roll_get_users</code>	Get all roll members
<code>roll_get_data</code>	Data of the role (identifiers, name, classes, rights)

Restriction sets (overview)

<code>d3set_add_filter</code>	adds a filter to an existing restriction set
<code>d3set_remove_filter</code>	removes a filter from an existing restriction set
<code>d3set_get_filter</code>	determine the filters of an existing restriction set
<code>d3set_get_sets</code>	searches restriction set associations according to specified criteria and returns them in a table
<code>d3set_remove_set</code>	deletes an existing restriction set.

TIFF functions (overview)

TIFF-functions	Description
<code>document_render</code>	Create TIFF or PDF-file
<code>tiff_create</code>	Create a TIFF-file
<code>tiff_concat</code>	Attach TIFF-file
<code>document_render_wfl_prot</code>	changes the workflow protocols to a document ID and creates a dependent w1 file.

File access (overview)

File access functions	Description
<code>file_add_string</code>	Attach a string to a file
<code>file_add_line</code>	Attach a line to a file
<code>file_read_line</code>	Read line from a file
<code>file_concat</code>	Concatenate files
<code>file_copy</code>	Copy a file
<code>file_delete</code>	Delete a file
<code>file_compare</code>	Compare files
<code>file_rename</code>	Rename files

File access functions	Description
file_exist	File exists

String conversion (overview)

String conversion functions	Description
string_to_lower	convert to lower case letters
string_to_upper	convert to upper case letters
string_to_number	Edit numeric
string_to_date	Edit date
string_split	Split string to a separator
string_cut_leading_char	remove leading characters
string_fill_leading_char	insert leading characters
string_fill_trailing_char	insert trailing characters
string_replace	replace characters in a string
string_get_position	search the position of a string in another
string_mask_special_char	mask the JAM-extended character of a string
api_str_sub	returns a part string of the passed character string
api_str_sub_utf8	returns a part string of the passed character string
api_str_sub_ansi	returns a part string of the passed character string
api_str_len	returns the number of characters in a given string
api_str_len_utf8	returns the number of characters in a given string
api_str_len_ansi	returns the number of characters in a given string
api_str_to_utf8	convert a character string to the UTF8-format
api_str_from_utf8	convert a UTF-8-string to ANSI
api_str_is_utf8	specifies, if a string in the UTF-8-format is valid
api_str_is_ascii	specifies, if a string in the ASCII-format is valid
api_str_cut_fract_utf8	this function removes a split multi-byte-character from the end of a UTF-8-encoded string.
api_str_hash	this function creates hash-values on strings

Miscellaneous (overview)

Miscellaneous functions	Description
restore_from_jukebox	restore from jukebox
restore_from_history	restore deleted document
send_email	Send e-mail
lock_token_acquire	creates a lock for an object in the database or extends an existing one
lock_token_release	removes a lock for an object in the database
start_ext_program	executes a command line call (start of an external program)
get_current_datetime	returns the current time stamp in the desired format
sleep	"sleeps" for a number of milliseconds without processor utilization
process_get_id	Returns the process ID in the global variable <code>api_single_info</code> (hexadecimal).
regular_expression_test	Test of a value to a regular expression
object_property_set	sets the property values for objects By setting a null string
object_properties_get	determines the property values for a property or for an object
set_signature_data	writes the signature data for a document into the database
get_objectid_from_synchid	Determines from a synchronization-ID, created by the transport system, the internal ID and type of Master Data-object.

D3fc-calls (overview)

D3fc-call-functions	Description
d3fc_user_set	with this function you can set an execute user for the d3fc-call
d3fc_password_set	With this function you can set a password for the execute user for the d3fc-call
d3fc_remote_server_set	Name / IP of the host, which shall be addressed
d3fc_port_set	Port of the host, which shall be addressed
d3fc_server_set	Server short name of d.3 processes
d3fc_timeout_set	Time in seconds, how long shall be waited for an answer of the remote server
d3fc_function_name_set	Name of the API call
d3fc_importing_set	Set import parameter
3fc_exporting_set	Set export parameter
d3fc_exporting_get	Get export parameter after d3fc_execute
d3fc_table_set_import_file	Set import file
d3fc_table_set_export_file	Set export file
d3fc_table_set_headline	Set table head
d3fc_table_add	Add table row to import table
d3fc_field_set	This function sets a field in the current row of the import table
d3fc_field_get	Read field in the current table row of the export table
d3fc_first	load first table row of the export table
d3fc_next	load next table row of the export table
d3fc_execute	execute an API call

JAVA (overview)

JAVA functions	Description
java_add_param	Add a parameter
java_clear_param	Delete a parameter
java_call_static	Call of a static method

1.4.2. Document

attribute_get_single

Function to query a single attribute. The return value of the attribute is written to the global field written into the `api_single_info`.

Parameters	Description
db_pos	Position of the company specific field, which is requested or name of the general attribute: doc_date_import: Import date doc_last_alter_no: Number of the last change doc_publ_alter_no: Number of the version in release doc_status: Document status doc_status_info: Additional information to the status (0/1), not verified / verified for verification, blocked / ready for release (ab 5.5.1 hotfix 14 or 6.0) doc_editor Editor doc_type_short Short name of the document type doc_no: Document number (zeich_nr) doc_var_no: Variant number doc_text_1 Comment text doc_owner Owner doc_file_name: File name doc_file_ext: File extension doc_encrypt: Encryption? (0/1) doc_cnt_access: Number of accesses doc_date_last_access: Date of the last access doc_user_last_access: User of the last access doc_last_update_file: last change of the file doc_web_published: web-published? (0/1) doc_workflow: in workflow? (0/1) doc_type: a for folder and d for document doc_color_code: Color code of the document (0 = not set, 1-24 = color code) Note new attribute doc_color_code from version 7.2 Mandatory field
row_no	Row number of multi-value fields optional, default value is 0
doc_id	Document ID, which the attribute is required for. If not specified, then the ID from the context is used. Mandatory field
user_name	Name of the user who queries the attribute (Optional)

Return value	Meaning
0	everything OK
> 0	Database error
-1	Panther syntax error, details see logviewer
-2	function was called incorrect
-3	undefined error, contact d.velop support
-101	neither document ID passed nor context set
-103	DB-position is not specified

Return value	Meaning
-201	invalid user name specified
-202	invalid DB-position specified
-203	Document <\:doc_id> of the document type <\:doc_type_short> does not contain the field <\:db-pos> api_error_var[...] = "doc_type_short" api_error-value[...] = determined document type short name for the document ID
-204	invalid row number specified
-301	invalid document ID specified or user has got no rights to it
-302	Error while determining the user rights
-303	User has got no access permission to document

Note

The function is available from d.3 version 5.5.1.

In the

- example (A) is requested, which value is saved on database position 2
- example (B) a multi-value field is requested
- example (C) the import date of the document is requested

Note

```
// -----
// Beispiel A:
// Abfragen eines einfachen Attributes
// -----
// Variablendeklaration
vars h_ReturnValue
vars h_DocIdRef = "Y000050257"
vars h_DB_PosRef = 2

// hier wird abgefragt, welchen Wert das Attribut unter
// Datenbankposition 2 (db_pos_ref=2) hat
h_ReturnValue = api_function ("attribute_get_single", \
                               h_DB_PosRef, 0, h_DocIdRef, "")

if (h_ReturnValue == 0)
{
    call api_log_info
    ("Wert von Attribut:h_DB_PosRef: :api_single_info")
}
else
{
    call api_log_error
    ("Fehlernummer:h_ReturnValue wurde zurückgeliefert!")
}
```

Note

```
//
-----
// Beispiel B:
// hier wird abgefragt welchen Wert das Mehrfach-Attribut
// in einer Zeile hat
//
-----

// Variablendeklaration
vars h_ReturnValue
vars h_DocIdRef = "Y000050257"
vars h_DB_PosRef = 62
vars attribute_row = 3

// Funktionsaufruf
h_ReturnValue = api_function ("attribute_get_single",
h_DB_PosRef, \
                                attribute_row, h_DocIdRef, "")

// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    call api_log_info ("Wert von Mehrfach-Attribut :h_DB_PosRef
                        in Zeile \
                        :attribute_row: :api_single_info")
}
else
{
    call api_log_error ("Fehlernummer :h_ReturnValue wurde
                        zurückgeliefert!")
}
```

Note

```
// -----
// Beispiel C:
// hier wird abgefragt welches Importdatum das Dokument hat
// -----

// Variablendeklaration
vars h_ReturnValue
vars h_DocIdRef = "D000000153"
vars h_AttributeName = "doc_date_import"

// Funktionsaufruf
h_ReturnValue = api_function ("attribute_get_single",
                             h_AttributeName, \
                             0, h_DocIdRef, "")

// Ergebnisausgabe
if (h_ReturnValue == 0)
{
    call api_log_info ("Importdatum des Dokuments:
:api_single_info")
}
// Fehlerausgabe
else
{
    call api_log_error ("Fehlernummer :h_ReturnValue wurde
zurückgeliefert!")
}
```

attribute_get_all

Function, that requests all attributes The values will be written into the global variables of the document context:

Global variables	Description
api_doc_field[1..89]	Attribute fields 1-89 (for 60-fields the first value is included here)
api_doc_field_60..69[1..2000]	60er attributes with all rows
api_doc_date_import	Import date
api_doc_last_alter_no	Number of the last change
api_doc_publ_alter_no	Number of the version in release
api_doc_status	Document status
api_doc_status_info	Additional information to the status: (0/1), not verified / verified for verification, blocked / ready for release (ab 5.5.1 hotfix 14 or 6.0)
api_doc_editor	Editor
api_doc_type_short	Short name of the document type
api_doc_no	Document number (zeich_nr)
api_doc_var_no	Variant number
api_doc_texts[1..4]	Comment text in single rows
api_doc_owner	Owner
api_doc_file_name	File name
api_doc_file_ext	File extension
api_doc_encrypt	Encryption? (0/1)
api_doc_cnt_access	Count access

Global variables	Description
api_doc_date_last_access	Date of the last access
api_doc_user_last_access	User of the last access
api_doc_last_update_file	last change of the file
api_doc_web_published	web-published? (0/1)
api_doc_workflow	in workflow? (0/1)
api_color_code	color code of the document (0 = not set, 1-24 = color code)

Parameters	Description
doc_id	Document ID, which the attribute is required for. If not specified, then the ID from the context is used. (mandatory field - either as parameter or in the context: api_doc_id)
user_name	Name of the user who queries the attribute (optional)

Return value	Meaning
0	everything OK
> 0	Database error
-1	Panther syntax error, details see logviewer
-2	function was called incorrect
-3	undefined error, contact d.velop support
-101	neither document ID passed nor context set
-201	invalid user name specified
-301	invalid document ID specified or user has got no rights to it
-302	Error while determining the user rights
-303	User has got no access permission to document

Note

available from d.3 version 5.5.1

In this example all attribute values of the document with the ID „D000000153“ will be written into the respective global variables and partially returned. If an error occurs, then the error number is returned.

Note

```
// -----
// Beispiel:
// Abfragen aller Dokumentattribute
// -----

// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_DocIdRef = "D000000153" // Dokumenten-ID

// hier werden alle Attribute abgefragt
// und in die globalen Variablen geschrieben.
h_ReturnValue = api_function ("attribute_get_all", h_DocIdRef,
"")

if (h_ReturnValue == 0)
{
    // hier werden nun die gleichen Ausgaben gemacht wie bei
    // den Beispielen von attribute_get_single:
    call api_log_info ("Wert von Attribut 2:
:api_doc_field[2]")
    call api_log_info ("Wert von Mehrfach-Attribut 60 in
Zeile 3: \
:api_doc_field_60[3]")
    call api_log_info ("Importdatum des Dokuments: \
:api_doc_date_import")
}
else
{
    call api_log_error ("Fehlernummer :h_ReturnValue wurde
zurückgeliefert!")
}
}
```

attribute_update_single

Change a single attribute. For more than three or four successive calls of this function `attribute_update_all` should be used to avoid to execute a validation for each single call.

Parameters	Description
db_pos	Position of the company specific field, which is requested or name of the general attribute: doc_text_1 Comment text (mandatory) doc_color_code: Color code of the document (0 = not set, 1-24 = color code)
	Note new attribute doc_color_code from version 7.2
	Mandatory field
new_value	Value for the attribute
	will be ignored by row_no = 1, else mandatory field

Parameters	Description
row_no	Row number of multi-value fields 0 = append as next row (default) -1 = for "normal" attribute fields: Use value from the respective <code>api_doc_field</code> and ignore <code>new_value</code> , for multi-value attribute fields: delete all rows of the multi-value attribute field Optionally,
doc_id	Document ID, which the attribute is required for. If not specified, then the ID from the context is used. Mandatory field - either as parameter or in the context: <code>api_doc_id</code>
user_name	Name of the user who saves the attribute (optional)

Return value	Meaning
0	everything OK
> 0	Database error
-1	Panther syntax error, details see logviewer
-2	function was called incorrect
-3	-3: unknown error, contact d.velop support
-101	neither document ID passed nor context set
-103	DB-position is not specified
-117	No new value for the mandatory field <code><\:db_pos></code> in the document type <code><\:doc_type_short></code> specified: <code>api_error_var[.] = "doc_type_short"</code> <code>api_error_value[.] = short name of the document type</code> <code>api_error_var[.] = "modifiable"</code> <code>api_error_value[.] = short name of the document type</code> <code>api_error_var[.] = "modifiable"</code> <code>api_error_value[.] = field is modifiable (0: no; 1: yes)</code> <code>api_error_var[.] = "doc_flags"</code> <code>api_error_value[.] = field properties</code> <code>api_error_var[.] = "data_type"</code> <code>api_error_value[.] = data type of the attribute</code>
-201	invalid user name specified
-202	invalid DB-position specified
-203	Document <code><\:doc_id></code> of the document type <code><\:doc_type_short></code> does not contain the field <code><\:db-pos></code> <code>api_error_var[.] = "doc_type_short"</code> <code>api_error-value[.] = determined document type short name for the document ID</code>
-204	invalid row number specified

Return value	Meaning
-206	invalid data type specified api_error_var[...] = "doc_type_short" api_error_value[...] = short name of the document type api_error_var[...] = "modifiable" api_error_value[...] = field is modifiable (0: no; 1: yes) api_error_var[...] = "doc_flags" api_error_value[...] = field properties api_error_var[...] = "data_type" api_error_value... = data type of the attribute
-215	invalid value specified adverse existing data for the field <\:db_pos> of the document type <\:doc_type_short> api_error_var[...] = "doc_type_short" api_error_value[...] = short name of the document type api_error_var[...] = "modifiable" api_error_value[...] = field is modifiable (0: no; 1: yes) api_error_var[...] = "doc_flags" api_error_value[...] = field properties api_error_var[...] = "data_type" api_error_value... = data type of the attribute
-216	invalid value specified adverse plausibility criteria for the field <\:db_pos> of the document type <\:doc_type_short> api_error_var[...] = "doc_type_short" api_error_value[...] = short name of the document type api_error_var[...] = "modifiable" api_error_value[...] = field is modifiable (0: no; 1: yes) api_error_var[...] = "doc_flags" api_error_value[...] = field properties api_error_var[...] = "data_type" api_error_value... = data type of the attribute
-301	invalid document ID specified or user has got no rights to it
-302	Error while determining the user rights
-303	User has got no access permission to document
-304	User has no permission to change the attributes of the documents
	api_error_var[...] = "doc_permission" api_error_value[...] = rights to the document.
-501	Error during determination of the document type of the document <\:doc_id>

Return value	Meaning
-515	Field <\:db_pos> must not be changed in the document type <\:doc_type_short> api_error_var[...] = "doc_type_short" api_error_value[...] = short name of the document type api_error_var[...] = "modifiable" api_error_value[...] = field is modifiable (0: no; 1: yes) api_error_var[...] = "doc_flags" api_error_value[...] = field properties api_error_var[...] = "data_type" api_error_value... = data type of the attribute
-516	Error when changing the attribute; only @dmrowcount rows in the database were changed api_error_var[...] = "doc_type_short" api_error_value[...] = short name of the document type api_error_var[...] = "modifiable" api_error_value[...] = field is modifiable (0: no; 1: yes) api_error_var[...] = "doc_flags" api_error_value[...] = field properties api_error_var[...] = "data_type" api_error_value... = data type of the attribute
< -9500	Error in customer-specific hook-function hook_upd_attrib api_error_var[...] = "hook_return_code" api_error_value[...] = return value of the hook

Note

available from d.3 version 5.5.1

Warning

In the example the attribute with the database position 2 of the document with the ID „Y000050257“ will not be updated to the value “317”. Because the parameter `h_AttributeValue` has got the value -1, the value “299” from `api_doc_field[h_DB_PosRef]` will be assigned. If an error occurs, then the error number is returned.

Note

```
// -----
// Beispiel:
// Aktualisierung eines Attributes
// -----
// Variablendeklaration
    vars h_ReturnValue      // Fehlercode
vars h_DocIdRef = "Y000050257" // Dokumenten-ID
vars h_DB_PosRef = 2        // Datenbankposition des
    betroffenen
                                // Attributs
vars h_AttributeValue = "317" // THEORETISCH neuer Attributwert,
                                // wird hier
                                // nur zur Erläuterung gebraucht;
                                // s. nächsten u. übernächsten
                                // Kommentar
vars h_RowNo = -1            // Wert aus api_doc_field[]
    verwenden,
                                // nicht h_AttributeValue!

api_doc_field[h_DB_PosRef] = "299"
// setzt den Attributwert auf den Wert,
    der in api_doc_field gesetzt wird
// h_AttributeValue wird ignoriert, da row_no = -1
h_ReturnValue = api_function ("attribute_update_single", \
    h_DB_PosRef, \
    h_AttributeValue, \
    h_RowNo, h_DocIdRef, "")

// Fehlerbehandlung des Funktionsaufrufes
if (h_ReturnValue == 0)
{
    api_single_info = ("")

    //auch hier sollte es natürlich eigentlich
    //eine Fehlerbehandlung geben
    call api_function ("attribute_get_single", \
        h_DB_PosRef, 0, h_DocIdRef, "")
    call api_log_info (\
        "Feld :h_DB_PosRef wurde erfolgreich geändert und enthält
\
        nun den Wert :api_single_info")
}
else
{
    call api_log_error ("Fehlernummer :h_ReturnValue wurde
        zurückgeliefert!")
}
}
```

This example appends the values `angehängt1`, `angehängt2` and `angehängt3` successive to the multi-value attribute field 60. To increase the understanding of this example, an error handling will be done only for the last call, although it should be done for each call.

For more than three or four successive calls of this function `attribute_update_all` should be used to avoid to execute a validation for each single call.

Note

```
// -----
// Beispiel A:
// Aktualisierung eines Mehrfach-Attributfeldes
// -----

// Variablendeklaration
vars h_ReturnValue           // Fehlercode
vars h_DocIdRef = "Y000050257" // Dokumenten-ID
vars h_DB_PosRef = 60         // Datenbankposition des
Attributs
vars h_RowNumber = 0         // Zeilennummer, in die
geschrieben
                                // werden soll
                                // 0 = anhängen
vars h_RowCounter = 1        // Zähler

api_doc_field[h_DB_PosRef] = ""

// hängt nacheinander die Werte "angehängt 1", "angehängt 2" und
// "angehängt 3" an das Mehrfach-Attributfeld 60
h_ReturnValue = api_function ("attribute_update_single", \
                                h_DB_PosRef, "angehängt 1", \
                                h_RowNumber, h_DocIdRef, "")

h_RowNumber = 1
h_ReturnValue = api_function ("attribute_update_single", \
                                h_DB_PosRef, "angehängt 2", \
                                h_RowNumber, h_DocIdRef, "")

h_RowNumber = 2
h_ReturnValue = api_function ("attribute_update_single",
                                h_DB_PosRef, \
                                "angehängt 3", \
                                h_RowNumber, h_DocIdRef, "")

// Fehlerbehandlung für den vorherigen Funktionsaufruf
if (h_ReturnValue == 0)
{
    call api_log_info ("Im folgenden die Werte aller Zeilen des \
                        Mehrfach-Attributs :h_DB_PosRef")
    while (h_RowCounter > 0)
    {
        api_single_info = ""

        // auch hier sollte es natürlich eigentlich eine
        // Fehlerbehandlung geben
        call api_function ("attribute_get_single", h_DB_PosRef, \
                                h_RowCounter,
                                h_DocIdRef,
                                "")

        if (api_single_info != "")
        {

```

```
        call api_log_info ("Wert von Mehrfach-Attribut \  
                           :h_DB_PosRef, Zeile \  
                           :h_RowCounter: :api_single_info")  
        h_RowCounter = h_RowCounter + 1  
    }  
    else  
    {  
        h_RowCounter = 0  
    }  
}  
else  
{  
    call api_log_error ("Fehlernummer :h_ReturnValue wurde  
                        zurückgeliefert!")  
}
```

Note

```
//
-----

// Beispiel B:
// Löschen und anschließendes Aktualisieren eines
// Mehrfachattributes
//
-----

// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_DB_PosRef = 60       // Datenbankposition des
                             // betroffenen
                             // Attributes
vars h_DocIdRef = "D000000266" // Dokumenten-ID
vars h_RowNo = -1           // Da Mehrfachfeld --> -1 =
alle                         // löschen

//leert alle Einträge des Mehrfach-Attributs 60, da row_no = -1
h_ReturnValue = \
  api_function ("attribute_update_single", h_DB_PosRef, "", \
                                                  h_RowNo, \
                                                  h_DocIdRef, "")

// Variablen für zweiten Aufruf
h_RowNo = 0
vars h_NewValue = "ABCD, Ltd. "

// Fügt den Wert "ABCD, Ltd. " an
h_ReturnValue = \
  api_function ("attribute_update_single", h_DB_PosRef, \
                                                  h_NewValue, \
                                                  h_RowNo, \
                                                  h_DocIdRef, "")

// Fehlerbehandlung
if (h_ReturnValue == 0)
{
  vars row_counter = 1

  call api_log_info ("Im folgenden die Werte aller Zeilen des \
                      Mehrfach-Attributs :h_DB_PosRef")
  while (row_counter > 0)
  {
    api_single_info = ""
    // auch hier sollte es natürlich eigentlich eine
    // Fehlerbehandlung geben
    call api_function ("attribute_get_single", h_DB_PosRef,
\
                                                  row_counter, \
                                                  h_DocIdRef, "")

    if (api_single_info != "")
    {

```

```

        call api_log_info ("Wert von Mehrfach-Attribut \
                           :h_DB_PosRef, \
                           Zeile :row_counter: \
                           :api_single_info")
        row_counter = row_counter + 1
    }
    else
        row_counter = 0
    }
}
else
{
    call api_log_error ("Fehlernummer :h_ReturnValue wurde
                        zurückgeliefert!")
}

```

attribute_update_all

Function to change all attributes in the context (comment field = api_doc_texts1..4]; attribute = api_doc_field...] and api_doc_field_6x[...])

Parameters	Description
doc_id	Document ID, which the attributes will be changed for. If not specified, then the ID from the context is used. Mandatory field
user_name	Name of the user who changes the attribute (Optional)

Return value	Meaning
0	everything OK
> 0	Database error
-1	Panther syntax error, details see logviewer
-2	function was called incorrect
-3	undefined error, contact d.velop support
-101	neither document ID passed nor context set
-117	No new value specified for the mandatory field <\:db_pos> in the document type <\:doc_type_short> api_error_var[...] = "doc_type_short" api_error_value[...] = short name of the document type api_error_var[...] = "db-position" api_error_value[...] = <Position> - missing value for mandatory field
-201	invalid user name specified
-215	invalid value specified adverse dataset for the field <\:db_pos> of the document type <\:doc_type_short> api_error_var[...] = "doc_type_short" api_error_value[...] = short name of the document type api_error_var[...] = "db-position" api_error_value[...] = <Position> - invalid value for mandatory field

Return value	Meaning
-216	invalid value specified adverse plausibility criteria for the field <\:db_pos> of the document type <\:doc_type_short> api_error_var[..] = "doc_type_short" api_error_value[..] = short name of the document type api_error_var[..] = "db-position" api_error_value[..] = <Position> - invalid value as per plausibility check
-301	invalid document ID specified or user has got no rights to it
-302	Error while determining the user rights
-303	User has got no access permission to document
-304	User has no permission to change the attributes of the document api_error_var[..] = "doc_permission" api_error_value[..] = rights to the document.
-501	Error during determination of the document type of the document <\:doc_id>
-516	Error when changing the attribute; only @dmrowcount rows in the database were changed api_error_var[..] = "doc_type_short" api_error_value[..] = short name of the document type
<- 9500	Error in customer-specific hook-function hook_upd_attrib ... api_error_var[..] = "hook_return_code" api_error_value[..] = return value of the hook

Note

available from d.3 version 5.5.1 hotfix 1

The first example changes the attribute fields 1 and 60 and overwrites them with new values. All other attribute values will be lost because the respective global variables are empty after clearing the document context. The attribute fields will be overwritten with an empty attribute value.

Note

```
// -----
// Beispiel A:
// Aktualisierung der Attributfelder 1 und 60 bei
// gleichzeitigem Verlust anderer Attributwerte
// -----

// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_DocIdRef = "D000000153" // Dokumenten-ID

// leert den Dokument-Kontext der Server API
call api_clear_document_context ()

// füllen des API-Dokument-Kontextes mit den neuen Attributwerten
api_doc_field[1] = "42"
api_doc_field_60[1] = "609"
api_doc_field_60[2] = "607"

// setzt die Attribute des Dokuments auf die Werte des
// API-Dokument-Kontextes
h_ReturnValue = api_function ("attribute_update_all",
h_DocIdRef, "")

// Fehlerbehandlung
if (h_ReturnValue != 0)
{
    call api_log_error ("Fehlernummer :h_ReturnValue wurde
        zurückgeliefert!")
}
```

In the second example after clearing the document context the document attributes will read into the global variables at first. Afterwards function will be called. This way, the original attribute values will still be there.

Note

```
// -----
// Beispiel B:
// Aktualisierung der Attributfelder 1 und 60 OHNE
// gleichzeitigem Verlust anderer Attributwerte
// -----

// Variablendeklaration
vars h_ReturnValue           // Fehlercode
vars h_DocIdRef = " D000000153" // Dokumenten-ID

// leert den Dokument-Kontext der Server API
call api_clear_document_context ()

// füllt den API-Dokument-Kontext mit den aktuellen
dok_dat_feldern.
// In Hooks werden damit meist die Werte des aktuellen Dokuments
// in den API-Dokument-Kontext geschrieben.
call api_set_api_doc_field ()

// ändern des API-Dokument-Kontextes mit den Werten,
// die aktualisiert werden sollen
api_doc_field[2] = "73000"
api_doc_field_61[1] = "Wert 1"
api_doc_field_61[2] = "Wert 2"

// setzt die Attribute des Dokuments auf die Werte des
// API-Dokument-Kontextes
h_ReturnValue = api_function ("attribute_update_all",
h_DocIdRef, "")

// In diesem Fall bleiben die alten Attribut-Werte also
// erhalten,
// weil sie zunächst in den API-Kontext geladen werden.
// Nur die explizit geänderten Felder bekommen neue Werte.

// Fehlerbehandlung
if (h_ReturnValue != 0)
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
    zurückgeliefert!")
}
```

attribute_get_valid_values

Function for query allowed attribute data. The global array `d3server_field_no_allowed[]` gets the db position of the respective field, the values will be written into the arrays `d3server_value_char_allowed[]`, `d3server_value_date_allowed[]`, `d3server_value_num_allowed[]` depending on data type. The number of allowed values retrieved will be specified in `api_single_info`.

Parameters	Description
doc_type_short	Short name of the document type, the attribute belongs to, which values shall be requested Mandatory field
db_pos	Position of the field, which values shall be requested, in the document type Mandatory field
user_name	User for whom the note shall be written; if nothing is specified, then the executive user will be used optional
row_no	Row number for 60 field in cases where the values will be determined by hook-functions optional - an dataset-hook will be passed

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see logviewer
-2	function was called incorrect
-3	-3: unknown error, contact d.velop support
-104	Document type short name is not specified
-201	invalid user name specified
-303	User has got no access permission to document
-517	Error while loading the values for the attribute
-518	Error during determination of the index for the document type
-519	for attribute or all attributes a value selection is not possible

This example displays all valid values of the dataset to position 42 of the document type with the document short name „DACAD“. If an error occurs, then the error number is returned.

Note

```
// -----
// Beispiel:
// Ausgabe gültiger Werte einer Wertemenge
// -----

// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_DocTypeRef = "DACAD" // Dokumentartkürzel
vars h_DBPosRef = 42        // Datenbankposition des Attributs
vars h_Counter              // Zähler für Schleife

// Ausführen der Funktion
h_ReturnValue = api_function( "attribute_get_valid_values", \
                              h_DocTypeRef, \
                              h_DBPosRef, "" )

// Fehlerbehandlung
if (h_ReturnValue == 0)
{
    call api_log_info("Wertemenge enthält :api_single_info \
                      gültige Elemente:")
    for h_Counter = 1 while h_Counter <= api_single_info step 1
    {
        call api_log_info("Element :h_Counter: \
:d3server_value_char_allowed[h_Counter]")
    }
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
zurückgeliefert!")
}
}
```

attribute_test_value

Function checks a value for validity and returns 0, if the value is invalid.

Parameters	Description
doc_type_short	Short name of the document type, the attribute belongs to, which values shall be requested (Mandatory field)
db_pos	Position of the field, which values shall be requested, in the document type (Mandatory field)
new_value	Value, which validity should be checked (Mandatory field)
user_name	User for whom the note shall be written; if nothing is specified, then the executive user will be used (optional)
row_no	Row number for 60-field in cases where the values will be determined by hook-functions (optional - necessary for 60-fields)

Return value	Meaning
0	everything OK
> 0	Database error
-1	Panther syntax error, details see logviewer
-2	function was called incorrect
-3	unknown error, contact d.velop support
-103	DB-position is not specified
-104	Document type short name is not specified
-117	a new value for the mandatory field <\:db_pos> in the document type <\:doc_type_short> api_error_var[...] = "doc_flags" api_error_value[...] = field properties api_error_var[...] = "data_type" api_error_value[...] = data type of the attribute
-201	invalid user name specified
-203	Document type <\:doc_type_short> does not contain the field <\:db_pos>.
-215	invalid value specified adverse existing data for the field <\:db_pos> of the document type <\:doc_type_short> api_error_var[...] = "doc_flags" api_error_value[...] = field properties api_error_var[...] = "data_type" api_error_value[...] = data type of the attribute api_error_var[...] = "new_value" api_error_value[...] = if so purified value
-216	invalid value specified adverse plausibility criteria for the field <\:db_pos> of the document type <\:doc_type_short> api_error_var[...] = "doc_flags" api_error_value[...] = field properties api_error_var[...] = "data_type" api_error_value[...] = data type of the attribute api_error_var[...] = "new_value" api_error_value[...] = if so purified value
-303	User has got no access permission to document

In this example it will be checked, if the test value “Meier” is valid as value for the attribute with the database position 5 of the document type “DER”. If an error occurs, then the error number is returned.

Note

```
// -----
// Beispiel:
// Validierung eines Attributwertes
// -----

// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_DocTypeRef = "DER"   // Dokumentenkürzel
vars h_DB_PosRef = "5"      // Datenbankposition
vars h_TestValue= "Meier"   // Testwert
vars h_Count = 1            // Zähler für Schleife

// Wert auf Gültigkeit prüfen
h_ReturnValue = api_function ("attribute_test_value",
h_DocTypeRef, \
                                h_DB_PosRef, h_TestValue, "")

// Ergebnisausgabe
if (h_ReturnValue == 0)
{
    call api_log_info("Abgefragter Wert ist gültig!")
}
// Fehlerausgabe
else if ((h_ReturnValue == 117) || \
        (h_ReturnValue == 215) || \
        (h_ReturnValue == 216))
{
    while (api_error_var[h_Count] != "")
    {
        call api_log_info(":api_error_var[h_Count]: \
                            :api_error_value[h_Count]")
        h_Count = h_Count + 1
    }
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
        zurückgeliefert!")
}
}
```

attribute_update_begin**Note**

new in d.3 version 7.0

With `attribute_update_begin` the current attributes of the document will be written into the `api_doc_fields`.

Referring to that the `api_doc_fields` can be changed.

The `dok_dat_feld` will NOT be filled and changes on the `dok_dat_feld` will also not be considered in the next step during the call of `attribute_update_execute`. With this function the usual process

of the updating will be executed. During this the condition of the `api_doc_fields` will be verified for validity and then written in the database. The same as for other attribute updates is valid for the update-hooks.

But consider two things:

- the `dok_dat_feld` is not filled with the values of the document. So it is necessarily to go back to the `api_doc_fields`. To find out, if exactly this form of update is active, you can query on `api_attribute_update_active == 1`.
- To avoid a mutual overwrite of the `api_doc_fields`, a recursively call of this function is not possible; a respective error message will be displayed in the log.

Warning

If `SERVER_API_NO_HOOKS` is activated, then the update-hooks will not be executed.

Parameters	Description
<code>doc_id</code>	Doc ID of the document whose attributes are to be changed
<code>user</code>	User executing the attribute update

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-101	no <code>doc_id</code> specified
-304	User has no permission to change the attributes
-640	too many recursive attribute updates
-641	unknown <code>doc_id</code> specified
-650	an attribute update is already executing via server API (Pair <code>attribute_update_begin/_execute</code>)

attribute_update_execute

Description: see [attribute_update_begin](#)

Parameters: None

Prerequisite for execution is a previous successful [attribute_update_begin](#)

(handle is known internally, `doc_id` and `user` will be taken from [attribute_update_begin](#) and returned in d.3 logviewer)

Return value	Meaning
0	everything OK
>0	Database error
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-203	Document type does not contain the field (see d.3 logviewer)
-216	Plausibility check failed (see d.3 logviewer)
-224	an attribute value is too long (see d.3 logviewer)
-651	There is no attribute update active, which could be executed here

document_change_type

Function to change the document type

Parameters	Description
document_type_short	Short name of the new document type Mandatory field
doc_id	ID of the document, whose document type shall be changed Mandatory field
user_name	Name of the changing user; if nothing is specified, then the executive user will be used optional

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-3	undefined error, contact d.velop support
-101	neither document ID passed nor context set
-104	Document type short name is not specified
-201	invalid user name specified
-205	Invalid document type short name specified
-301	invalid document ID specified or user has got no rights to it
-302	Error while determining the user rights
-303	User has got no access permission to document
-304	User has no permission to change the attributes of the documents <code>api_error_var[.] = "doc_permission"</code> <code>api_error_value[.] = rights to the document.</code>
-505	Error saving the attributes
< -9500	Error in customer-specific hook-function <code>hook_upd_attrib</code> <code>api_error_var[.] = "hook_return_code"</code> <code>api_error_value[.] = return value of the hook</code>

In the following example the document type of the document with the identifiers „D000000513“ will be changed to the document type with short name „DER“, if no error occurs.

Note

```
// -----
// Beispiel:
// Ändern der Dokumentart
// -----

// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_DocIdRef = "D000000153" // Dokumenten-ID
vars h_NewDocType = "DER"    // Neuer Dokumententyp

// Ändern der Dokumentart
h_ReturnValue = api_function ("document_change_type",
h_NewDocType , \

h_DocIdRef, "")

// Fehlerbehandlung
if (h_ReturnValue == 0)
{
    call api_log_info("Die Dokumentart wurde erfolgreich zu \
                        :api_doc_type_short geändert.")
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
                        zurückgeliefert!")
}
```

document_delete

Function deletes a document from the d.3 system.

Parameters	Description
reason	Reason for deletion, necessary for release/archive-version Mandatory field for the status release/archive, else optional
del_from_each_status	delete from all status 0 = only the current version 1 = all versions optional, default: 0
del_file_always	Delete document file independent from status 0 = delete only document files in editing/verification 1 = delete document file in each case optional, default: 0
doc_id	ID of the document to be deleted; if nothing if nothing is passed, the ID of the context is used Mandatory field, if not specified in the context (api_doc_id)
user_name	User for whom the document shall be deleted; if nothing is specified, then the executive user will be used optional

Parameters	Description
del_privileged	If this is set to "1", then the respective license will be checked and the right of the executive user to delete privileged will be tested; if this is not the case, the errors 319 and 320 occur.

Note
 New in version 7.0

Return value	Meaning
0	everything OK
> 0	Database error
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-3	undefined error, contact d.velop support
-101	neither document ID passed nor context set
-120	Reason for deletion is not specified
-201	invalid user name specified
-301	invalid document ID specified or user has got no rights to it
-302	Error while determining the user rights
-303	User has got no access permission to document
-311	User has no right to delete the document <code>api_error_var[..] = "doc_permission"</code> <code>api_error_value[..] = right</code>
-319	no license to delete privileged
-320	User is not permitted to delete the document privileged
<-9500	Error in customer-specific hook-function <code>hook_delete</code> <code>api_error_var[..] = "hook_return_code"</code> <code>api_error_value[..] = return value of the hook</code>

This example deletes the document with the ID „Y000050256” from the archive, if it is in processing or verification. If an error occurs, then the error number is returned.

Note

```
// -----
// Beispiel:
// Löschen eines Dokumentes
// -----

// Deklaration von Variablen
vars h_ReturnValue                                // Fehlercode
vars h_DocIdRef ="Y000050256"                    // zu löschendes
Dokument
vars h_ReasonText = "Dokument wurde doppelt importiert"
                                                    // Begründung für
Löschung
vars h_UserRef= "dvelop"                          // Benutzer

// Ausführen der Funktion: Löschen eines Dokumentes
h_ReturnValue = api_function ("document_delete",
                              h_ReasonText, 0, 0, \
                              h_DocIdRef, h_UserRef)

// Fehlerbehandlung
if (h_ReturnValue == 0)
{
    call api_log_info("Dokument gelöscht!")
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
    zurückgeliefert!")
}
```

document_find_ids

Function searches documents to the context. No full-text search; no call of search-hook, no dataset-verification; no case-conversion.

In `api_single_info` the count of found document-IDs will be entered This will be deposited in the array `api_doc_ids[]`.

Consideration of the following context variables:

context variables	Description
<code>api_doc_field[1..89]</code>	Attribute fields 1-89
<code>api_doc_type_short</code>	Short name of the document type
<code>api_doc_status</code>	current status of the document:
<code>api_doc_editor</code>	Editor of the document

Note

The `api_error_var[...]` / `api_error_value[...]` contain the considered search criteria.

Note

This function only provides a rudimentary search functionality.

To map more complex search scenarios (with full text, rights check or wildcards), please use the `docSearch` functions described in the d.3 hook documentation.

Parameters	Description
<code>user_name</code>	User for whom shall be searched; if nothing is specified, then the executive user will be used optional

Return value	Meaning
0	everything OK
> 0	Database error
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-121	no search criteria specified or only the dummy document type

The example searches for document-IDs of the global variables `api_doc_type_short` and

`api_doc_field[2]`. So that only this both variables are used for the search, the document context is cleared before. Afterwards the total number and the IDs of the found documents will be displayed. If an error occurs, then the error number is returned.

Note

```
// Suche nach Dokumenten
// -----
// Variablendeklaration
vars h_ReturnValue      // Fehlercode
vars h_Count            // Zählervariable für die Schleife
// Löschen des API-/Dokumenten-Kontextes
call api_clear_document_context()

//Suchkriterien im API-Dokument-Kontext setzen
api_doc_type_short = "D405"
api_doc_field[2] = "73000"

// Ausführen der Funktion: Suchen von Dokumenten(-IDs)
h_ReturnValue = api_function ("document_find_ids", "")

// Fehlerbehandlung und Ausgabe des Ergebnisses
//
// Ausgabe der Anzahl der Ergebnisse und der gefundenen
// Dokumenten-IDs
if (h_ReturnValue == 0)
{
    call api_log_info("Es wurden \
                        :api_single_info Dokument-IDs gefunden!")
    for h_Count = 1 while h_Count <= api_single_info step 1
    {
        call api_log_info("doc_id :h_Count:
:api_doc_ids[h_Count]")
    }
}
// Fehlerbehandlung
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
zurückgeliefert!")
}
```

document_get_permission

The function specifies the rights to query of a user to a document The specification will be written into the `api_single_info` in the form, which is described in the API-call `SearchDocument (doc_permission)`: each with 1: Yes, 0: No a1a2a3a4a5a6a7a8-b1b2b3b4b5b6b7b8-c1c2-d1d2 mit:

Parameters	Meaning
a1	Access to document type permitted
a2	Note file exists
a3	Document links exist
a4	Document file exists
a5	Document file is encrypted
a6	Version in release exists (1: released, 2: blocked)
a7	newest version in verification (1: verified, 2: not verified)
a8	At least one version in status archive
b1	Right to visualize the document file

Parameters	Meaning
b2	Right to change the attributes
b3	Right to change the document files
b4	Right to delete the document
b5	Right for status transfer
b6	Right to verify the document
b7	Right to release/block the document
b8	Right to execute both verification steps in one step (verification + release)
c1	Right to visualize older version in release
c2	Right to release/block older version in release
d1	Right to see versions in the status archive.
d2	Document is in the workflow

Parameters	Description
doc_id	ID of the document, which rights shall be queried. If nothing is passed, then the ID of the context is used. Mandatory field - either as parameter or in the context as <code>api_doc_id</code>
user_name	User, whom rights shall be queried; if nothing is specified, then the executive user will be used optional

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer

Note

available from d.3 version 5.5.1 hotfix 14, 6.0

This example queries the rights for the document “Y000050257” and returns them coded. If an error occurs, then the error number is returned.

Note

```
// -----
// Beispiel:
// Berechtigungsabfrage
// -----

// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_DocIdRef = "Y000050257" // Dokumenten-ID
vars h_UserName = "dvelop"    // Benutzername

// Funktionsaufruf
h_ReturnValue = api_function ("document_get_permission",
h_DocIdRef, \

h_UserName )

// Fehlerbehandlung
if (h_ReturnValue == 0)
{
    call api_log_info("Berechtigungen: :api_single_info")
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
    zurückgeliefert!")
}
```

document_publish_for_web

Marks a document as (not-) web-published

Parameters	Description
publish	0: set document as not web-published 1: set document as web-published (default)
doc_id	Document to be published
user_name	User who does the publishing

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer

Note

available since d.3 version 6.2

This example marks the document with the ID "Y000005174" as web-published. In case of an error the error code will be returned.

Note

```
// -----
// Beispiel 1:
// Dokument markieren als „web-veröffentlicht“
// -----
// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_DokID = "Y000005174" // betroffenes Dokument
// Funktionsaufruf
h_ReturnValue = api_function ("document_publish_for_web", 1,
h_DokID)

// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    call api_log_info("Dokument veröffentlicht.")
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
    zurückgeliefert!")
}
```

This example marks the document with the ID "Y000005174" as not web-published.

In case of an error the error code will be returned.

Note

```
// -----
// Beispiel 2:
// Dokument markieren als „nicht-web-veröffentlicht“
// -----

// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_DokID = "Y000005174" // betroffenes Dokument

// Funktionsaufruf
h_ReturnValue = api_function ("document_publish_for_web", 0,
h_DokID)

// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    call api_log_info("Dokument ist nicht mehr veröffentlicht.")
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
    zurückgeliefert!")
}
```

document_dependent_add

Note

New in version 7.1:

The function can no longer store a signature-file (.s1) for a document. The error -670 will be returned in this case.

Stores a dependent document or overwrites an existing document. The corresponding file must not be created in the new document tree, but should ideally be located in the temporary directory.

Parameters	Description
filename	Name of the new dependent file
doc_status	Status of the dependent document to be stored (B for Processing, P for Verification, F for Release) Default: Newest version
doc_ext	Extension of the dependent file (default: extension of filename)
doc_id	Document that belongs to the dependent document
user_name	Name of the executing user

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-101	no document ID specified
-150	filename is not specified
-214	invalid status specified (must be B, P or F)
-250	File not found
-313	User is not allowed to change document files
-350	Access to files revoked
-555	Error while renaming the file
-556	Error while copying the file
-617	Error while polling the current status from the database
-670	the function can no longer store a signature-file (.s1) for a document.

Stores a dependent document or overwrites an existing document.

Note

```
// -----
// Beispiel:
// Neues abhängiges Dokument anfügen
// -----
// Variablendeklaration
vars h_ReturnValue           // Fehlercode
vars h_Filename = "c:\\test.tif" // abhängige Datei
vars h_DocStatus = "B"        // Status zu dem die Ablage
                               // erfolgen soll
vars h_DocExt = "T1"          // Dateiendung der abhängigen
Datei
vars h_DokID = "Y000050256"   // Dokument zu dem das Dokument
                               // angehängt werden soll

// Funktionsaufruf
h_ReturnValue = api_function ("document_dependent_add",
h_Filename, \

h_DocStatus, \

h_DocExt, \

                               h_DokID)

// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    call api_log_info("Abhängige Datei abgelegt!")
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
zurückgeliefert!")
}
```

document_dependent_delete

Deletes a dependent document.

Parameters	Description
doc_status	Status of the dependent document to be deleted (B for Processing, P for Verification, F for Release) default: newest version
doc_ext	File extension of the dependent file
doc_id	Document ID of the dependent document
user_name	Name of the executing user

Return value	Meaning
0	everything OK
>0	Database error
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-101	no document ID specified
-139	no doc_ext specified

Return value	Meaning
-214	invalid status specified (must be B, P or F)
-313	User is not allowed to change document files
-350	Access to files revoked
-617	Error while polling the current status from the database

Deletes a dependent document.

Note

```
// -----
// Beispiel:
// Neues abhängiges Dokument löschen
// -----
// Variablendeklaration
vars h_ReturnValue      // Fehlercode
vars h_DocStatus = "B" //Status aus dem die abhängige Datei
                        // gelöscht werden soll
vars h_DocExt = "T1"      // Dateiendung der abhängigen
Datei
vars h_DokID = "Y000050256" // Dokument zu dem das
                        // Dokument abgehängt werden
soll

// Funktionsaufruf
h_ReturnValue = api_function ("document_dependent_delete",
                              h_DocStatus, \
                              h_DocExt, \
                              h_DokID)

// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    call api_log_info("Abhängige Datei gelöscht!")
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
zurückgeliefert!")
}
```

document_get_file_path

This function is used to determine the complete path of a document in the d.3 document tree.

The variable `api_single_info` contains the found path.

Parameters	Description
<code>doc_id</code>	Base document for which the path shall be determined.
<code>status</code>	Status of the document: "Processing", "Verification", "Release" or "Archive"
<code>archive_index</code>	Index of the archive version is required for <code>status = "archive"</code>
<code>user_group</code>	If this parameter is passed, then it is prior to the actual editor of the document specified under <code>doc_id</code> ; only relevant for status processing.

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-115	no index or archive version specified for status archive
-126	either document ID passed nor context set

Note

available from d.3 version 6.1.1

document_start_lifetime

Sets the lifetime for a document.

Parameters	Description
<code>doc_id</code>	Document ID of the document.
<code>overwrite_old_date</code>	Overwrite the current value (default: 1)
<code>life_time_days</code>	If the configured lifetime for the document types shall not be entered, then you can specify the number of days here (default 0 - configured lifetime).

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-101	no <code>doc_id</code> specified
-621	It is not intended to change the lifetime of the document type and to pass <code>overwrite_old_date</code> with 0.
-622	Database-error; see logviewer for details

Sets the lifetime of the document:

Note

```
// -----
// Beispiel:
// Lebensdauer eines Dokumentes auf ca. 10 Jahre setzen
// -----
// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_DokID = "Y000050533" // Dokument,
                             // dessen Lebensdauer gesetzt werden
                             // soll
vars h_Overwrite = "1"      // Überschreiben der aktuellen
                             // Lebensdauer
                             // (default 1)
vars h_Doc_life_time_days = "3560" // Lebensdauer, Anzahl in
Tage

// Funktionsaufruf
h_ReturnValue = api_function ("document_start_lifetime", \
                             h_DokID, \
                             h_Overwrite, \
                             h_Doc_life_time_days)
                             //ca. 10
Jahre
// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    call api_log_info("Lebensdauer des Dokumentes gesetzt!")
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
zurückgeliefert!")
}
```

document_set_cache_days

Sets the number of days that a document version should remain in the d.3 document tree "from today". This number of days is only relevant, if the document was or will be written to a secondary storage.

Parameters	Description
doc_id	Document ID of the document.
status	Status of the document: "Release" or "Archive" (default "Release")
archive_index	Index of the archive version is required for status = "archive"
	Mandatory field for the status "archive", will not be considered for "release"
days_in_cache	Days from now on the document version remains in the cache
	Mandatory field

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-115	no index or archive version specified for status archive

Return value	Meaning
-223	invalid status specified (must be F or A)
-623	Number of days that the document version should remain in the d.3 document tree, not specified
-624	Document version does not exist or has already been deleted from the d.3 document tree

Note

This function is available from d.3 version 6.3.

document_send_to_dsearch

Sends OCR-information of a document version to d.3 search.

Parameters	Description
<code>doc_id</code> (P)	Document to be updated in d.3 search.
<code>ocr_file</code>	OCR-file containing the OCR-information. If the attributes are to be transferred or <code>use_existent_ocr_file</code> is enabled, then this parameter does not need to be filled.
<code>version</code>	Version of the document to be changed. If not specified, then the current version is used. For note files, the constant <code>not</code> must be passed here.
<code>dsearch_corpus</code>	Corpus to which the document is to be passed. Must be set, if an existing document is to be transferred to a different/new corpus.
<code>use_existent_ocr_file</code>	Set this parameter to use existing OCR-files. This can be set to pass existing documents to d.3 search again. (Recovery, new fulltext engine, ...) default: 0
<code>doc_status</code>	Status of the document version to be updated. If not specified, then the current document version is used.
<code>archive_index</code>	Archive index of an archive version. Only has to be passed, if the <code>doc_status</code> is "archive".
<code>user_name</code>	Name of the executing user

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-101	no <code>doc_id</code> specified
-313	User is not allowed to change document files
-660	no <code>DSEARCH_SUPPORT</code>
-661	Error creating the OCR-sub directory
-662	Error copying the OCR-file
-663	d.3 search reports an error, see d.3 logviewer for the error number.

add_inherit_doc_rights

Function for the inheritance of rights to a document.

Parameters	Description
<code>doc_id</code>	Document for which the rights are to be inherited
<code>granter</code>	User who inherited the right
<code>grantee</code>	User to whom the rights are inherited.
<code>right_flags</code>	Position 1: 1=Read rights, 2=Write rights (implies read rights) Position 2: 1 = expires after time stamp, 2 = expires after acknowledgement Position 3: 1= Recursive (affects all documents linked to the document as children)

Parameters	Description
<code>tstamp_expire</code>	Time stamp for expiry (format <code>DD.MM.YYYY - hh:mm:ss</code>). Empty, if <code>right_flags[2] = 2</code>

Return value	Meaning
0	everything OK
3	invalid or missing parameter.
120	User (<code>granter</code> or <code>grantee</code>) does not exist.
450	<code>granter</code> is not allowed to inherit rights.
451	<code>granter</code> does not have the rights to be inherited.
452	<code>grantee</code> already has the permissions to be inherited.
9000	Database error

Note

```
//-----

-----//

// NAME

// rechtevererben.jpl

//

// DESCRIPTION

//

// this JPL-file is an example for the function
// add_inherit_doc_rights. In this example
// grantee from user Adminis1 the right to write to the document
// P000000792 inherited.

//

//-----//

wfl_hook_content_type = "text/string"

vars tmpMasterAPI = SET_API_USER_MASTER

SET_API_USER_MASTER = 1

vars h_ReturnValue = 0

vars wf_DokuID

vars h_granter

vars h_grantee

vars h_ReadWriteFlag

vars h_DateFlag

vars h_Rekursive

vars h_InheritDate

wf_DokuID="P000000792"

//add_inherit_doc_rights

h_granter = "Adminis1"

h_grantee = "grantee"

// The variables h_ReadWriteFlag, h_DateFlag, h_Rekursive and
h_InheritDate
```

```

// build the right. If the right to write recursively to this
document

// inherited. The right expires on 01.01.9999

h_ReadWriteFlag = 2

h_DateFlag = 1

h_Rekursive = 1

h_InheritDate = "01.01.9999"

h_ReturnValue = api_function( "add_inherit_doc_rights", wf_Do-
kuID, h_granter, \

h_grantee, h_ReadWriteFlag ## h_DateFlag \

## h_Rekursive, h_InheritDate ## " 00\:00\:00" )

if (h_ReturnValue == 0)

{

call api_log_info("Rights were inherited.")

}

else

{

call api_log_error"Rights were not inherited. Error:h_ReturnVal-
ue")

}

SET_API_USER_MASTER = tmpMasterAPI

```

remove_inherit_doc_rights

Function for the deletion of inherited rights to a document.

Parameters	Description
doc_id	Document to which the rights shall be deleted.
user_name	User in whose name the right is to be deleted.
grantee	User whose rights are to be deleted.

Return value	Meaning
0	everything OK
456	granter is not allowed to delete the rights.

1.4.3. Folder

folder_create

Function to create a folder.

Attributes of the new folder must be written in the

```

api_doc_fields
api_doc_texts[1..4]
api_doc_status
api_doc_editor
api_doc_noundapi_doc_type_short
.

```

If synchronous creation is successful, `api_doc_id` contains the document ID of the new file.

Parameters	Description
<code>synchron</code>	Create the folder immediately (0: asynchron, 1: synchron) optional - default: 0

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-104	Document type is not specified in <code>api_doc_type_short</code>
-548	Error creating the folder; for details see the logviewer
-566	Error determining the temporary file name
-567	Error creating the JPL-file
-568	Error creating the document file

This example sets the global variables with the information for the creation of the folder. Afterwards, the function for the folder generation is called. If an error occurs, then the error number is returned.

Note

```
// -----
// Beispiel:
// Anlegen einer Akte
// -----

// Variablendeklaration
vars h_ReturnValue      // Fehlercode
vars h_Modus = "1"      // Erstellungsmodus: synchron

// setzen der Attributwerte für die neue Akte
api_doc_type_short  = "D405"
api_doc_texts[1]    = "Test: neue Akte"
api_doc_status      = "Freigabe"
api_doc_field[1]    = "42"
api_doc_field[2]    = "1000"
api_doc_field_60[1] = "Wert 1"
api_doc_field_60[2] = "Wert 2"

// synchrones Anlegen der Akte
h_ReturnValue = api_function ("folder_create", h_Modus)

// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    call api_log_info("Akte erfolgreich angelegt unter der ID \
                      :api_doc_id")
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
                      zurückgeliefert!")
}
```

1.4.4. Links**link_get_parents**

This function determines the parent documents. The IDs of the superordinate documents are stored in the global array `api_links[]`, the `link_creator` in `api_link_creators[]`, the time stamp in `api_link_tstamps[]`. The variable `api_single_info` contains the number of links found.

Note

`link_get_children` also writes into `api_links[]`.

Parameters	Description
<code>doc_id</code>	ID of the document, which rights shall be queried. If nothing is passed, then the ID of the context is used. Mandatory field - either as parameter or in the context as <code>api_doc_id</code>
<code>user_name</code>	User in whose name the link shall be determined. If nothing is specified, then the executive user is used. optional

Return value	Meaning
0	everything OK
> 0	Database error
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-3	undefined error, contact d.velop support
-101	neither document ID passed nor context set
-201	invalid user name specified
-301	invalid document ID specified or user has got no rights to it
-302	Error while determining the user rights
-303	User has got no access permission to document

This example queries the parent documents for "Y000050257" and returns them coded. If an error occurs, then the error number is returned.

Note

```
// -----
// Beispiel:
// Abfrage eines übergeordneten Dokumentes
// -----

// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_DocIdRef = "Y000050257" // Dokumenten-ID

// Funktionsaufruf
h_ReturnValue = api_function ("link_get_parents", h_DocIdRef, "")

// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    vars h_Count
    for h_Count = 1 while h_Count <= api_single_info step 1
    {
        // Ausgabe der Nummer, der DokID und des
        // Zeitstempels des übergeordneten Dokumentes
        call api_log_info("Parent :h_Count: ID
:api_links[h_Count], \
                        erstellt am
:api_link_tstamps[h_Count]")
    }
}
// Fehlerbehandlung
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
zurückgeliefert!")
}
```

link_get_children

This function determines the child documents. The IDs of the subordinate documents are stored in the global array `api_links[]`, the `link_creator` in `api_link_creators[]`, the time stamp in `api_link_tstamps[]`. The variable `api_single_info` contains the number of links found.

Note

`link_get_children` also writes into `api_links[]`.

Parameters	Description
<code>doc_id</code>	ID of the document, whose child documents are to be determined. If nothing is specified, then the ID from the context is used. Mandatory field - either as parameter or in the context as <code>api_doc_id</code>
<code>user_name</code>	User in whose name the link shall be determined. If nothing is specified, then the executive user is used. optional

Return value	Meaning
0	everything OK
> 0	Database error
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-3	unknown error, contact d.velop support
-101	neither document ID passed nor context set
-201	invalid user name specified
-301	invalid document ID specified or user has got no rights to it
-302	Error while determining the user rights
-303	User has got no access permission to document

This example queries the child documents for “Y000050501” and returns them coded.

If an error occurs, then the error number is returned.

Note

```
// Beispiel:
// Abfrage untergeordneter Dokumente
// -----

// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_DocIdRef = "Y000050501" // Dokumenten-ID

// Funktionsaufruf
h_ReturnValue = api_function("link_get_children", h_DocIdRef, "")

// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    vars h_Count
    for h_Count = 1 while h_Count <= api_single_info step 1
    {
        // Ausgabe der Kindnummer, der DokID des Kindes und
        // des Zeitstempels
        call api_log_info("Child :h_Count: ID :api_links[h_Count], \
                           erstellt am :api_link_tstamps[h_Count]")
    }
}
// Fehlerbehandlung
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
                       zurückgeliefert!")
}
```

link_documents

This function links two documents.

If a job is created, the job number is stored in `api_single_info`, if the `doc_id_parent` is not specified.

Parameters	Description
<code>doc_id_parent</code>	ID of the parent document. If nothing is specified, an async job for linking is created with the recognition attributes, which are written in the <code>api_doc_fields</code>. The <code>inherit_field_name_1..2 / inherit_field_value_1..2</code> and the inheriting of attributes must be done by the user himself. optional
<code>doc_id_child</code>	ID of the child document. If nothing is specified, then the Document ID from the context is used. Mandatory field - either as parameter or in the context as <code>api_doc_id</code>
<code>user_name</code>	User in whose name the link shall be created. If nothing is specified, then the executing user is used. optional
<code>folder_definition</code>	Create folder, if it does not exist. optional - default: 0

Parameters	Description
<code>use_folderplan</code>	If the <code>doc_id_parent</code> is not specified. 0: Create LIN002-JOB with recognition attributes in <code>api_doc_field</code> and <code>api_doc_type_short</code> and the inherited fields in <code>ernie_inherit_field_name_1..4</code> 1: Use dossier generation for the document optional – default: 0

Return value	Meaning
0	everything OK
> 0	Database error
-1	Panther syntax error, details see logviewer
-2	function was called incorrect
-3	unknown error, contact d.velop support
-101	neither document ID passed nor context set
-201	invalid user name specified
-219	Superordinate and subordinate document ID are identical.
-301	invalid document ID specified or user has got no rights to it
-302	Error while determining the user rights
-303	User has got no access permission to document
-312	User has no right to link/unlink the documents
	<code>api_error_var[.] = "right_level"</code>
	<code>api_error_value[.] = "<Rights level>"</code>
	<code>api_error_var[.] = "right_link"</code>
	<code>api_error_value[.] = "<Right to create folder links>"</code>
-520	The documents are already linked.
-521	The documents are already linked in the opposite direction.
<-9500	Error in customer-specific hook-function <code>hook_link</code>
	<code>api_error_var[.] = "hook_return_code"</code>
	<code>api_error_value[.] = return value of the hook</code>

Note

Available from d.3 version 5.5.1 hotfix 14, 6.0

The first example links the child document "Y000050256" with the parent document "Y00050501".

In the second example, the file "Y000050256" is created using the file creation.

Note

```
// -----
// Beispiel A:
// Direkte Verlinkung durch Angabe der übergeordneten doc_id
// -----

// Variablendeklaration
vars h_ReturnValue           // Fehlercode
vars h_DocIdRef = "Y000050257" // untergeordnete
vars h_DocIdRefParent = "Y000050501" // übergeordnete
Dokumenten-ID

// Funktionsaufruf:
// direkte Verlinkung durch Angabe des parent
h_ReturnValue = api_function ("link_documents",
h_DocIdRefParent, \
                                h_DocIdRef, \
                                "", \
                                0, \
                                "")

// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    call api_log_info("Verknüpfung erfolgreich erstellt!")
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
    zurückgeliefert!")
}
```

Note

```
// -----
// Beispiel B:
// Verlinkung durch Angabe von Erkennungsattributen
// und Erzeugen eines Verlinkung-Jobs
// -----

// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_DocIdRef = "Y000050256" // Dokumenten-ID

// Angabe von Erkennungs-Attributen
api_doc_type_short = "D405"
api_doc_field[1]   = "42"
api_doc_field[2]   = "1000"

// Funktionsaufruf:
// Verlinkung durch Job über Erkennungs-Attribute
h_ReturnValue = api_function
                ("link_documents", "", h_DocIdRef, "", 0, 0)

// Ergebnisaufruf und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    call api_log_info("Verknüpfungs-Job erstellt!")
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
                        zurückgeliefert!")
}
```

link_exists

This function checks, if two documents are linked to each other.

Parameters	Description
doc_id_parent	ID of the parent document.
doc_id_child	ID of the child document.
test_vice_versa	1: also check in opposite direction (default=0).

Return value	Meaning
0	Documents are not linked
1	Documents are linked
2	Documents are linked in the opposite direction.
>2	Database error
-1	Panther syntax error, details see logviewer
-2	function was called incorrect
-101	either document ID passed nor set in the context for superordinate or subordinate documents
-219	super- and subordinate document ID are identical

This example checks, if two documents are linked to each other.

Note

```
// -----
// Beispiel:
// Prüfen, ob zwei Dokumente verknüpft sind
// -----

// Variablendeklaration
vars h_ReturnValue           // Fehlercode
vars h_DokIDFather = "Y000050256" // übergeordnetes Dokument
vars h_DokIDChild  = "Y000050255" // untergeordnetes Dokument

// Funktionsaufruf
h_ReturnValue = api_function ("link_exists", h_DokIDFather, \
                               h_DokIDChild, 1 )

// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 1)
{
    call api_log_info("Dokumente sind verknüpft!")
}
else if (h_ReturnValue == 0)
{
    call api_log_info("Dokumente sind nicht verknüpft!")
}
else if (h_ReturnValue == 2)
{
    call api_log_info("Dokumente sind umgekehrt verknüpft!")
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
                       zurückgeliefert!")
}
```

link_delete

This function deletes a link between a parent and a child document.

Parameters	Description
doc_id_parent	ID of the parent document. Mandatory field - either as parameter or in the context as api_doc_id
doc_id_child	ID of the child document. If the doc_id is taken from the context because it is not specified as parameter, then an error occurs. Mandatory field - either as parameter or in the context as api_doc_id
user_name	User in whose name the link shall be determined. If nothing is specified, then the executive user is used. optional

Return value	Meaning
0	everything OK
> 0	Database error
-1	Panther syntax error, details see logviewer
-2	function was called incorrect

Return value	Meaning
-3	unknown error, contact d.velop support
-101	neither document ID passed nor set in the context for superordinate or subordinate documents
-201	invalid user name specified
-219	Superordinate and subordinate document ID are identical.
-301	invalid document ID specified or user has got no rights to it
-302	Error while determining the user rights
-303	User has got no access permission to document
-312	User has no right to link/unlink the documents <code>api_error_var[..] = "right_level"</code> <code>api_error_value[..] = "<Rights level>"</code> <code>api_error_var[..] = "right_link"</code> <code>api_error_value[..] = "<Right to create folder links>"</code>
-522	The documents are not linked this way
<-9500	Error in customer-specific hook-function <code>hook_unlink</code> <code>api_error_var[..] = "hook_return_code"</code> <code>api_error_value[..] = return value of the hook</code>

Note

The function `link_delete` returns the error message (-219), if both

`doc_ids` (this means `doc_id_parent` and `doc_id_child`) are not passed.

In this case, the respective `api_doc_id` is taken.

Then `doc_id_parent` and `doc_id_child` are identical.

This example deletes the link between the documents Y000050256 and Y00050501. If an error occurs, then the error number is returned. If an error occurs, then the error number is returned.

Note

```

/ -----
// Beispiel:
// Lösen einer Verknüpfung
// -----
// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_DocIdRef      = "Y000050256"    // Untergeordnete
Dokumenten-ID
vars h_DocIdRefParent = "Y000050501" // Übergeordnete Dokumenten-
ID
// Funktionsaufruf
h_ReturnValue = api_function ("link_delete", h_DocIdRefParent, \
                             h_DocIdRef, "")
// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    call api_log_info("Verknüpfung erfolgreich entfernt!")
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
    zurückgeliefert!")
}

```

1.4.5. Directories**directory_create**

Creates a directory on the hard disk.

Parameters	Description
directory	Directory, mandatory field

Return value	Meaning
0	Directory created
-1	Panther syntax error, details see logviewer
-2	function was called incorrect
-160	Directory is not specified
-581	The specified directory cannot be created!

Note

available from d.3 version 6.0.1 hotfix 1

This example creates the directory D:\api_test\dir1.

Note

```
// -----
// Beispiel:
// Anlegen eines Verzeichnisses
// -----
// Variablendeklaration
vars h_ReturnValue           // Fehlercode
vars h_PathRef = "d:\\api_test\\dir1" // Verzeichnis

// Funktionsaufruf:
// Ordner anlegen - ":" und "\" müssen dabei immer maskiert sein
h_ReturnValue = api_function ("directory_create", h_PathRef)

// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    call api_log_info("Ordner erfolgreich angelegt!")
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
        zurückgeliefert!")
}
```

directory_delete

This function deletes a directory from the hard disk.

Parameters	Description
directory	Directory, mandatory field

Return value	Meaning
0	Directory deleted
-1	Panther syntax error, details see logviewer
-2	function was called incorrect
-3	undefined error, contact d.velop support
-160	Directory is not specified
-580	The directory could not be deleted.

This example deletes the directory D:\api_test\dir1. If an error occurs, then the error number is returned.

Note

```
// -----
// Beispiel:
// Löschen eines Verzeichnisses
// -----

// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_PathRef = "d:\\api_test\\dir1" // Verzeichnis

// Funktionsaufruf
h_ReturnValue = api_function ("directory_delete", h_PathRef)

// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    call api_log_info("Ordner erfolgreich gelöscht!")
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
        zurückgeliefert!")
}
```

directory_exist

This function determines, if a directory exists.

Parameters	Description
directory	Directory, mandatory field

Return value	Meaning
1	Directory exists
0	Directory does not exist
-1	Panther syntax error, details see logviewer
-2	function was called incorrect
-3	undefined error, contact d.velop support
-160	Directory is not specified

This example checks, if the directory `D:\api_test\dir1` exists and returns the result. If an error occurs, then the error number is returned.

Note

```
// -----
// Beispiel:
// Prüfen, ob Verzeichnis existiert
// -----

// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_PathRef = "d:\\api_test\\dir1" // Verzeichnis

// Funktionsaufruf
h_ReturnValue = api_function ("directory_exist", h_PathRef)

// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 1)
{
    call api_log_info("Ordner existiert!")
}
else if (h_ReturnValue == 0)
{
    call api_log_info("Ordner existiert nicht!")
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
        zurückgeliefert!")
}
```

directory_list

This function lists the content of a directory (max. 100 entries). The determined entries will be written into the `api_list`, the count will be written in `api_single_info`.

Parameters	Description
directory	Directory Mandatory field
pattern	Search Pattern optional - default = *, so everything
mode	Mode (as string): 1 only files (default) 2 only directories 3. both optional - default = 1

Return value	Meaning
0	no errors
-1	Panther syntax error, details see logviewer
-2	function was called incorrect
-3	undefined error, contact d.velop support
-160	Directory is not specified
-580	Error listing the directory content

This example lists the sub-directories of the directory `D:\api_test`. If an error occurs, then the error number is returned.

Note

```
// -----
// Beispiel:
// Abfrage von Unterverzeichnissen
// -----

// Variablendeklaration
vars h_ReturnValue           // Fehlercode
vars h_PathRef = "d:\\api_test" // Verzeichnis
varsh_PrintingMode = "2"      // Modus: Nur Verzeichnisse
ausgeben
vars h_Count                 // Zählervariable für
Schleife

// Funktionsaufruf
// Liste aller direkten Unterordner des Ordners
// "d:\api_test" ermitteln
h_ReturnValue = api_function ("directory_list", h_PathRef, "", \
                             h_PrintingMode)

// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    for h_Count = 1 while h_Count <= api_single_info step 1
    {
        call api_log_info("Element :h_Count:
:api_list[h_Count]")
    }
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
zurückgeliefert!")
}
```

directory_count_files

This function determines the number of entries in a directory based on a search mask. Subfolders can be included, too

Parameters	Description
directory	Directory Mandatory field
pattern	Search Pattern optional – default = *, so everything
mode	Mode (as string): 0 only directories 1 also sub-directories

Return value	Meaning
0	no errors
-1	Panther syntax error, details see logviewer
-2	function was called incorrect
-3	undefined error, contact d.velop support
-160	Directory is not specified
-583	Error counting the files

Number of files in a folder.

Warning

The path to the directory ":" and "\" must be masked with an additional "\\".

Note

```
// -----
// Beispiel:
// Anzahl der Dateien in einem Verzeichnis nach Suchmuster
// bestimmen
// -----

// Variablendeklaration
vars h_ReturnValue                                // Fehlercode
vars h_Path = "c:\\temp"                          // Suchverzeichnis
varsh_Pattern = "*.tif"                          // Suchmuster: alle "tif"-
Dateien
vars h_Mode = "0"                                // Modus: keine
Unterverzeichnisse

// Funktionsaufruf
// Anzahl aller Dateien mit Endung "tif" im Verzeichnis
// "c:\\temp" ermitteln
h_ReturnValue = api_function ("directory_count_files", h_Path, \
                                h_Pattern, h_Mode)

// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue >= 0)
{
    call api_log_info("Es wurden :h_ReturnValue Dateien
gefunden.")
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
zurückgeliefert!")
}
```

1.4.6. Status

document_block

This function blocks or releases a document.

Parameters	Description
block	0: Document unblock 1: block document (default) optional
user_name	User, who (un)blocks. If nothing is specified, then the user from the context is used. optional
doc_id	Document to be transferred. If nothing is specified, then the ID from the context is used. Mandatory field - either as parameter or in the context as <code>api_dok_id</code>

Return value	Meaning
0	everything OK
> 0	Database error
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-3	undefined error, contact d.velop support
-101	neither document ID passed nor context set
-201	invalid user name specified
-301	invalid document ID specified or user has got no rights to it
-302	Error while determining the user rights
-303	User has got no access permission to document
-306	User does not have the permission to block or unblock this document.
-506	There is no released version. <code>api_error_var[...] = "doc_permission"</code> <code>api_error_value[...] = "<permission>"</code>
-507	Version in release is already blocked <code>api_error_var[...] = "doc_permission"</code> <code>api_error_value[...] = "<permission>"</code>
-508	Version in release is not blocked <code>api_error_var[...] = "doc_permission"</code> <code>api_error_value[...] = "<permission>"</code>
< -9500	Error in customer-specific hook-function <code>hook_release_...</code> <code>api_error_var[...] = "hook_return_code"</code> <code>api_error_value[...] = return value of the hook</code>

The first example block the document with the ID `Y000050257` and returns an error if necessary.

The second example unblock the document with the ID `Y000050257` and returns an error if necessary.

Note

```
// -----  
// Beispiel A:  
// Sperren eines Dokumentes  
// -----  
  
// Variablendeklaration  
vars h_ReturnValue           // Fehlercode  
vars h_DocIdRef = "Y000050257" // Dokumenten-ID  
varsh_BlockMode = 1          // Modus: Dokument sperren  
  
// Funktionsaufruf:  
// Dokument sperren  
h_ReturnValue = api_function ("document_block", h_BlockMode,  
    "", \                                h_DocIdRef)  
  
// Ergebnisausgabe und Fehlerbehandlung  
if (h_ReturnValue == 0)  
{  
    call api_log_info("Dokument erfolgreich gesperrt!")  
}  
else  
{  
    call api_log_error("Fehlernummer :h_ReturnValue wurde  
        zurückgeliefert!")  
}
```

Note

```
// -----
// Beispiel B:
// Entsperren eines Dokumentes
// -----

// Variablendeklaration
vars h_ReturnValue           // Fehlercode
vars h_DocIdRef = "Y000050257" // Dokumenten-ID
varsh_BlockMode = 0           // Modus: Dokument entsperren

// Funktionsaufruf:
// Dokument entsperren
h_ReturnValue = api_function ("document_block", h_BlockMode,
    "", \
                                h_DocIdRef )

// Ergebnisausgabe und Variablendeklaration
if (h_ReturnValue == 0)
{
    call api_log_info("Dokument erfolgreich entsperrt!")
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
        zurückgeliefert!")
}
```

document_verify

This function transfers a document from the status "Verification, unverified" into the status "Verification, verified".

Parameters	Description
user_name	User who shall execute the verification. If nothing is specified, then the user from the context is used. optional
doc_id	Document to be transferred. If nothing is specified, then the ID from the context is used. Mandatory field - either as parameter or in the context as <code>api_dok_id</code>

Return value	Meaning
0	everything OK
> 0	Database error
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-3	undefined error, contact d.velop support
-101	neither document ID passed nor context set
-201	invalid user name specified
-301	invalid document ID specified or user has got no rights to it
-302	Error while determining the user rights
-303	User has got no access permission to document

Return value	Meaning
-307	User does not have the permission to verify this document. api_error_var[...] = "doc_permission" api_error_value[...] = "<permission>"
-509	Version in status verification is already verified api_error_var[...] = "doc_permission" api_error_value[...] = "<permission>"
-510	There is no version in the status archive api_error_var[...] = "doc_permission" api_error_value[...] = "<permission>"
< -9500	Error in customer-specific hook-function hook_release_... api_error_var[...] = "hook_return_code" api_error_value[...] = return value of the hook

This example verifies the document with the ID Y000050257. If an error occurs, then the error number is returned.

Note

```
// -----
// Beispiel:
// Prüfung eines Dokumentes
// -----
// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_DocIdRef = "Y000050257" // Dokumenten-ID

// Funktionsaufruf:
// Dokument prüfen
h_ReturnValue = api_function( "document_verify", "", h_DocIdRef )

// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    call api_log_info("Dokument befindet sich nun im Status \
        \\"Prüfung, geprüft!\\"")
        // " im Augabetext muss mit \ maskiert werden
}
else
{
    if (h_ReturnValue == -509)
    {
        call api_log_info("Dokument befindet sich bereits im
Status \
        \\"Prüfung, geprüft!\\"")
        // " im Augabetext muss mit \ maskiert werden
    }
    else if (h_ReturnValue == -510)
    {
        call api_log_info("Dokument befindet sich nicht im
Status \
        \\"Prüfung!\\"")
        // " im Augabetext muss mit \ maskiert werden
    }
    else
    {
        call api_log_error("Fehlernummer \
            :h_ReturnValue wurde
zurückgeliefert!")
    }
}
}
```

document_transfer**Note**

New in version 7.1:

The function may only transfer signed documents from the status **Release** to the status **Archive**.

This function executes a status transfer. The current version is used unless an archived version exists and an `archiv_index` is specified.

Parameters	Description
<code>destination</code>	Target status ("Bearbeitung" (Processing), "Prüfung"(Verification), "Freigabe" (Release), "Archiv"(Archive)) Mandatory field
<code>user_group_group</code>	User-/group name is required for <code>destination = processing => user or group</code> <code>destination = verification => group</code> Mandatory field, if the parameter <code>destination</code> is specified as "Processing" Note changed for version 7.0: The parameter <code>destination</code> is no longer mandatory for the status "Verification".
<code>change_remark</code>	Editing comment: must be specified, if a document is transferred from processing into verification and the document was changed compared to an earlier version in the status processing or archive. If a remark is specified for a status transfer from B -> P, then the document is filed with a new version number. If you want to maintain the old version number, then no comment must be passed. optional
<code>asynchronous</code>	Specifies, if the transfer should be executed immediately or asynchronously via an asynchronous job for the d.3 async (0:immediately (default); 1: later through d.3 async). optional - default: 0 0
<code>archive_index</code>	Index for the version in the status archive to be restored. optional
<code>user_name</code>	User who shall execute the transfer. If nothing is specified, then the user from the context is used. optional
<code>doc_id</code>	Document to be transferred. If nothing is specified, then the ID from the context is used. Mandatory field - either as parameter or in the context as <code>api_dok_id</code>

Return value	Meaning
0	everything OK
> 0	Database error
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-3	undefined error, contact d.velop support
-101	either document ID passed nor context set
-112	User or group name is not specified <pre>api_error_var[..] = "doc_permission"</pre> <pre>api_error_value[..] = "<permission>"</pre> <pre>api_error_var[..] = "state"</pre> <pre>api_error_value[..] = "<source status>"</pre> <pre>api_error_var[..] = "editor"</pre> <pre>api_error_value[..] = "<source editor>"</pre>

Return value	Meaning
-113	Group name is not specified <pre>api_error_var[..] = "doc_permission" api_error_value[..] = "<permission>" api_error_var[..] = "state" api_error_value[..] = "<source status>" api_error_var[..] = "editor" api_error_value[..] = "<source editor>"</pre> Note new in d.3 version 7.0: Error -113 no longer occurs in status verification.
-114	Target status is not specified
-115	Archive index is not specified <pre>api_error_var[..] = "doc_permission" api_error_value[..] = "<permission>" api_error_var[..] = "state" api_error_value[..] = "<source status>" api_error_var[..] = "editor" api_error_value[..] = "<source editor>"</pre>
-116	Editing comment is not specified <pre>api_error_var[..] = "doc_permission" api_error_value[..] = "<permission>" api_error_var[..] = "state" api_error_value[..] = "<source status>" api_error_var[..] = "editor" api_error_value[..] = "<source editor>"</pre>
-201	invalid user name specified
-213	invalid group name specified <pre>api_error_var[..] = "doc_permission" api_error_value[..] = "<permission>" api_error_var[..] = "state" api_error_value[..] = "<source status>" api_error_var[..] = "editor" api_error_value[..] = "<source editor>"</pre>
-214	invalid target status specified <pre>api_error_var[..] = "doc_permission" api_error_value[..] = "<permission>" api_error_var[..] = "state" api_error_value[..] = "<source status>" api_error_var[..] = "editor" api_error_value[..] = "<source editor>"</pre>

Return value	Meaning
-301	invalid document ID specified or user has got no rights to it
-302	Error while determining the user rights
-303	User has got no access permission to document
-308	User does not have the permission to release this document. api_error_var[...] = "doc_permission" api_error_value[...] = "<permission>"
-309	User does not have the permission to transfer this document. api_error_var[...] = "doc_permission" api_error_value[...] = "<permission>"
-310	User does not have the permission to see old versions in the status archive. api_error_var[...] = "doc_permission" api_error_value[...] = "<permission>" api_error_var[...] = "state" api_error_value[...] = "<source status>" api_error_var[...] = "editor" api_error_value[...] = "<source editor>"
-511	This transfer is not allowed. api_error_var[...] = "doc_permission" api_error_value[...] = "<permission>" api_error_var[...] = "state" api_error_value[...] = "<source status>" api_error_var[...] = "editor" api_error_value[...] = "<source editor>"
-512	There is a newer version. To overwrite it, it must be transferred in the processing of the user <:user_group>. api_error_var[...] = "doc_permission" api_error_value[...] = "<permission>"
-513	Error <:job_error> while creating the job
-514	Error while comparing the document file of old and new version api_error_var[...] = "doc_permission" api_error_value[...] = "<permission>" api_error_var[...] = "state" api_error_value[...] = "<source status>" api_error_var[...] = "editor" api_error_value[...] = "<source editor>"
< -9500	Error in customer-specific hook-function hook_transfer... api_error_var[...] = "hook_return_code" api_error_value[...] = return value of the hook

The first example transfers a document with the specified document ID into the "Processing" of the user "dvelop".

The second example (B) transfers the specified document into the "Verification" of the group "Test".

The third example (C) releases the specified document.

Note

```

/ -----
// Beispiel A:
// Ein Dokument in Bearbeitung holen
// -----

// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_DocIdRef = "Y000050257" // Dokumenten-ID des
betreffenen                 // Dokumentes

vars h_StatusRef = "Bearbeitung" // Ziel-Status
vars h_GroupRef = "dvelop"       // Benutzername
vars h_asynchronus = 0           // Sofortiger Statustransfer

// Funktionsaufruf
h_ReturnValue = api_function ("document_transfer", h_StatusRef, \
                                h_GroupRef, \
                                "", \
                                h_asynchronus, \
                                "", \
                                "", \
                                h_DocIdRef)

// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    call api_log_info("Dokument erfolgreich in den gewünschten
Status \
                        verschoben!")
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
zurückgeliefert!")
}

```

Note

```
// -----
// Beispiel B:
// Ein Dokument in die Prüfung verschieben
// -----

// Variablendeklaration
vars h_ReturnValue           // Fehlercode
vars h_DocIdRef = "Y000050257" // Dokumenten-ID
vars h_StatusRef = "Prüfung"   // Ziel-Status
vars h_GroupRef = "test"       // Gruppenname
vars h_asynchronus = 0         // Sofortiger Statustransfer

// Funktionsaufruf
h_ReturnValue = api_function ("document_transfer", h_StatusRef, \
                                h_GroupRef, \
                                "", \
                                h_asynchronus, \
                                "", \
                                "", \
                                h_DocIdRef)

// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    call api_log_info("Dokument erfolgreich in den gewünschten
Status \
                                verschoben!")
}
else
{
    call api_log_error("Fehlernummer \
:h_ReturnValue wurde zurückgeliefert!")
}
```

Note

```
// -----
// Beispiel C:
// Ein Dokument freigeben
// -----

// Variablendeklaration
vars h_ReturnValue           // Fehlercode
vars h_DocIdRef = "Y000050257" // Dokumenten-ID
vars h_StatusRef = "Freigabe"  // Ziel-Status
vars h_asynchronus = 1        // Asynchroner Statustransfer

// Funktionsaufruf
// das Dokument sollte vor dem Freigeben bereits geprüft sein
h_ReturnValue = api_function ("document_transfer", h_StatusRef, \
                                "", \
                                "", \
                                h_asynchronus, \
                                "", \
                                "", \
                                h_DocIdRef)

// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    call api_log_info("Dokument erfolgreich in den gewünschten
Status \
                                verschoben!")
}
else
{
    call api_log_error("Fehlernummer \
:h_ReturnValue wurde zurückgeliefert!")
}
```

1.4.7. Notes**note_add_file**

This function append the content of a file to the note file of the document.

Parameters	Description
file_name	Name of the file, whose content should be appended to the note. Mandatory field
doc_id	ID of the document, whose note file is to be changed. Ff nothing is specified, the ID of the context is used. Mandatory field - either as parameter or in the context: api_doc_id
user_name	User for whom the note shall be written; if nothing is specified, then the executive user will be used optional

Return value	Meaning
0	everything OK

Return value	Meaning
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-3	undefined error, contact d.velop support
-101	neither document ID passed nor context set
-150	File name is not specified
-201	invalid user name specified
-250	File not found
-301	invalid document ID specified or user has got no rights to it
-302	Error while determining the user rights
-303	User has got no access permission to document
-401	The file could not be encrypted/decrypted
-402	Encrypted / decrypted file could not be renamed properly.
-403	Encrypted / decrypted temporary file could not be deleted.
< -9500	Error in customer-specific hook-function hook_transfer...
	api_error_var[..] = "hook_return_code"
	api_error_value[..] = return value of the hook

This example appends the content of the file C:\temp\note.txt to the note file of the document with the ID Y000050257. If an error occurs, then the error number is returned.

Note

```
// -----
// Beispiel:
// Notizdatei an Dokument anhängen
// -----

// Variablendeklaration
vars h_ReturnValue                                // Fehlercode
vars h_DocIdRef = "Y000050257"                    // Dokumenten-ID
vars h_NoteFilePath = "c:\\temp\\note.txt"         // Datei mit Inhalt
für                                                    // Notizdatei

// Funktionsaufruf
h_ReturnValue = api_function ("note_add_file", h_NoteFilePath, \
                               h_DocIdRef, "")

// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    call api_log_info("Die Notiz wurde erfolgreich angehängt!")
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
                        zurückgeliefert!")
}
```

note_add_string

This function append the content of a file to the note file of the document.

Parameters	Description
line	Line to be appended to the note. Mandatory field
doc_id	ID of the document, whose note file is to be changed. If nothing is specified, the ID of the context is used. Mandatory field - either as parameter or in the context: <code>api_doc_id</code>
user_name	User for whom the note shall be written; if nothing is specified, the executing user is used (optional)

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-3	undefined error, contact d.velop support
-101	neither document ID passed nor context set
-111	no line for insert specified
-201	invalid user name specified
-301	invalid document ID specified or user has got no rights to it
-302	Error while determining the user rights
-303	User has got no access permission to document
-401	The file could not be encrypted/decrypted
-402	Encrypted / decrypted file could not be renamed properly.
-403	Encrypted / decrypted temporary file could not be deleted.
< -9500	Error in customer-specific hook-function <code>hook_transfer_...</code> <code>api_error_var[..] = "hook_return_code"</code> <code>api_error_value[..] = return value of the hook</code>

In this example, the note file for the document with the ID `Y000050257` is appended with the line specified. If an error occurs, then the error number is returned.

Note

```
// -----
// Beispiel:
// Zeile an eine Notizdatei anhängen
// -----

// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_DocIdRef = "Y000050257" // Dokumenten-ID
vars h_NoteLine = "Diese Zeile wurde durch die Funktion \
                    note_add_string angehängt"

// Funktionsaufruf
h_ReturnValue = api_function ("note_add_string", h_NoteLine, \
                              h_DocIdRef, "")

// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    call api_log_info("Die Notiz wurde erfolgreich angehängt!")
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
                      zurückgeliefert!")
}
```

1.4.8. Resubmission**hold_file_send**

This function sends a document to the mailbox of a d.3 user. All users / groups listed under recipient are written into `api splitted_string`.

Parameters	Description
recipient	Recipient of the resubmission (d.3 user / group); several recipients must be separated with a semi-colon (;). Mandatory field
notice	Subject text optional
doc_id	ID of the document to be sent. If nothing is specified, the ID of the context is used. Mandatory field - either as parameter or in the context: <code>api_dok_id</code>
date_acknowledge	Date up to which entries are to be acknowledged. optional
date_remember	Date, on which a reminder is to be sent. optional
expand_groups	Resolve a group and notify users individually. optional - default: 0 0

Parameters	Description
ignore_checkout	Send even if recipient has checked out. optional - default: 0 0 0 = do not send if recipient is checked out. 1 = send, even if recipient is checked out.
date_activate	Date on which the resubmission is to be sent. optional
type	Type of the resubmission (W=workflow) optional
sender	Sender; if nothing is specified, then the executing user is used. optional
chain_id	ID of the holdfile-chain optional
remove_immediately	1 = Having sent the resubmission the loesch_flags_sender is set immediately so that no entry appears under the sent resubmissions.
inherit_class_rule	1 = Expiry of the permission after timestamp (in this case inherit_class_tstamp must be set); 2 = Expires after acknowledgement in the mailbox
inherit_class_tstamp	Time of expiration for the delegated permission as a time stamp of the form DD.MM.YYYY - hh:mm:ss
inherit_class_right	1 = Reading rights 2 = Writing rights (also implies reading rights)
check_write_access	Checks, if the recipient has write access for the document.
Note New parameter in d.3 Version 7.0.	

Return value	Meaning
0	everything OK
> 0	Database error
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-3	undefined error, contact d.velop support
-101	neither document ID passed nor context set
-110	Recipient is not specified
-201	invalid user name specified api_error_var[..] = "sender" api_error_value[..] = "<Sender name> - invalid"
-207	invalid recipient specified api_error_var[..] = "recipient" api_error_value[..] = "<recipient name> invalid"

Return value	Meaning
-208	Invalid date specified <code>api_error_var[..] = "date_activate_holdfile"</code> <code>api_error_value[..] = Activation date</code> <code>api_error_var[..] = "date_acknowledge_until"</code> <code>api_error_value[..] = acknowledge until date</code> <code>api_error_var[..] = "date_remember_at"</code> <code>api_error_value[..] = Reminder date</code>
-209	invalid chain ID specified
-212	invalid Doc ID specified.
-244	invalid rule or right specified
-301	invalid document ID specified or user has got no rights to it <code>api_error_var[..] = "recipient" / "sender"</code> <code>api_error_value[..] = "<recipient-/sender name> - no Access right ?"</code>
-302	Error while determining the user rights
-303	User has got no access permission to document <code>api_error_var[..] = "recipient" / "sender"</code> <code>api_error_value[..] = "<recipient-/sender name> - has no access right for document"</code>
-315	no group member has got access rights to the document to be sent
-316	Recipient hasn't got access rights but sender can grant rights. Note new return value in d.3 version 7.0
-317	Sender hasn't got write access. The right cannot be granted to the recipient. Note new return value in d.3 version 7.0
-318	Sender hasn't got write access, but the right can be inherited.
-450	User hasn't got the right for inheritance
-451	User hasn't got the rights he wants to inherit.
-452	The recipient already has got the permissions to be inherited.
-453	There are no document classes or authorization profiles assigned to the user.
-454	The recipient must have a right to the document type to be inherited.
-455	The temporary created document class was already assigned to the user.
-503	The recipient is checked out. <code>api_error_var[..] = "recipient"</code> <code>api_error_value[..] = "<recipient name> checked out"</code> <code>api_error_var[..] = "checkout_text"</code> <code>api_error_value[..] = Checkout text</code>

In the first example, the document with the ID "Y000050257" is sent to the mailboxes of the user "dvelop" and to the individual users of the group "test" – even if they are checked out.

The subject line is specified as "Test von `hold_file_send`". No remind or acknowledge date is specified. Since no sender is specified, the executive user is used.

In the second example, the user "dvelop" forwards the document with the ID "Y000050527" to the user/group "test". This will only succeed, if the user "dvelop" is the last user in the holdfile chain with

the ID "9" and the chain with the ID "9" refers to the document with the ID "Y000050257". For a call in this form, only one user or user group should be specified as a recipient. Otherwise, the forwarding to another recipient would always fail.

Note

```
// -----
// Beispiel A:
// Verschieben eines Dokumentes in den Postkorb
// -----
// Variablendeklaration
vars h_ReturnValue                // Fehlercode
vars h_RecipientUserName = "test;dvelop" // Die beiden
Empfänger                          // (-gruppen)

vars h_Subject = "Test von hold_file_send" // Betreff
vars h_DocIdRef = "Y000050257"           // Dokumenten-ID
vars h_ExpandGroups = 1                  // Empfänger einzeln
                                          // benachrichtigen

vars h_IgnoreCheckout = 1               // Auch bei
Abwesenheit                            // des Empfängers
                                          // versenden

// Funktionsaufruf
// verschickt eine Wiedervorlage an den Benutzer "dvelop"
// und an alle Benutzer der Gruppe "test"
h_ReturnValue = api_function ("hold_file_send",
h_RecipientUserName, \
                        h_Subject, h_DocIdRef, "", "",
                        h_ExpandGroups, \
                        h_IgnoreCheckout, "", "", "", "")

// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    call api_log_info ("Wiedervorlage wurde versendet!")
}
else
{
    call api_log_error ("Fehlernummer :h_ReturnValue wurde
zurückgeliefert!")
}
```

Note

```
// -----
// Beispiel B:
// Weiterleitung
// -----
// Variablendeklaration
vars h_ReturnValue // Fehlercode
vars h_RecipientUserName = "test" // Benutzername des
// Empfängers

vars h_Subject = "Test der Wiedervorlage-Kette" // Betreff
vars h_DocIdRef = "Y000050257" // Dokumenten-ID
vars h_ExpandGroups = 1 // Empfänger
// einzeln
// benachrichtigen

vars h_IgnoreCheckout = 1 // Auch bei
Abwesenheit

// des
// Empfängers versenden
vars h_SenderUserName = "dvelop" // Benutzername des
Senders
vars h_ChainID = "9" // ID der Wiedervorlage
// -Kette

// Funktionsaufruf
h_ReturnValue = api_function ("hold_file_send",
h_RecipientUserName, \
h_Subject, h_DocIdRef, "", "",
h_ExpandGroups, \
h_IgnoreCheckout, "", "", \
h_SenderUserName, h_ChainID)

// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
call api_log_info ("Wiedervorlage wurde versendet!")
}
else
{
call api_log_error ("Fehlernummer :h_ReturnValue wurde
zurückgeliefert!")
}
}
```

Note

hold_file_send should not be called in an entry-hook, if still e-mails will be sent.

hold_file_find

This function determines the users, in whose mailbox the specified document is located.

The user names are stored in api_users.

The number of users found are stored in api_single_info.

Parameters	Description
<code>recipient</code>	Recipient of a resubmission optional
<code>doc_id</code>	ID of the document to be searched. If nothing is specified, the ID of the context is used. Mandatory field - either as parameter or in the context: <code>api_dok_id</code>
<code>user_name</code>	User for whom shall be searched; if nothing is specified, then the executive user will be used optional

Return value	Meaning
0	everything OK
> 0	Database error
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-3	undefined error, contact d.velop support
-101	neither document ID passed nor context set
-201	invalid user name specified
-301	invalid document ID specified or user has got no rights to it
-302	Error while determining the user rights
-303	User has got no access permission to document

In the first example, all mailboxes or their users are returned which contain the document with the ID Y000050257. If an error occurs, then the error number is returned.

In the second example, it is checked, if a document with the ID Y000050257 is located in the mailbox of the user "dvelop" and the result is returned. If an error occurs, then the error number is returned.

Note

```
// -----  
// Beispiel A:  
// Abfrage von Postkörben  
// -----  
  
// Variablendeklaration  
vars h_ReturnValue           // Fehlercode  
vars h_DocIdRef = "Y000050257" // Dokumenten-ID  
vars h_Count                 // Zähler für die Schleife  
  
// Funktionsaufruf:  
// es werden alle Postkörbe ausgegeben, in denen das Dokument  
// liegt  
h_ReturnValue = api_function ("hold_file_find", "", h_DocIdRef,  
"") )  
  
// Ergebnisausgabe und Fehlerbehandlung  
if (h_ReturnValue == 0)  
{  
    call api_log_info("Dokument liegt in :api_single_info  
Postkörben!")  
  
    for h_Count = 1 while h_Count <= api_single_info step 1  
    {  
        call api_log_info("Postkorb :h_Count:  
:api_users[h_Count]")  
    }  
}  
else  
{  
    call api_log_error("Fehlernummer :h_ReturnValue wurde  
zurückgeliefert!")  
}
```

Note

```
// -----
// Beispiel B:
// Prüfung ob Datei in bestimmtem Postkorb
// -----

// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_DocIdRef = "D000000153" // Dokumenten-ID
vars h_UserName = "dvelop"    // Benutzername

// Funktionsaufruf: Es wird geschaut, ob das Dokument
// im Postkorb des Benutzers "dvelop" liegt
h_ReturnValue = api_function ("hold_file_find", h_UserName, \
                              h_DocIdRef, "")

// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    if (api_single_info > 0)
    {
        call api_log_info("Das Dokument :h_DocIdRef liegt im \
                           Postkorb des Benutzers dvelop")
    }
    else
    {
        call api_log_info("Das Dokument :h_DocIdRef befindet
sich \
                           nicht im Postkorb des Benutzers
dvelop")
    }
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
                       zurückgeliefert!")
}
}
```

hold_file_delete

This function acknowledges received resubmissions and deletes sent ones.

Note

The parameter `chain_id` required for this function `hold_file_delete` can be determined with the function `hold_file_get_document_info`.

Parameters	Description
<code>chain_id</code>	ID of the resubmission chain, that belongs to the entry Mandatory field
<code>sent_received</code>	Process sent (1) or received (2) resubmission. (optional – default = 2)

Parameters	Description
workflow_only	0/1: acknowledge workflow-resubmission (only for sent_received = 2). optional
recipient	Recipient (only for sent_received = 1 or recipient is a group). optional
doc_id	ID of the document to be deleted/acknowledged. If nothing is specified, the ID of the context is used. optional
user_name	User, in whose name it should be deleted/acknowledged. If nothing is specified, then the executive user will be used. optional

Return value	Meaning
0	everything OK
> 0	Database error
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-3	undefined error, contact d.velop support
-101	neither document ID passed nor context set
-110	no recipient specified
-124	no chain ID specified
-528	Error while deleting from the list of the sender
-529	Error while deleting from the list of the recipient

Note

available from d.3 version 5.5.1 hotfix 11, 6.0

In this example, the reception of a document "Y000050257" is acknowledged in the resubmissionchain with the ID "8". If an error occurs, then the error number is returned.

Note

```
// -----
// Beispiel:
// Quittierung eines Dokumentes
// -----
// Variablendeklaration
vars h_ReturnValue           // Fehlercode
vars h_ChainID = "8"         // ID der
Wiedervorlagenkette
vars h_DocID = "D000000153"  // Dokumenten-ID
vars h_RecipientUserName = "dvelop" // Benutzername des
Empfängers
// Empfangene Wiedervorlage löschen. Unter h_UserName sollte
// der Benutzer stehen, der die Wiedervorlage erhalten hat,
// wenn dieser Benutzer nicht der ausführende Benutzer ist
h_ReturnValue = api_function ("hold_file_delete", h_ChainID, \
                             2 ,0 , "" ,h_DocID,
h_RecipientUserName)
// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    call api_log_info("Wiedervorlage erfolgreich quittiert.")
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
zurückgeliefert!")
}
```

hold_file_get_document_info

This function retrieves information on a resubmission. The values will be written into the global variables:

Global variables	Description
api_hold_file_date	Date of the resubmission
api_hold_file_recipient	Recipient
api_hold_file_notice	Subject text
api_hold_file_informed	Recipient notified (0/1)
api_hold_file_tstamp_sent	Time stamp "sent"
api_hold_file_tstamp_ack	Time stamp "acknowledged"
api_hold_file_date_ack_until	Date "acknowledge until"
api_hold_file_sender	Sender
api_hold_file_date_remember	Date "remind on"
api_hold_file_deleted_sender	Deleted by sender (0/1)
api_hold_file_chain_id	Chain ID
api_hold_file_type	Type (W=Workflow)

Parameters	Description
sender	Sender Mandatory field
recipient	Recipient Mandatory field

Parameters	Description
<code>doc_id</code>	ID of the document. If nothing is specified, the ID of the context is used. Mandatory field - either as parameter or in the context as <code>api_doc_id</code>
<code>user_name</code>	User in whose name the information is to queried. If nothing is specified, then the executive user is used. optional

Return value	Meaning
0	everything OK
> 0	Database error
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-3	undefined error, contact d.velop support
-101	neither document ID passed nor context set
-110	no recipient specified
-125	no sender specified
-201	invalid user name specified
-301	invalid document ID specified or user has got no rights to it
-302	Error while determining the user rights
-303	User has got no access permission to document
-530	no resubmission with this sender and recipient

Note

The variable `api_hold_file_chain_id` was previously only 5 digits long. If this is now changed to 6 digits in a system (database) then the return values cannot be interpreted since the ID is then wrong.

This can be resolved by installing Hotfix 15 or higher for d.3 version 6.2.1 or Hotfix 4 or higher for d.3 version 6.3.0, if for example a consecutive `hold_file_delete` encounters errors. Thus, a hint could be included in `hold_file_delete`.

In this example, the information on the resubmission of the document with the specified document ID is written to the global variables and returned. If an error occurs, then the error number is returned.

Note

```
// -----
// Beispiel:
// Abfrage von Wiedervorlagedaten
// -----

// Variablendeklaration
vars h_ReturnValue                                // Fehlercode
vars h_SenderUserName = "dvelop"                  // Name des Senders
vars h_RecipientUserName = "user1"                // Name des Empfänges
vars h_DocID = "D000000153"                       // Dokumenten-ID

// Ermitteln von Wiedervorlage-Eigenschaften
// (nützlich um z.B. die Ketten-ID für das Löschen mit
// "hold_file_delete" zu erhalten)
h_ReturnValue = api_function ("hold_file_get_document_info", \
                               h_SenderUserName, \
                               h_RecipientUserName, h_DocID, "")

// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    call api_log_info("Datum:      :api_hold_file_date")
    call api_log_info("Empfänger: :api_hold_file_recipient")
    call api_log_info("Betreff:    :api_hold_file_notice")
    call api_log_info("Gesendet:   :api_hold_file_tstamp_sent")
    call api_log_info("Sender:     :api_hold_file_sender")
    call api_log_info("Ketten ID: :api_hold_file_chain_id")
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
                       zurückgeliefert!")
}
}
```

1.4.9. Workflow**workpath_get_document_info**

This function queries information on the current workflow. The values will be written into the global variables:

Global variables	Description
api_wfl_header_id	HeaderID
api_wfl_header_name	Workflow name
api_wfl_step_id	StepID
api_wfl_step_exit_string	StepExitString
api_wfl_step_tstamp_entry	TStampEntry
api_wfl_step_type	ElementType (1: step, 2: action)
api_wfl_step_recipient	Recipient
api_wfl_step_action_type	Action type (api_wfl_step_type=2) (1: status transfer, 2: resubmission, 3: JPL hook)

Global variables	Description
api_wfl_step_action_string_1	for api_wfl_step_action_type=1: Processing Verification, (= Verification, unverified) Verification,1 (= Verification, verified) Approval for api_wfl_step_action_type=2: Recipient of a resubmission for api_wfl_step_action_type=3: Name of the JPL-procedure
api_wfl_step_action_string_2	for api_wfl_step_action_type=1: Editor / group for api_wfl_step_action_type=2: Subject line of the resubmission

Parameters	Description
doc_id	ID of the document to be searched. If nothing is specified, the ID of the context is used.
user_name	User whose name is to be searched for; if nothing is specified, the executing user is used

Return value	Meaning
0	everything OK
> 0	Database error
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-3	undefined error, contact d.velop support
-101	neither document ID passed nor context set
-201	invalid user name specified
-220	Invalid workflow ID specified; inconsistent DB entries, contact d.velop support
-301	invalid document ID specified or user has got no rights to it
-302	Error while determining the user rights
-303	User has got no access permission to document
-523	The document is not currently active in a workflow.
-524	The workflow support is not enabled for this archive.

workpath_end_document

This function removes the document from a workflow.

Note

new parameter since version 6.3.1 hotfix 1

`delete_jobs`

The JPL server-API-function `workpath_end_document` now optionally removes all related possible existing workflow jobs in the asynchronous job queue after a document was successfully removed from a workflow.

Parameters	Description
doc_id	ID of the document. If nothing is specified, the ID of the context is used.
user_name	User for whom shall be searched; if nothing is specified, then the executive user will be used
delete_jobs	0/1: remove all respective workflow jobs still existing in the asynchronous job queue (default: 0)

Return value	Meaning
0	everything OK

Return value	Meaning
> 0	Database error
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-3	undefined error, contact d.velop support
-101	neither document ID passed nor context set
-201	invalid user name specified
-301	invalid document ID specified or user has got no rights to it
-302	Error while determining the user rights
-303	User has got no access permission to document
-523	The document is not currently active in a workflow.
-524	The workflow support is not enabled for this repository

In this example, the document with the specified document ID is removed from the workflow. If an error occurs, then the error number is returned.

Note

```
// -----
// Beispiel:
// Entfernen eines Dokumentes aus einem Leitweg
// -----

// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_DocIdRef = "D000000284" // Dokumenten-ID

// Funktionsaufruf
h_ReturnValue = api_function ("workpath_end_document",
h_DocIdRef, "")

// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    call api_log_info("Dokument aus Leitweg entfernt.")
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
    zurückgeliefert! ")
}
```

workpath_start_document

This function places the document into the workflow.

Parameters	Description
wfl_id	ID of the workflow into which the document is to be placed
doc_id	ID of the document. If nothing is specified, the ID of the context is used.
user_name	User for whom shall be searched; if nothing is specified, then the executive user will be used

Return value	Meaning
0	everything OK

Return value	Meaning
> 0	Database error
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-3	undefined error, contact d.velop support
-101	neither document ID passed nor context set
-122	no workflow ID specified
-201	invalid user name specified
-220	Invalid workflow ID specified
-301	invalid document ID specified or user has got no rights to it
-302	Error while determining the user rights
-303	User has got no access permission to document
-524	The workflow support is not enabled fort his archive.
-525	This workflow cannot be started manually
-526	The document is already active in a workflow.

In this example, the document with the document ID "D000000436" is activated in the specified workflow using the function call. If an error occurs, then the error number is returned.

Note

```
// -----
// Beispiel:
// Start eines Leitweges
// -----

// Variablendeklaration
vars h_ReturnValue                                // Fehlercode
vars h_DocIdRef = "D000000436"                    // Dokumenten-ID
vars h_WflId = "000000000000MT059ITI100CP200"    // ID des zu
startenden                                       // Workflows

// Funktionsaufruf
h_ReturnValue = api_function ("workpath_start_document",
h_WflId, \
                                h_DocIdRef, "")

// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    call api_log_info("Leitweg für Dokument gestartet.")
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
zurückgeliefert! ")
}
```

workpath_go_to_next_step

This function moves the document in the workflow, either by specifying a decision value or by specifying the next step.

Note

This function can (from d.3 version 6.2) no longer be called for the document from JPL code in a workflow step.

The function moves the position of a document within a WFL step to another step.

In the future, this function is only applicable from the d.3 flow designer since its call might cause problems.

Parameters	Description
exit_value	Exit with which the step is to be completed: 1: still in start element 2: TRUE 3: FALSE 100: Manual move
next_step_id	ID of the step, with which the workflow is to be continued (required for exit_value = 100)
doc_id	ID of the document. If nothing is specified, the ID of the context is used.
user_name	User, in whose name the action shall be performed. If nothing is specified, then the executive user will be used.

Return value	Meaning
0	everything OK
> 0	Database error
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-3	undefined error, contact d.velop support
-101	neither document ID passed nor context set
-201	invalid user name specified
-212	invalid Doc ID specified.
-220	Invalid workflow ID specified
-221	invalid workflow step ID specified
-222	invalid exit value specified
-301	invalid document ID specified or user has got no rights to it
-302	Error while determining the user rights
-303	User has got no access permission to document
-523	The document is not currently active in a workflow.
-524	The workflow support is not enabled for this repository
-527	Error while determining the current workflow position.

Example A returns the exit value TRUE for the document with the document ID "D000000284" at its current workflow position.

Example B specifies the next workflow step for the document.

Note

```
// -----
// Beispiel A:
// Weiterleitung durch Vorgabe eines Entscheidungswertes
// -----

// Variablendeklaration
vars h_ReturnValue           // Fehlercode
vars h_DocRefID = "D000000284" // Dokumenten-ID

// Funktionsaufruf
call api_function ("workpath_go_to_next_step", 2, "",
h_DocRefID, "")
```

Note

```
// -----
// Beispiel B:
// Weiterleitung durch Vorgabe des neuen Schritts
// -----

// Variablendeklaration
vars h_ReturnValue           // Fehlercode
vars h_DocRefID = "D000000284" // Dokumenten-ID
vars next_step_ref_id = "00000000000000MT089ITI10OCP200"
                                // ID des nächsten Workflowschritts

// Funktionsaufruf
call api_function
    ("workpath_go_to_next_step", 100, next_step_ref_id, \
    h_DocRefID)
```

1.4.10. Document type**document_type_get_long**

This function determines the document ID type long text either for a document ID or for a document type short name. If both are specified, then the `doc_id` is used. If nothing is specified, then the `doc_id` of the context is used. The long text is stored in `api_single_info`.

Parameters	Description
<code>doc_type_short</code>	Short name of the long text to be determined. optional
<code>doc_id</code>	Document, whose long text is to be queried. (optional – may be used for <code>api_doc_id</code>)

Return value	Meaning
0	everything OK
> 0	Database error
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect

Return value	Meaning
-101	neither document ID passed nor context set
-501	Error during determination of the document type of the document <\:doc_id>

The first example returns the document type name of the document type identified by the document ID. If an error occurs, then the error number is returned.

Example B determines the document type name based on the given document type short name. If an error occurs, then the error number is returned.

Note

```
// -----
// Beispiel A:
// Abfragen des Dokumentartnamen anhand ID
// -----

// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_DocID = "D000000153" // Dokumenten-ID

// Funktionsaufruf: Ermitteln des Langtextes der Dokumentart
// zu einem vorgegebenen Dokument
h_ReturnValue = api_function ("document_type_get_long", "",
h_DocID)

// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    call api_log_info("Dokument hat als Dokumentart
:api_single_info")
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
zurückgeliefert!")
}
```

Note

```
// -----
// Beispiel B:
// Abfrage des Dokumentartnamen anhand des Kürzels
// -----

// Variablendeklartion
vars h_ReturnValue          // Fehlercode
vars h_DocTypeShort = "DACAD" // Dokumentart-Kürzel

// Funktionsaufruf: Ermitteln des Langtextes der Dokumentart
// zu vorgegebenem Dokumentart-Kürzel
h_ReturnValue = api_function ("document_type_get_long", \
                             h_DocTypeShort, "")

// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    call api_log_info("Dokumentart hat als Bezeichnung: \
                      :api_single_info")
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
                      zurückgeliefert!")
}
```

document_type_get_short

This function determines the document type short name either for a document ID or for a document type long text.

If both are specified, then the `doc_id` is used.

If nothing is specified, then the `doc_id` of the context is used.

The short text is stored in `api_single_info`.

Parameters	Description
<code>doc_type_long</code>	Long text for which the ID is to be determined. optional
<code>doc_id</code>	Document, whose document type short name is to be queried. (optional – may be used for <code>api_doc_id</code>)

Return value	Meaning
0	everything OK
> 0	Database error
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-101	neither document ID passed nor context set
-501	Error during determination of the document type of the document <\:doc_id>
-502	Error during determination the document type long text for this ID.

The first example returns the document type abbreviation for a document type, which is identified by a document ID. If an error occurs, then the error number is returned.

The second example does not call the function with a document ID, but with the document type name.

Note

```
// -----  
// Beispiel A:  
// Abfrage des Dokumentartkürzels anhand ID  
// -----  
  
// Variablendeklaration  
vars h_ReturnValue          // Fehlercode  
vars h_DocID = "D000000153" // Dokumenten-ID  
  
// Funktionsaufruf:  
// Ermitteln des Dokumentart-Kürzels eines vorgegebenen Dokument  
h_ReturnValue = api_function ("document_type_get_short", "",  
h_DocID)  
  
// Ergebnisausgabe und Fehlerbehandlung  
if (h_ReturnValue == 0)  
{  
    call api_log_info("Dokument hat als Dokumentart-Kürzel \  
                      :api_single_info")  
}  
else  
{  
    call api_log_error("Fehlernummer :h_ReturnValue wurde  
                      zurückgeliefert!")  
}
```

Note

```
// -----
// Beispiel B:
// Abfrage des Dokumentartkürzels anhand Name
// -----

// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_DocTypeName = "AutoCAD" // Dokumentart-Kürzel

// Funktionsaufruf: Ermitteln des Dokumentart-Kürzels
// zu vorgegebener Dokumentart-Bezeichnung
h_ReturnValue = api_function ("document_type_get_short", \
                               h_DocTypeName, "")

// Ergebnisausgabe
if (h_ReturnValue == 0)
{
    call api_log_info("Dokumentart hat als Kürzel:
:api_single_info")
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
zurückgeliefert!")
}
```

1.4.11. Datasets**predefined_values_get**

The function queries a dataset.

The return parameters are stored in the global variable `api_data_type` (=data type of the value set) and the global arrays

`d3server_value_char_allowed[]`,

`d3server_value_date_allowed[]`,

`d3server_value_num_allowed[]`.

The number of returned values is stored in the variable `api_single_info`.

Warning

The function queries only the static datasets.

Parameters	Description
<code>pre_val_id</code>	ID of the dataset, whose values shall be read. (has priority over <code>pre_val_name</code>)
<code>pre_val_name</code>	Name of the dataset, whose values shall be read. (alternative to <code>pre_val_id</code>)

Return value	Meaning
0	everything OK
> 0	Database error
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-119	Neither value set-ID nor value set-name specified
-218	Invalid value set-ID or value set-name specified

Example A returns the name of a value set identified by its ID. If an error occurs, then the error number is returned.

Example B performs the same task. However, it is called using the name of the value set.

Note

```
// -----
// Beispiel A:
// Angabe der ID der Wertemenge
// -----

// Variablendeklaration
vars h_ReturnValue      // Fehlercode
vars h_PreValID= 42     // ID der Wertemenge
vars h_Count            // Fehlercode

// Funktionsaufruf
h_ReturnValue = api_function ("predefined_values_get",
h_PreValID)

// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    if (api_single_info > 0)
    {
        for h_Count = 1 while h_Count <= api_single_info step 1
        {
            // Ausgabe des Ergebnisses mit Hilfe des Arrays
            // d3server_value_char,
            // da die Wertemenge im Beispiel alphanumerisch ist
            call api_log_info("Eintrag :h_Count: \
:d3server_value_char_allowed[:h_Count]")
        }
    }
    else
    {
        call api_log_info("Die Wertemenge enthält keine Einträge.
")
    }
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
zurückgeliefert! ")
}
```

Note

```
// -----
// Beispiel B:
// Angabe des Namens der Wertemenge
// -----

// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_PreValName = "Ausgabe" // Name der Wertemenge
vars h_Count                // Fehlercode

// Funktionsaufruf
h_ReturnValue = api_function
    ("predefined_values_get", "", h_PreValName)

// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    if (api_single_info > 0)
    {
        for h_Count = 1 while h_Count <= api_single_info step 1
        {
            // Ausgabe des Ergebnisses mit Hilfe des Arrays
            // d3server_value_char,
            // da die Wertemenge im Beispiel alphanumerisch
            ist
                call api_log_info("Eintrag :h_Count: \
:d3server_value_char_allowed[:h_Count]")
            }
        }
    }
    else
    {
        call api_log_info("Die Wertemenge enthält keine Einträge.
")
    }
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
    zurückgeliefert! ")
}
}
```

predefined_values_test

This function checks values against a dataset.

Parameters	Description
test_value	Value to be tested
pre_val_id	ID of the dataset, whose values shall be read. (has priority over pre_val_name)
pre_val_name	Name of the dataset, whose values shall be read. (alternative to pre_val_id)

Return value	Meaning
1	Value is in the dataset
0	Value is not in the dataset
< 0	Database error
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-119	Neither value set-ID nor value set-name specified
-218	Invalid value set-ID or value set-name specified

Note

available from d.3 version 5.5.1 hotfix 15, 6.0

This example code verifies, if the value "Gronau" is included in the value set with the ID 39 and returns the result. If an error occurs, then the error number is returned.

Note

```
// -----
// Beispiel:
// Prüfen eines Wertes gegenüber Wertemenge
// -----

// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_TestValue= "Gronau"  // Testwert
vars h_PreValID= 39         // ID der Wertemenge

// Funktionsaufruf
h_ReturnValue = api_function ("predefined_values_test", \
                             h_TestValue, h_PreValID)

// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 1)
{
    call api_log_info("Der Eintrag ist in der Wertemenge
    vorhanden. ")
}
else if( h_ReturnValue == 0 )
{
    call api_log_info("Der Eintrag ist in der Wertemenge
    nicht vorhanden. ")
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
    zurückgeliefert! ")
}
```

1.4.12. User**user_get**

The function for the query of all user names. The result is stored in the global array `api_users[]`. The number of returned values is stored in the variable `api_single_info`.

Parameters	Description
user_type	1: non-system-user (default)
	2: system-user (d3_server, ...)
	3: system- and non-system-users

Note

available from d.3 version 6.0 HF 3

Return value	Meaning
0	everything OK
> 0	Database error
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect

In this example, all system-user names are queried and returned. If an error occurs, then the error number is returned.

Note

```

/ -----
// Beispiel:
// Abfrage des Systembenutzers
// -----

// Variablendeklaration
vars h_ReturnValue      // Fehlercode
vars h_UserType = 2     // Benutzertyp -> 2: System-Benutzer
vars h_Count            // Zähler für die Schleife

// Funktionsaufruf
h_ReturnValue = api_function ("user_get", h_UserType)

// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    // Ausgabe der Benutzeranzahl
    call api_log_info ("Es wurden :api_single_info Benutzer
gefunden. ")

    // Ausgabe der Benutzernamen
    for h_Count = 1 while h_Count <= api_single_info step 1
    {
        call api_log_info (":api_users[:h_Count]")
    }
}

```

user_is_system

This function tests, if a specified user name is a system-user. The result (0=no system -user, 1=system-user) is store in api_single_info.

Parameters	Description
user_name	Name of the user to be tested.

Return value	Meaning
0	no errors
> 0	Database error
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect

Note

available from d.3 version 6.0 HF 3

His example checks, if the user "dvelop" is a system-user and returns the result. The result is displayed. If an error occurs, then the error number is returned.

Note

```
// -----
// Beispiel:
// Prüfen ob Benutzer ein Systembenutzer ist
// -----
// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_UserName = "dvelop"  // Benutzername

// Funktionsaufruf; das Ergebnis wird in api_single_info
// gespeichert
h_ReturnValue = api_function ("user_is_system", h_UserName)
// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    // Ausgabe des Ergebnisses anhand des in api_single_info
    // geschriebenen
    // Wertes: 0 -> Kein Systembenutzer
    //          1 -> Systembenutzer
    if (api_single_info == 0)
    {
        call api_log_info("Der Benutzer ist kein System-Benutzer.
")
    }
    if (api_single_info == 1)
    {
        call api_log_info("Der Benutzer ist ein System-Benutzer. ")
    }
}
else
{
    // Fehlerausgabe
    call api_log_error("Fehlernummer :h_ReturnValue wurde
        zurückgeliefert! ")
}
```

user_get_groups

This function returns the groups, of which the specified user is a member.

The result is stored in the global array `api_groups[]`.

The number of returned values is stored in the variable `api_single_info`.

Note

New in version 7.1:

In `api_groups_id[]` the group-IDs of the determined user groups are returned.

Parameters	Description
<code>user_name</code>	User, for which the group memberships shall be queried. if nothing is specified, then the executive user will be used.
<code>group_name</code>	optional; group name, which shall be verified <code>api_groups[1]</code> then contains this group name, if the user is a member or is empty, if the user is no member of this group.

Note
changed for version 7.0:

Parameter:
`group_name` can now also be a group ID.

Return value	Meaning
0	everything OK
> 0	Database error
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-201	invalid user name specified

Example A queries all group memberships for the user "dvelop" and returns them. For this effect, this function is only called with the user name as the parameter. If an error occurs, then the error number is returned.

Example B calls this function with both parameters. It checks, if the user "dvelop" is a member of the group "Purchasing". If an error occurs, then the error number is returned.

Note

```
// -----  
// Beispiel A:  
// Abfrage aller Gruppenmitgliedschaften  
// -----  
  
// Variablendeklaration  
vars h_ReturnValue          // Fehlercode  
vars h_UserName = "dvelop"  // Benutzername  
vars h_Count                // Zähler für Schleife  
  
// Funktionsaufruf: Die Anzahl der Gruppenmitgliedschaften  
// wird in api_single_info gespeichert  
// Die Name der Gruppen werden in dem Array api_groups  
// gespeichert  
h_ReturnValue = api_function ( "user_get_groups", h_UserName)  
  
// Ergebnisausgabe und Fehlerbehandlung  
if (h_ReturnValue == 0)  
{  
    call api_log_info("Der Benutzer :h_UserName ist Mitglied in \  
                      :api_single_info Gruppen. ")  
    for h_Count = 1 while h_Count <= api_single_info step 1  
    {  
        call api_log_info("Der Benutzer :h_UserName ist Mitglied \  
                          in der Gruppe :api_groups[h_Count]")  
    }  
}  
else  
{  
    // Fehlerausgabe  
    call api_log_error("Fehlernummer :h_ReturnValue wurde  
                      zurückgeliefert! ")  
}
```

Note

```
// -----
// Beispiel B:
// Überprüfung einer Gruppenmitgliedschaft
// -----

// Variablendeklaration
vars h_ReturnValue
vars h_UserName = "dvelop" // Benutzername
vars h_GroupName = "Vertrieb" // Gruppenname

// Funktionsaufruf:
// api_groups[1] enthält den Gruppennamen bei Mitgliedschaft
// api_groups[1] ist leer, wenn der User kein Mitglied ist
h_ReturnValue = api_function ("user_get_groups", h_UserName, \
                               h_GroupName )

// Ergebnisausgab
if (h_ReturnValue == 0)
{
    if (api_groups[1] != "")
    {
        call api_log_info("Der Benutzer IST Mitglied der Gruppe \
                           :api_groups[1]. ")
    }
    else
    {
        call api_log_info("Der Benutzer ist KEIN Mitglied der
Gruppe. ")
    }
}
// Fehlerbehandlung
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
zurückgeliefert! ")
}
```

user_get_verifier_groups

This function specifies, in which group the user is defined as a verifier.

The result is stored in the global array `api_groups[]`.

The number of returned values is stored in the variable `api_single_info`.

Parameters	Description
<code>user_name</code>	User, for which the group memberships shall be queried. if nothing is specified, then the executive user will be used.

Return value	Meaning
0	everything OK
> 0	Database error
-1	Panther syntax error, details see d.3 logviewer

Return value	Meaning
-2	function was called incorrect
-201	invalid user name specified

The example specifies, in which groups the user `dvelop` was defined with verifying rights. The group names and the overall number of groups are returned. If an error occurs, then the error number is returned.

Note

```
// -----
// Beispiel:
// Bestimmt Gruppen mit Prüfrechten für Benutzer
// -----

// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_UserName = "dvelop"  // Benutzername
vars h_Count                // Zähler für Schleife

// Funktionsaufruf:
// Die Anzahl der Gruppenmitgliedschaften in denen der Benutzer
// Prüfer ist
// wird in api_single_info gespeichert.
// Die Name der Gruppen werden in dem Array api_groups
// gespeichert.
h_ReturnValue = api_function ("user_get_verifier_groups",
h_UserName )

// Ergebnisausgabe und Fehlerbehandlung
// Ergebnisausgabe
if (h_ReturnValue == 0)
{
    call api_log_info("Der Benutzer :h_UserName ist Prüfer in \
                        :api_single_info Gruppen. ")
    for h_Count = 1 while h_Count <= api_single_info step 1
    {
        call api_log_info("Der Benutzer :h_UserName ist Prüfer \
                            in der Gruppe :api_groups[h_Count]")
    }
}
// Fehlerbehandlung
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
                        zurückgeliefert! ")
}
```

user_get_rolls

The function specifies, which authorization profiles are assigned to the user.

The result is stored in the global array `api_groups[]`.

The number of returned values is stored in the variable `api_single_info`.

Parameters	Description
<code>user_name</code>	User, for which the group memberships shall be queried. If nothing is specified, then the executive user will be used.

Return value	Meaning
0	everything OK
> 0	Database error
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-201	invalid user name specified

This example checks which authorization profiles are assigned to the user "dvelop". The count of found profiles and their names will be returned. If an error occurs, then the error number is returned.

Note

```
// -----
// Beispiel:
// Abfrage von Benutzerprofilen
// -----

// Variablendeklaration
vars h_ReturnValue      // Fehlercode
vars h_UserName = "dvelop" // Benutzername
vars h_Count            // Zähler für Schleife

// Funktionsaufruf:
// Die Anzahl der Rollen denen der Benutzer zugeordnet ist
// wird in api_single_info gespeichert.
// Die Name der Rollen werden in dem Array api_rolls gespeichert.
h_ReturnValue = api_function ("user_get_rolls", h_UserName)

// Ergebnisausgabe
if (h_ReturnValue == 0)
{
    call api_log_info("Der Benutzer :h_UserName ist \
                      :api_single_info Rollen zugeordnet. ")
    for h_Count = 1 while h_Count <= api_single_info step 1
    {
        call api_log_info("Dem Benutzer :h_UserName ist die Rolle
\
                      :api_rolls[h_Count] zugeordnet")
    }
}
// Fehlerbehandlung
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
                      zurückgeliefert! ")
}
```

user_get_doc_types

This function queries the document types to which a user has got access. The document type short names are stored in the global array `api_doc_types[]` the number of found document types is stored in the variable `api_single_info`.

Parameters	Description
<code>user_name</code>	Name of the user to be tested.

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-3	undefined error, contact d.velop support

Note

available from d.3 version 6.0 HF 1

This example tests a number of document types specified by their IDs and returns those to which the user "dvelop" has got access. If an error occurs, then the error number is returned.

Note

```
// -----
// Beispiel:
// Abfrage von Dokumentart mit Zugriff
// -----

// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_UserName = "dvelop"  // Benutzername
vars h_Count                // Zähler für Schleife

// Funktionsaufruf:
// Die Anzahl der Dokumentenarten auf die der Benutzer Zugriff
// hat wird in api_single_info gespeichert.
// Die Namen der Dokumentenarten werden in dem Array
api_doc_types
// gespeichert.
h_ReturnValue = api_function ("user_get_doc_types", h_UserName)

// Ergebnisausgabe
if (h_ReturnValue == 0)
{
    call api_log_info("Der Benutzer :h_UserName hat Zugriff auf \
                        :api_single_info Dokumentarten. ")
    for h_Count = 1 while h_Count <= api_single_info step 1
    {
        call api_log_info("Der Benutzer :h_UserName hat Zugriff
auf \
                        Dokumentart
:api_doc_types[h_Count]. ")
    }
}
// Fehlerbehandlung
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
zurückgeliefert! ")
}
```

user_get_data

In this function, user data will be written into the global variables of the user context:

Global variables	Description
api_user_name	User name
api_user_right_level	Rights level (1..5)
api_user_right_archive	Right to view of archive versions (0/1: no/yes)
api_user_right_link	Right to link folders (0/1: no/yes)

Global variables	Description
api_user_right_flags	Right flags (0/1: no/yes) Position 1: Permission to create links Position 2: Web access permitted Position 3: User disabled Position 4: User is blocked Position 5: Admin rights Position 6-10: reserved
api_user_field[1..10]	user related fields
api_user_checkout_flag	User ist absent (0/1: no/yes)
api_user_checkout_text	Absence text of the user
api_user_realname	Name
api_user_email	e-mail address
api_user_phone	Phone number
api_user_plant	Plant
api_user_department	Department
api_user_tstamp_current_login	Time stamp of current login
api_user_tstamp_last_login	Time stamp of last login
api_user_tstamp_hf_info	Time stamp of the information about new resubmissions
api_user_ldap_dn	LDAP name
api_user_change_count	Alteration number
api_user_pw_valid_until	Date up to which the current password is valid
api_user_tstamp_login_failed	Time stamp of the login attempt with wrong password
api_user_count_login_failed	Count of login attempts with wrong password

Parameters	Description
user_name	User, for which the data shall be queried. If nothing is specified, then the executive user will be used.

Return value	Meaning
0	everything OK
> 0	Database error
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-201	invalid user name specified

This example reads the user data of the user "dvelop" and writes them to the global variables. These global variables are then partly displayed. If an error occurs, then the error number is returned.

Note

```

// -----
// Beispiel:
// Abfrage von Benutzerdaten
// -----

// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_UserName = "dvelop"  // Benutzername
vars h_Count                // Zähler für Schleifen

// Funktionsaufruf
h_ReturnValue = api_function ("user_get_data", h_UserName)

// Ergebnisausgabe
if (h_ReturnValue == 0)
{
    // Recht zur Aktenverknüpfung
    // 0=nein, 1=ja
    if (api_user_right_link == 0)
    {
        call api_log_info("Der Benutzer hat nicht das Recht zur \
            Aktenverknüpfung. ")
    }
    else
    {
        call api_log_info("Der Benutzer hat das Recht zur \
            Aktenverknüpfung. ")
    }
    // Rechteflags:
    // api_user_right_flags(1,1): Recht Verknüpfungen zu erstellen
    if (api_user_right_flags(1,1) == 0)
    {
        call api_log_info("Der Benutzer darf keine Verknüpfungen \
            anlegen. ")
    }
    else
    {
        call api_log_info("Der Benutzer darf Verknüpfungen
anlegen")
    }
    // Rechteflags:
    // api_user_right_flags(2,1): Webzugriff
    if (api_user_right_flags(2,1) == 0)
    {
        call api_log_info("Der Benutzer hat keinen Webzugriff. ")
    }
    else
    {
        call api_log_info("Der Benutzer hat Webzugriff. ")
    }
    // Rechteflags:
    // api_user_right_flags(3,1): Benutzer deaktiviert
    if (api_user_right_flags(3,1) == 0)

```

```

    {
        call api_log_info("Der Benutzer ist nicht deaktiviert. ")
    }
else
{
    call api_log_info("Der Benutzer ist deaktiviert. ")
}
// Rechteflags:
// api_user_right_flags(4,1): Benutzer gesperrt
if (api_user_right_flags(4,1) == 0)
{
    call api_log_info("Der Benutzer ist nicht gesperrt. ")
}
else
{
    call api_log_info("Der Benutzer ist gesperrt. ")
}
// Benutzerbezogene Felder, 10 Felder insgesamt)
for h_Count = 1 while h_Count <= 10 step 1
{
    call api_log_info("Benutzerbezogenes Feld :h_Count: \
                        :api_user_field[h_Count]")
}
// Abwesenheit & Abwesenheitstext
if (api_user_checkout_flag == 0)
{
    call api_log_info("Der Benutzer ist anwesend. ")
}

else
{
    call api_log_info("Der Benutzer ist abwesend: ")
    call api_log_info(api_user_checkout_text)
}
// Name, Email, Telefonnummer, Abteilung und Werk des
Benutzers
call api_log_info("Name: :api_user_realname")
call api_log_info("Email: :api_user_email")
call api_log_info("Telefon: :api_user_phone")
call api_log_info("Abteilung: :api_user_department")
call api_log_info("Werk: :api_user_plant")
}
// Fehlerbehandlung
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
                        zurückgeliefert! ")
}

```

user_validate_password

Tests, if an user with this password is known in d.3.

Note

This function cannot be used for LDAP-users.

Parameters	Description
password	Password to be tested
user	Name of the user to be tested.

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-620	Username / password invalid

Tests, if an user with this password is known in d.3 (not LDAP).

Note

```
// -----
// Beispiel:
// Prüfen, ob Benutzername und Passwort zusammenpassen
// -----
// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_UserName = "armin"   // Benutzername
vars h_Password = "frikadelle" // Passwort
// Funktionsaufruf
h_ReturnValue = api_function ("user_validate_password",
h_Password, \
                        h_UserName)

// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    call api_log_info ("Kombination aus Benutzername und Passwort
gültig!")
}
else
{
    call api_log_info ("Kombination aus Benutzername und Passwort
ungültig!")
}
```

user_get_favourites

Returns the document ID of the favorites of the user in the array `api_doc_ids`. The count of favorites will be returned in the return value of the function and will be additionally written into the `api_single_info`.

Note

A maximum of 500 changes is logged.

Parameters	Description
user_name	User for which the favourites are to be read.

Return value	Meaning
0	everything OK

Return value	Meaning
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-3	undefined error, contact d.velop support

Returns the document IDs of a users's favourites.

Note

```
vars ret_value
ret_value = api_function ("user_get_favourites", "dvelop")
if (ret_value > 0)
{
  vars i
  //Ausgabe aller Dokument-IDs der Favoriten
  for i=1 while i<=ret_value step 1
    call api_log_info(":i :api_doc_ids[i]")
}
```

user_get_sid_rights

Description: Determines the user-specific SID-permissions/properties S-1-*-**

Parameters	Description
user_name	User, for which the data shall be queried. If nothing is specified, then the executive user will be used.
SID	Optional; SID in the form S-1-*-**. Only userpermissions (S-1-*-**) can be queried.

Global variables	Meaning
api_single_info	Number of elements in the following fields
api_user_sid_str[]	SID-string in the form S-1-*-**.
api_user_sid_right[]	0: right/property revoked/not set 1: right/property granted/set

Return value	Meaning
0	everything OK
> 0	Database error

user_checkout

Function for reporting the presence or absence of d.3 users.

Parameters	Description
in_out	in = user will be set as checked in out = user will be set as checked out
checkout_text	Absence text
check_user	User to be checked in or checked out
call_user	Executing user who must be a superior in case of check_user being set. If check_user is not set, then call_user is used.
force_dist	1: Enables the immediate redirection of mailbox entries for the checked-out user, which were sent to him via activity profiles.

Return value	Meaning
0	everything OK

Return value	Meaning
3	invalid or missing parameter.
120	User (<i>granter</i> or <i>grantee</i>) does not exist.
420	<i>call_user</i> is not the superior of <i>check_user</i>
423	The selected alternative user is currently checked out.
9000	Database-error (see logviewer for details)

user_get_superior

Function determines the superior of a user.

Parameters	Description
<i>user_name</i>	User for whom the superiors are to be determined. If no user is passed, the caller of the function is used.

Global variables	Meaning
<i>api_single_info</i>	Number of elements in the following fields
<i>api_users[]</i>	List of superiors

Return value	Meaning
0	everything OK
120	User data was not found in cache.

1.4.13. Groups

group_get

The function queries group names. The result is stored in the global array *api_groups[]*. The number of returned values is stored in the variable *api_single_info*.

Parameters	Description
<i>language</i>	Language for determination of the group nams (default = 049)

Return value	Meaning
0	everything OK
> 0	Database error
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect

Note

new from d.3 version 7.0:

The IDs for the groups from *api_groups[]* will be stored in the global array *api_groups_id[]*.

Note

```

/ -----
// Beispiel:
// Abfrage aller Gruppennamen
// -----

// Variablendeklaration
vars h_ReturnValue    // Fehlercode
vars h_Count          // Zähler für Schleife

// Funktionsaufruf
h_ReturnValue = api_function ("group_get")

// Ergebnisausgabe
if (h_ReturnValue == 0)
{
    call api_log_info("Es gibt :api_single_info Gruppen")
    for h_Count = 1 while h_Count <= api_single_info step 1
    {
        call api_log_info("Gruppe :h_Count:
:api_groups[:h_Count]")
    }
}
// Fehlerbehandlung
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
zurückgeliefert! ")
}

```

group_get_members

The function queries the members of a group. The result is stored in the global array `api_users[]`. The number of returned values is stored in the variable `api_single_info`.

Parameters	Description
<code>group_name</code>	Name of the group, whose members shall be determined.
Note new in d.3 version 7.0: Parameter: <code>group_name</code> can now also be a group ID.	
<code>user_name</code>	User name (optional), to be checked. <code>api_users[1]</code> contains this user name, if it is a member or remains empty, if the user is not a member of this group.
<code>check_sub-groups</code>	New from d.3 version 7.2. optional; default = 0, if = 1, then all sub-groups of <code>group_name</code> will be searched through.

Return value	Meaning
0	everything OK
> 0	Database error
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect

Return value	Meaning
-113	no group name specified
-213	invalid group name specified

Example A returns all members of the group "Purchasing".

The group is identified with its name here (`group_name = "Purchasing"`).

After the function call, the result is displayed (for `h_ReturnValue=0`, i.e. no errors) or the error number is returned.

Example B checks, if the specified user name (`h_UserName = "dvelop"`) is a member of the given group (`group_name = "Purchasing"`). After the function call, the result is displayed (for `h_ReturnValue=0`, i.e. no errors) or the error number is returned. If no name is stored in the variable `api_users` after a successful function call, then the user is not a member of the group.

If the user is a member of the groups, then the user name is stored in `api_users`.

Note

```
// -----
// Beispiel A:
// Ausgabe aller Gruppenmitglieder
// -----

// Variablendeklaration
vars h_ReturnValue           // Fehlercode
vars h_GroupName = "Vertrieb" // Angabe des Gruppennamens
vars h_Count                 // Zählervariable für die Schleife

// Funktionsaufruf
h_ReturnValue = api_function ("group_get_members", h_GroupName)

// Ergebnisausgabe
if (h_ReturnValue == 0)
{
    call api_log_info("Es gibt :api_single_info
Gruppenmitglieder")
    for h_Count = 1 while h_Count <= api_single_info step 1
    {
        call api_log_info("Gruppenmitglied :h_Count: \
                        :api_users[:h_Count]")
    }
}
// Fehlerbehandlung
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
zurückgeliefert! ")
}
```

Note

```
// -----
// Beispiel B:
// Ist der angegebene Benutzer in der angegebenen Gruppe?
// -----

// Variablendeklaration
varsh_ReturnValue          // Fehlercode
varsh_GroupName="Vertrieb" // Angabe des Gruppennamens
varsh_UserName="dvelop"    // Angabe des Benutzernamens

// Funktionsaufruf
h_ReturnValue= api_function ("group_get_members", h_GroupName, \
                             h_UserName )

// Ergebnisausgabe
if (h_ReturnValue == 0)
{
    if (api_users != "")
    {
        call api_log_info("Der Benutzer :h_UserName \
                           ist Mitglied der Gruppe. ")
    }
    else
    {
        call api_log_info("Der Benutzer \
                           :h_UserName ist nicht Mitglied der Gruppe.
")
    }
}
// Fehlerbehandlung
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
                       zurückgeliefert! ")
}
```

group_get_verifiers**Note**

new in d.3 version 7.0:

Since verifier groups no longer exist, this function returns the same as the function `group_get_members`.

This function determines the verifier of a group. The result is stored in the global array `api_users[]`. The number of returned values is stored in the variable `api_single_info`.

Parameters	Description
<code>group_name</code>	Name of the group, whose verifiers shall be determined.

Return value	Meaning
0	everything OK
> 0	Database error
-1	Panther syntax error, details see logviewer
-2	function was called incorrect
-113	no group name specified
-213	invalid group name specified

The example retrieves the verifiers of the group "Purchasing".

All found verifiers are stored in the array `api_users[]`, which is browsed after a successful function call to display the names. If an error occurs, then the error number is returned.

Note

```
// -----
// Beispiel:
// Prüfer einer Gruppe abfragen
// -----

// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_GroupName = "Vertrieb" // Name der Gruppe
vars h_Count                // Zähler für die Schleife

// Funktionsaufruf
h_ReturnValue = api_function ("group_get_verifiers", h_GroupName)

// Ergebnisausgabe
if (h_ReturnValue == 0)
{
    call api_log_info("Der Gruppe sind :api_single_info \
                      Prüfer zugeordnet. ")
    for h_Count = 1 while h_Count <= api_single_info step 1
    {
        call api_log_info("Der Gruppe ist der Prüfer \
                          :api_users[h_Count] zugeordnet. ")
    }
}
// Fehlerbehandlung
else
{
    call api_log_info("Fehlernummer :h_ReturnValue wurde
                      zurückgeliefert! ")
}
```

group_get_doc_types

Note

new in d.3 version 7.0:

Since direct assignments of document types or user groups no longer exist, this does not return any results any more.

This function determines the document types assigned to a group. The result is stored in the global array `api_doc_types[]`.

The number of returned values is stored in the variable `api_single_info`.

Parameters	Description
<code>group_name</code>	Name of the group, whose document types shall be queried.

Return value	Meaning
0	everything OK
> 0	Database error
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-113	no group name specified
-213	invalid group name specified

The example queries all document types assigned to the group purchasing "Purchasing".

The names of the document types are stored in the array `api_doc_type[]`, the overall number is written into the variable `api_single_info`.

If an error occurs, then the error number is returned.

Note

```

/ -----
// Beispiel:
// Dokumentarten einer Gruppe abfragen
// -----

// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_GroupName = "Vertrieb" // Name der Gruppe
vars h_Count                // Zähler für die Schleife

// Funktionsaufruf
h_ReturnValue = api_function ("group_get_doc_types", h_GroupName)

// Ergebnisausgabe
if (h_ReturnValue == 0)
{
    call api_log_info("Der Gruppe sind \
                      :api_single_info Dokumentenarten
zugeordnet. ")
    for h_Count = 1 while h_Count <= api_single_info step 1
    {
        call api_log_info("Der Gruppe ist die Dokumentart \
                          :api_doc_types[h_Count] zugeordnet. ")
    }
}
// Fehlerbehandlung
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
zurückgeliefert! ")
}

```

1.4.14. Roles**roll_get**

This function queries all authorization profile IDs. The result is written into the array `api_rolls[]`, while the number of returned values is stored in the variable `api_single_info`.

Return value	Meaning
0	everything OK
> 0	Database error
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect

The example queries the user profiles with the function call Then, the authorization profile IDs which were stored in the array `api_rolls[]` are returned. If an error occurs, then the error number is returned.

Note

```
// -----
// Beispiel:
// Abfrage aller Benutzerprofil-IDs
// -----

// Variablendeklaration
vars h_ReturnValue // Fehlercode
vars h_Count       // Zähler für die Schleife

// Funktionsaufruf
h_ReturnValue = api_function ("roll_get")

// Ergebnisausgabe
if (h_ReturnValue == 0)
{
    call api_log_info("Es gibt :api_single_info Rollen")
    for h_Count = 1 while h_Count <= api_single_info step 1
    {
        call api_log_info("Es gibt die RolleID
:api_rolls[h_Count]")
    }
}
// Fehlerbehandlung
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
zurückgeliefert! ")
}
```

roll_get_names

This function queries all authorization profile names. The result is written into the array `api_rolls[]`. The number of returned values is stored in the variable `api_single_info`.

Return value	Meaning
0	everything OK
> 0	Database error
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect

This example queries all authorization profiles or their names and stores them in the array `api_rolls[]`. The number of found authorization profiles is stored in the variable `api_single_info`. The saved names are displayed after the function call. If an error occurs, then the error number is returned.

Note

```

/ -----
// Beispiel:
// Abfrage aller Benutzerprofilnamen
// -----

// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_Count                // Zähler für die Schleife

// Funktionsaufruf
h_ReturnValue = api_function ("roll_get_names")

// Ergebnisausgabe
if (h_ReturnValue == 0)
{
    call api_log_info("Es gibt :api_single_info Rollen")
    for h_Count = 1 while h_Count <= api_single_info step 1
    {
        call api_log_info(":api_rolls[h_Count]")
    }
}
// Fehlerbehandlung
else
{
    call api_log_info("Fehlernummer :h_ReturnValue wurde
        zurückgeliefert! ")
}

```

roll_get_users

This function shows all users who are assigned to authorization profiles.

The result is written into the array `api_users[]`.

The number of returned values is stored in the variable `api_single_info`.

Note

new in d.3 version 7.0:

Group IDs can now also be returned in `api_users` since user groups can now also be assigned to profiles.

Parameters	Description
<code>roll_id</code>	ID of the authorization profile whose assigned users are to be found (has priority over <code>roll_name</code>)
<code>roll_name</code>	Name of the authorization profile whose assigned users are to be found (alternative to <code>roll_id</code>)

Return value	Meaning
0	everything OK
> 0	Database error

Return value	Meaning
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-118	Neither authorization profiles ID nor Authorization profile name specified
-217	invalid authorization profiles ID or authorization profile name specified

Example A queries the user names which are assigned to the authorization profile ID 1. The number of users found is saved to the variable `api_single_info`, the names are stored in the array `api_users[]`. Afterwards the number of users and user names will be returned. If an error occurs, then the error number is returned.

Example B also returns the names of the users assigned to an authorization profile and the number of users assigned. However, the function call is different from the first one, as the name of the authorization profile is passed rather than the ID.

Note

```
// Beispiel A:
// Angabe der Benutzerprofil-ID
// -----

// Variablendeklaration
vars h_ReturnValue      // Fehlercode
vars h_RollenId = 1     // ID der Rolle
vars h_Count            // Zähler für die Schleife

// Funktionsaufruf
h_ReturnValue = api_function ("roll_get_users", h_RollenId)

// Ergebnisausgabe
if (h_ReturnValue == 0)
{
    call api_log_info("Es gibt :api_single_info \
                      Mitglieder in der Rolle ")
    for h_Count = 1 while h_Count <= api_single_info step 1
    {
        call api_log_info(":api_users[h_Count] ist Mitglied")
    }
}
// Fehlerbehandlung
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
                      zurückgeliefert! ")
}
```

Note

```
// -----
// Beispiel B:
// Angabe des Benutzerprofil-Names
// -----

// Variablendeklaration
vars h_ReturnValue      // Fehlercode
vars h_RollName = "Mail" // Name der Rolle
vars h_Count            // Zähler für die Schleife

// Funktionsaufruf
h_ReturnValue = api_function( "roll_get_users", "", h_RollName )

// Ergebnisausgabe
if (h_ReturnValue == 0)
{
    call api_log_info("Es gibt :api_single_info \
                      Mitglieder in der Rolle ")
    for h_Count = 1 while h_Count <= api_single_info step 1
    {
        call api_log_info(":api_users[h_Count] ist Mitglied")
    }
}
// Fehlerbehandlung
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
                      zurückgeliefert! ")
}
}
```

roll_get_data

This function queries the authorization profile data (class, rights). The result is stored in the global variables:

Global variables	Description
api_roll_id	ID of the role
api_roll_name	Name of the role

As well as the global arrays:

Global arrays	Description
api_classes[]	The class assigned to the authorization profile.

Global arrays	Description
<code>api_class_right[]</code>	Right of the authorization profile for the class in <code>api_classes</code> with the same index. 0: Right denied 1: Right assigned 2: not specified Position 1: Read Position 2: Write Position 3: Validate Position 4: Release

Parameters	Description
<code>roll_id</code>	ID of the authorization profile whose data shall be found (has priority over <code>roll_name</code>)
<code>roll_name</code>	Name of the authorization profile whose data shall be found (alternative to <code>roll_id</code>)

Return value	Meaning
0	everything OK
> 0	Database error
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-118	Neither authorization profiles ID nor Authorization profile name specified
-217	invalid authorization profiles ID or authorization profile name specified

This example queries the document class and the permissions for the authorization profile with the ID 9. In two for-loops, all classes and their permissions are browsed and returned. If an error occurs, then the error number is returned.

The alternative function call of `roll_get_data` passing the name of the authorization profile without the ID is:

```
returnValue = api_function( "roll_get_data", "", h_RollName )
```

In this case, the variable `h_RollName` must hold the respective name. In the example code, the according variable declaration and the function call are commended out.

Note

```

// -----
// Beispiel:
// Abfrage von Benutzerprofildaten
// -----

// Variablendeklaration
vars h_ReturnValue      // Fehlercode
vars h_RollID = 9        // ID der Rolle
vars h_Count             // Zähler für die Schleife
vars h_CountRight        // Zähler für die Schleife
vars h_TextTemp          // temporäre Datei für die Ausgabe
vars h_Right[4]          // Beschreibung der Rechte

h_Right[1] = "Lese"       // Position des Leserechte-flag
h_Right[2] = "Schreib"    // Position des Schreibrechte-flag
h_Right[3] = "Prüf"       // Position des Prüfrechte-flag
h_Right[4] = "Freigabe"   // Position des Freigaberechte-flag

// Funktionsaufruf
h_ReturnValue = api_function( "roll_get_data", h_RollID )

// alternativer Aufruf mit Rollen-Name:
// vars h_RollName = "Mail"
// returnValue = api_function ( "roll_get_data", "", h_RollName)

// Ergebnisausgabe
if (h_ReturnValue == 0)
{
    // Durchgehen aller Klassen
    for h_Count = 1 while api_class_right[h_Count] != "" step 1
    {
        call api_log_info("Klasse :api_classes[h_Count]:")
        // Durchgehen aller 4 Rechtearten:
        for h_CountRight = 1 while h_CountRight <= 4 step 1
        {
            if (api_class_right[h_Count](h_CountRight, 1) == 0)
            {
                call api_log_info("Diese Klasse hat KEINE \
                                :h_Right[h_CountRight] Rechte ")
            }
            else if (api_class_right[h_Count](h_CountRight, 1) ==
1)
            {
                call api_log_info("Diese Klasse hat \
                                :h_Right[h_CountRight] Rechte ")
            }
            else
            {
                call api_log_info("Es gibt keine Angaben zu \
                                :h_Right[h_CountRight] Rechte ")
            }
        } // Ende for-loop: h_CountRight
    } // Ende for-loop: h_Count
}

```

```

} // Ende h_ReturnValue == 0
//
// Fehlerbehandlung
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
        zurückgeliefert! ")
}

```

1.4.15. Restriction sets

d3set_add_filter

Adds a filter to an existing restriction set.

If the restriction set assignment (set_name, object_id) does not exist, then this is created.

Parameters	Description
user_name	d.3 user name in whose name the change should be performed.
doc_id	Document ID of the restriction set document. If it is set, then set_name and object_id can remain empty.
set_name	Name of the restriction set
object_id	User name or group ID. If empty, then a set without assignment is searched.
filter	Filter value(s) to be added to the set. Several filter values can be separated by a ";"-characters. Max. length of the filters is 150 characters.
overwrite	0: Filter is appended to the existing filter. 1: Previous filter is overwritten.

Return value	Meaning
0	everything OK
1505	Restriction set document not found.
1510	no access right to the set-document.
1512	No write right to the set document.
1504	Length of the set filter of the restriction set exceeds the maximum of 150.000 characters.
1513	The restriction set does not exist.
9000	Database error

d3set_remove_filter

Removes a filter from an existing restriction set.

Parameters	Description
user_name	d.3 user name in whose name the change should be performed.
doc_id	Document ID of the restriction set document. If it is set, then set_name and object_id can remain empty.
set_name	Name of the restriction set
object_id	User name or group ID. If empty, then a set without assignment is searched.
filter	Filter value(s) to be removed from the set. Several filter values can be separated by a ";"-character. Max. length of the filters is 150 characters.

Return value	Meaning
0	everything OK
1505	Restriction set-document not found.
1510	no access right to the set-document.
1512	no write permission to the set-document.
1504	Length of the set filter of the restriction set exceeds the maximum of 150.000 characters.

Return value	Meaning
9000	Database error

d3set_get_filter

Determines the filters of an existing restriction set.

The filter is returned in the context variable `api_doc_field_6x[...]`.

Parameters	Description
<code>user_name</code>	d.3 user name in whose name the change should be performed.
<code>doc_id</code>	Document ID of the restriction set document. If it is set, then <code>set_name</code> and <code>object_id</code> can remain empty.
<code>set_name</code>	Name of the restriction set
<code>object_id</code>	User name or group ID. If empty, then a set without assignment is searched.

Return value	Meaning
0	everything OK
1505	Restriction set document not found.
1510	no access right to the set-document.
1512	No write right to the set document.
1504	Length of the set filter of the restriction set exceeds the maximum of 150.000 characters.

The restriction set-assignment refers to a document in the d.3 in the document type `$SET`.

The 60-fields of this document will be used for the storage of the set search filter. The single attributes can be separated by a ";"-character.

`d3set_get_filter` fills the 60-fields in the `api_doc_fields`.

The filter of the restriction set assignment is stored in the 60-fields and can extend max. to all possible 60-fields (field 60, row 1 - field 69, row 100).

The filter is returned in:

- `api_doc_field_60[1]`
- `api_doc_field_60[2]`
- `api_doc_field_60[3]`
- ...
- `api_doc_field_60[100]`
- `api_doc_field_61[1]`
- ...
- `api_doc_field_61[100]`
- ...
- `api_doc_field_69[100]`

If the first of this fields is empty, there are no further entries. This means, if `api_doc_field_60[1]` has got a value, but `api_doc_field_60[2]` is empty, then no further values will follow.

d3set_get_sets

Note

New in d.3 version 8.0

Searches restriction set associations according to specified criteria and returns them in a table.

The name of the restriction set (`set_name`), an object (`object_id`: user, group, organizational unit) and the object class (`object_class_id`) can be used as search criteria.

If `object_id` is set, `object_class_id` must also be set.

Parameters	Description
<code>set_name</code>	Name of the restriction set
<code>object_id</code>	User name or group ID. If empty, then a set without assignment is searched.
<code>object_class_id</code>	Object-class (0: none object-assignment, 1: d.3 user, 2: d.3 group, 4: Organizational unit)

The number of returned results will be written into `api_single_info`. A maximum of 100 lines is logged.

The following arrays of the context will be filled:

Variable	Description
<code>api_set_doc_id</code>	Document ID of the restriction set document
<code>api_set_name</code>	Name of the restriction set
<code>api_set_object_id</code>	User name, group ID or organizational unit ID
<code>api_set_object_class_id</code>	Object-class ID of value from <code>api_set_object_id</code> (0: none object-assignment, 1: d.3 user, 2: d.3 group, 4: Organizational unit)

Return value	Meaning
0	everything OK
>0	Database error
-1	invalid parameter (see d.3 log file)

d3set_remove_set

Note

New in d.3 version

Deletes an existing restriction set. The following options are possible:

If only the parameter `set_name` is filled, then all references of this restriction set including the assignments within document classes are removed.

If the parameter `set_name` is filled in addition to `object_id`, then only the set is removed from that system, which is assigned to the object (user or group). The actual set or the reference to the existence of the set is maintained.

If the parameter `doc_id` is set, the specific set is removed. This parameter has priority to `set_name` and `object_id`.

Parameters	Description
<code>user_name</code>	d.3 user name in whose name the change should be performed; this d.3 user must have write rights to restriction sets.
<code>doc_id</code>	Document ID of the restriction set-document. If it is set, then <code>set_name</code> and <code>object_id</code> can remain empty.
<code>set_name</code>	Name of the restriction set
<code>object_id</code>	User name or group ID. If empty, then a set without assignment is searched.

Return value	Meaning
0	everything OK
3	Mandatory parameter not set

Return value	Meaning
110	Invalid Doc ID specified.
1506	The set document could not be removed (see d.3 log file)
1507	Object name already exists as user, group, activity profile or organizational unit.
1509	Restriction set is still referenced in a document class
1511	No permission for the deletion of restriction sets
1513	The restriction set does not exist.
9000	Database error
14020	Error while logging to the audit trail

1.4.16. TIFF-functions

document_render

`document_render` creates either a TIFF or PDF (document or file rendition).

Note

This function replaces the function `tiff_create`.

Difference: `tiff_create` only creates TIFF-files; `document_render` creates a TIFF or a PDF-file.

Note

available from d.3 version 6.1

Parameters	Description
<code>source</code>	Source file (mandatory), unless the document <code>doc_id</code> (latest version) shall be rendered.
<code>destination</code>	Target file (mandatory), if the rendered file is not to be stored as the dependent <code>t1</code> respectively <code>p1</code> file for <code>doc_id</code> .
<code>render_option</code>	0: No render file creation, e.g. only create the OCR file 1 or " ": Create TIFF-file as render file 2: Create PDF-file as render file 3: Create TIFF- and PDF-file as render files default: 1
<code>OCR</code>	Extract index keywords default: 0
<code>asynchronous</code>	Asynchronous TIFF conversion, i.e. only a rendition job is generated (<code>job_id</code> is supplied for <code>error_var[.] = "job_id"</code> ; this option is only available for the document-based TIFF conversion, i.e. <code>doc_id</code> must exist. default since 6.2: 1, previously 0 from 6.2 only the value 1 is supported
<code>replace_doc</code>	Generated TIFF replaces source document in d.3. (compare <code>ERLAUBE_STAMM_2_TIFF_KONVERT</code>) default: 0
<code>overwrite</code>	Replace target file, if it already exists default: 0
<code>doc_id</code>	Document to be converted to TIFF / for which a dependent <code>T1</code> or <code>P1</code> document is to be created. If nothing is specified, then the document ID from the context is used.
<code>user_name</code>	User in whose name the call is to be made. If nothing is specified, then the executive user is used.

Parameters	Description
status	Status of the version to be rendered. If nothing is specified, the current version is used ({B, Processing, P, Verification, F, Release, A, Archive})
archiv_index	Index for the version in the status archive to be rendered (only required for status="A"/"Archive").
prio	Valid values are low, normal, high If another value is passed, then always "normal" is used; is also available without executing upd721.jpl in 7.2.1.)

Return value	Meaning
0	everything OK
>0	Database error
-1	Panther syntax error, details see logviewer
-2	function was called incorrect
-3	unknown error, contact d.velop support
-101	neither document ID passed nor context set
-151	Source file name is not specified
-152	Target file name is not specified
-201	invalid user name specified
-301	invalid document ID specified or user has got no rights to it
-302	Error while determining the user rights
-303	User has got no access permission to document
-543	Error in the rendition service connection
-544	Rendition service is not available
-545	Documents in the status archive must not be rendered
-546	Placeholders of the bidirectional proxy must not be rendered
-547	DTIFF_SUPPORT is not enabled api_error_var[...] = "dtiff_support" api_error_value[...] = "dtiff_support"
-557	Target file already exists and shall not be overwritten
-558	For asynchronous=1, the length of path and name must not exceed 100 characters
-587	Document could not be retrieved from secondary storage
-588	Documents in the processing of the d.3 server cannot be rendered
-589	The document status has changed during the process

tiff_create

Note

tiff_create only creates TIFF-files

You should use the new function [document_render](#) which creates TIFF or PDF-files.

This functions converts a file or document to TIFF.

Parameters	Description
source	Source file (mandatory), unless the document doc_id (latest version) shall be rendered.
destination	Target file (mandatory), if the rendered TIFF-file is not to be stored as the dependent T1 file for doc_id.
TIFF	Create a TIFF-file (default: 1)
OCR	Extract index keywords. (default: 0)

Parameters	Description
<code>asynchronous</code>	Asynchronous TIFF conversion, i.e. only a rendition job is generated (<code>job_id</code> is supplied for <code>error_var[...]</code> = " <code>job_id</code> "; this option is only available for the document-based TIFF conversion, i.e. <code>doc_id</code> must exist. default: 0
<code>replace_doc</code>	Generated TIFF replaces source document in d.3 (see also <code>ERLAUBE_STAMM_2_TIFF_KONVERT</code>) default: 0
<code>overwrite</code>	Replace target file, if it already exists default: 0
<code>doc_id</code>	Document to be converted to TIFF / for which a dependent T1 document shall be created. If nothing is specified, then the document ID from the context is used.
<code>user_name</code>	User in whose name the call is to be made. If nothing is specified, then the executive user is used.

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-3	undefined error, contact d.velop support
-101	either document ID passed nor set in the context
-151	Source file name is not specified
-152	Target file name is not specified
-201	invalid user name specified
-301	invalid document ID specified or user has got no rights to it
-302	Error while determining the user rights
-303	User has got no access permission to document
-543	Error in the d.tiff connection
-544	d.tiff is not available
-545	Documents in the status archive must not be rendered
-546	Placeholders of the bidirectional proxy must not be rendered
-547	DTIFF_SUPPORT is not enabled <code>api_error_var[...] = "dtiff_support"</code> <code>api_error_value[...] = "dtiff_support"</code>
-557	Target file already exists and shall not be overwritten
-558	For <code>asynchronous=1</code> , the length of path and name must not exceed 100 characters
<-9500	Error in customer-specific hook-function <code>hook_rendition_...</code>

Note

available since d.3 version 6.0

tiff_concat

The function appends a TIFF document to another one.

Parameters	Description
<code>source</code>	Source TIFF-file
<code>destination</code>	Target TIFF-file
<code>source_rdl</code>	Source redlining file (optional)
<code>destination_rdl</code>	Target redlining file (optional)

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-3	unknown error, contact d.velop support
-151	Source file name is not specified
-152	Target file name is not specified
-559	Error while attaching the TIFF-files
-560	Count of target pages uneven the sum of the source pages
-561	Source TIFF-file does not exist
-562	Source TIFF-file is not supported (possibly corrupt or wrong format)
-563	Target TIFF-file cannot be opened
-564	Source redlining file is not supported (possibly corrupt or wrong format)
-565	Source redlining file contains not assigned pages

document_render_wfl_prot

Renders the workflow protocols to a document ID and creates a dependent `w1` file.

If a `w1` file already exists, the function executes no action (return 1). If a new TIFF-file shall be rendered because e.g. a further protocol for the document ID was added, then the dependent `w1` must be removed at first.

Parameters	Description
<code>doc_id (P)</code>	Document whose workflow protocols shall be rendered
<code>user_name</code>	Name of the executing user

Return value	Meaning
1	rendered workflow protocol already exists for this document ID
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-630	Temporary collection file for all workflow-protocols for this document ID could not be created (see d.3 logviewer)
-631	Temporary collection file for all workflow-protocols for this document ID could not be successfully completed (see d.3 logviewer)
-632	A workflow-protocols for this document ID could not be found (see d.3 logviewer)

1.4.17. File access

file_add_string

This function writes a string into a file. If the file does not exist in the specified directory, then it will be created. If the file exists, then the string can either be appended or the file will be overwritten.

Note

changed from 7.1.0 HF5:

The global variable `UTF8_BOM` allows to write a UTF8 Byte Order Mark in front of a file.

It was not easily possible to add a UTF8 Byte-Order-Mark at the beginning of a text-file written by the server-API-functions `file_add_line()` or `file_add_string()`. To do so, the global variable `UTF8_BOM` can now be used.

The BOM can now be written to the beginning of a file with call `api_function('file_add_string', file_name, UTF8_BOM, 'OVERWRITE')` before the actual content of the file is written in 'APPEND'-mode. The developer must make sure that all following strings are passed to the API-function as UTF8.

Parameters	Description
<code>file_name</code>	Name and path of the file to be written to.
<code>text</code>	String to be written to the file.
<code>mode</code>	Mode of the file operation: APPEND: If the file exists, then the text will be appended (default) OVERWRITE: If the file exists, then it will be overwritten. If no value is passed, then APPEND will be used.
<code>count</code>	Number of characters to be minimally written. If <code><text></code> is shorter, then spaces are appended.

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-3	unknown error, contact d.velop support
-150	File name is not specified
-210	invalid mode specified
-552	File cannot be opened for overwriting
-553	File cannot be opened for appending

This example appends the file `c:\text.txt` with a text "I, this text, am written to this file." If an error occurs, then the error number is returned.

Note

```

/ -----
// Beispiel:
// Zeichenkette in Datei schreiben
// -----

// Variablendeklaration
vars h_ResultValue    // Fehlercode

// Im Pfad müssen ":" und "\" mit vorgestelltem \ maskiert
werden.
vars h_Pfad = "C:\\\\text.txt"
vars h_Text = "Ich, dieser Text, werde in die Datei geschrieben.
"

// Funktionsaufruf
h_ResultValue = api_function ("file_add_string", h_Pfad, h_Text)

// Fehlerbehandlung
if (h_ResultValue != 0)
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
        zurückgeliefert! ")
}

```

file_add_line

This function writes into a new line of the specified file. If the file does not exist in the specified directory, then it will be created. If the file exists, then the string can either be appended or the file is overwritten.

Note

changed from 7.1.0 HF5:

The global variable `UTF8_BOM` allows to write a UTF8 Byte Order Mark in front of a file.

It was not easily possible to add a UTF8 Byte-Order-Mark at the beginning of a text-file written by the server-API-functions `file_add_line()` or `file_add_string()`. To do so, the global variable `UTF8_BOM` can now be used.

The BOM can now be written to the beginning of a file with `call api_function('file_add_string', file_name, UTF8_BOM, 'OVERWRITE')` before the actual content of the file is written in 'APPEND'-mode.

The developer must make sure that all following strings are passed to the API-function as UTF8.

Parameters	Description
<code>file_name</code>	Name and path of the file to be written to.
<code>line</code>	Line to be written to the file.

Parameters	Description
mode	Mode of the file operation: APPEND: If the file exists, then the text will be appended (default) OVERWRITE: If the file exists, then it will be overwritten. If no value is passed, then APPEND will be used.
count	Number of characters to be minimally written. If <line> is shorter, then spaces are appended.

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-3	undefined error, contact d.velop support
-150	File name is not specified
-210	invalid mode specified
-552	File cannot be opened for overwriting
-553	File cannot be opened for appending

In this example a new line with the string content of the variable

`h_Line` will be written into the file `C:\text.txt`. If an error occurs, then the error number is returned.

Note

```
// -----
// Beispiel:
// Zeichenkette in eine bestimmte Zeile einer Datei schreiben
// -----

// Variablendeklaration
vars h_ReturnValue // Fehlercode

// Im Pfad müssen ":" und "\" mit vorgestelltem \ ergänzt werden.
vars h_Pfad = "c:\\text.txt"
vars h_Line = "Ich, dieser Text, wurde in die Datei geschrieben.
"

// Funktionsaufruf
h_ReturnValue = api_function ("file_add_line", h_Pfad, h_Line)

// Fehlerbehandlung
if (h_ReturnValue != 0)
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
        zurückgeliefert! ")
}
```

file_read_line

This function reads a line from a file. The result is stored in the variable `api_single_info`. If the end of file is before the line to be read, then `api_single_info` contains the string "EOF".

Parameters	Description
file_name	File name
line	Start line
mode	Reading direction: BOF=forward, EOF=backward.
count	Maximum number of characters to be read.

Return value	Meaning
>=0	everything OK; is equivalent to the number of characters found.
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-3	undefined error, contact d.velop support
-150	File name is not specified
-210	Invalid reading direction specified.
-550	File cannot be opened for reading

Note

This function is not suitable (currently) to read and process large files. If e.g. line 10000 of a file shall be read, so currently (internal) all 9999 lines before will be read, too, because you cannot jump directly to line 10000.

For large files it is recommended to access to the Panther-functions directly (unless only the of the first lines must be read). With one of the next server versions a improved reading of files could be possible.

file_concat

This function append the content of a file to another one.

Parameters	Description
source	Source file whose content is to be added
destination	Target file which is to be appended

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-3	unknown error, contact d.velop support
-151	Source file name is not specified
-152	Target file name is not specified
-550	File cannot be opened for reading
-551	File cannot be opened for writing
-554	Error while writing the file

file_copy

This function copies a file.

Parameters	Description
source	Source data
destination	Target file

Return value	Meaning
0	everything OK

Return value	Meaning
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-3	undefined error, contact d.velop support
-151	Source file name is not specified
-152	Target file name is not specified
-504	Error while allocating the memory
-550	File cannot be opened for reading
-551	File cannot be opened for writing
-554	Error while writing the file

In this example, the file `C:\text.txt` will be copied. If the file `C:\text2.txt` does not exist, then it is created with the content of `C:\text.txt`. If it does exist, however, then it is overwritten with the content of `C:\text.txt`. If an error occurs, then the error number is returned.

Note

```
// -----
// Beispiel: Datei kopieren
// -----
// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_Quelle = "c:\\text.txt" // Ausgangs-Datei
vars h_Ziel = "c:\\text2.txt"  // Ziel-Datei

// Funktionsaufruf
h_ReturnValue = api_function ("file_copy", h_Quelle, h_Ziel)
// Fehlerbehandlung
if (h_ReturnValue != 0)
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
        zurückgeliefert! ")
}
```

file_delete

This function deletes a file.

Parameters	Description
<code>file_name</code>	Path of the file to be deleted

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-3	undefined error, contact d.velop support
-150	File name is not specified
-250	File not found
-350	Access denied

This example deletes the file in `C:\text2.txt`. If an error occurs, then the error number is returned.

Note

```
// -----
// Beispiel: Löschen einer Datei
// -----
// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_Pfad = "c:\\text2.txt" // Dateipfad
// Funktionsaufruf
h_ReturnValue = api_function ("file_delete", h_Pfad)
if (h_ReturnValue != 0)
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
        zurückgeliefert! ")
}
```

file_compare

This function compares two files.

Parameters	Description
file_name1	File 1
file_name2	File 2

Return value	Meaning
4	File content is different
3	File size is different
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-3	undefined error, contact d.velop support
-153	File name 1 is not specified
-154	File name 2 is not specified
-504	Error while allocating the memory
-550	File cannot be opened for reading

This example compares the two files `C:\text.txt` and `C:\text2.txt` to each other and returns the result. If an error occurs, then the error number is returned.

Note

```

/ -----
// Beispiel:
// Vergleich von Dateien
// -----

// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_Datei1 = "c:\\text.txt" // Datei 1
vars h_Datei2 = "c:\\text2.txt" // Datei 2

// Funktionsaufruf
h_ReturnValue = api_function ("file_compare", h_Datei1, h_Datei2)

// Ergebnisausgabe
if (h_ReturnValue == 4)
{
    call api_log_info("Die Dateien haben einen unterschiedlichen
        Inhalt")
}
if (h_ReturnValue == 3)
{
    call api_log_info("Die Dateien haben unterschiedliche
        Dateigrößen")
}
if (h_ReturnValue == 0)
{
    call api_log_info("Die Dateien sind identisch")
}
// Fehlerbehandlung
if (h_ReturnValue < 0)
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
        zurückgeliefert! ")
}

```

file_rename

This function renames a file.

Parameters	Description
source	old file name
destination	new file name

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-3	undefined error, contact d.velop support
-151	Source file name is not specified
-152	Target file name is not specified
-555	Error while renaming the file

This example renames the file `C:\text.txt` to `text2.txt`. If an error occurs, then the error number is returned.

Note

```
// -----
// Beispiel:
// Umbenennen einer Datei
// -----
// Variablendeklaration
vars h_ReturnValue // Fehlercode
vars h_AlterName = "c:\\text.txt" // Alter Dateiname
vars h_NeuerName = "c:\\text2.txt" // Neuer Dateiname
// Funktionsaufruf
h_ReturnValue = api_function ("file_rename", h_AlterName,
h_NeuerName)

// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    call api_log_info("Datei erfolgreich umbenannt. ")
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
    zurückgeliefert! ")
}
```

file_exist

This function checks, if a file exists.

Parameters	Description
file_name	File name

Return value	Meaning
1	File exists
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-3	undefined error, contact d.velop support
-150	File name is not specified

This example checks, if the file `C:\text.txt` already exists. The result is displayed. If an error occurs, then the error number is returned.

Note

```

/ -----
// Beispiel:
// Prüfen, ob Datei existiert
// -----
// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_Pfad = "c:\\text.txt" // Pfad zur Datei
// Funktionsaufruf
h_ReturnValue = api_function ("file_exist", h_Pfad)
// Ergebnisausgabe
if (h_ReturnValue == 1)
{
    call api_log_info("Die Datei existiert. ")
}
if (h_ReturnValue == 0)
{
    call api_log_info("Die Datei existiert nicht. ")
}
// Fehlerbehandlung
if (h_ReturnValue < 0)
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
        zurückgeliefert! ")
}

```

1.4.18. String conversion**string_to_lower**

This function converts a string to lower case letters. The converted string will be written into the global variable `api_converted_string`.

Note

From version 6.2: Strings of any code page can be converted.

Parameters	Description
text	String to be converted
code_page	Code page of the passed string to be converted to lower case (optional; default "1252", from version 6.2)

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect

In this example, the string "Alpha Bravo Charly Delta Echo Foxtrott" is converted to the string "alpha bravo charly delta echo foxtrott". If an error occurs, then the error number is returned.

Note

```
// -----
// Beispiel:
// Zeichenkette in Kleinbuchstaben konvertieren
// -----
// Variablendeklaration
vars h_ReturnValue
           // Fehlercode
vars h_Text = "Anton Berta Caesar Dora Emil"
           // Ursprünglicher String
// Funktionsaufruf
h_ReturnValue = api_function ("string_to_lower", h_Text)
// Ergebnisausgabe
if (h_ReturnValue == 0)
{
    call api_log_info("Konvertierter String:
:api_converted_string")
}
// Fehlerbehandlung
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
zurückgeliefert! ")
}
```

string_to_upper

This function converts a string to upper case letters. The converted string will be written into the global variable `api_converted_string`.

Parameters	Description
text	String to be converted
code_page	Code page of the passed string to be converted to upper case (optional; default "1252", from version 6.2)

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect

In this example, the string "Alpha Bravo Charly Delta Echo Foxtrott" is converted to the string "ALPHA BRAVO CHARLY DELTA ECHO FOXTROTT". If an error occurs, then the error number is returned.

Note

```
// -----
// Beispiel:
// Zeichenkette in Grossbuchstaben konvertieren
// -----

// Variablendeklaration
vars h_ReturnValue
    // Fehlercode
vars h_Text = "Anton Berta Caesar Dora Emil"
    // Ursprünglicher String

// Funktionsaufruf
h_ReturnValue = api_function ("string_to_upper", h_Text)

// Ergebnisausgabe
if (h_ReturnValue == 0)
{
    call api_log_info("Konvertierter String:
:api_converted_string")
}
// Fehlerbehandlung
else
{
    call api_log_info("Fehlernummer :h_ReturnValue wurde
zurückgeliefert! ")
}
```

string_to_number

This function converts a string into the JPL numeric format, i.e. without thousands delimiter and a with a decimal point ".". The converted string will be written into the global variable `api_converted_string`.

Parameters	Description
text	String to be converted
data_type	Valid data types M=MONEY and N=NUMERIC; this is equivalent to the respective d.3 data type and is required for the determination of the thousands delimiter. If nothing is specified, then NUMERIC is used as default.

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-206	invalid data type specified

string_to_date

This function formats a string in the d.3 default-date-format. The converted string will be written into the global variable `api_converted_string`. If the `text` does not contain a proper date, then a null string is written into the global variable.

Parameters	Description
text	String to be converted

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-208	Invalid date specified

The following example code converts the date of the string into the d.3 date format. If an error occurs, then the error number is returned.

Note

```
// -----
// Beispiel:
// Umwandeln eines String-Datums in ein d.3-Datum
// -----
// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_Datum = "01.01.1972" // Ursprungsdatum als String
// Funktionsaufruf
h_ReturnValue = api_function ("string_to_date", h_Datum)
// Ergebnisausgabe
if (h_ReturnValue == 0)
{
    call api_log_info("Konvertiertes Datum:
:api_converted_string")
}
// Fehlerbehandlung
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
zurückgeliefert! ")
}
```

string_split

This function splits a string at a specified separator. The part strings are stored in the global array `apiSplittedString[]`.

Note

If no error occurred, then the number of determined values is returned by the function and is NOT stored in the variable `apiSingleInfo`. If an error occurs, then the error number is returned.

Parameters	Description
text	String to be split
separator	(Character)string, at which the split should be executed

Return value	Meaning
>0	Number of values in the array.

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect

In this example, the string "Alpha|Bravo|Charly|Delta|Echo" is divided into the five strings "Alpha", "Bravo", "Charlie", "Delta", and "Echo". These are displayed. If an error occurs, then the error number is returned.

Note

```
// -----
// Beispiel:
// Zeichenkette aufspalten
// -----
// Variablendeklaration
vars h_ReturnValue
    // Anzahl zurückgegebener Werte
vars h_String = "Anton|Berta|Caesar|Dora|Emil"
    // zu teilender String
vars h_Splitter = "|"
    // Trenn-Zeichen
vars h_Count
    // Zähler für Schleife
// Funktionsaufruf
h_ReturnValue = api_function ("string_split", h_String,
h_Splitter)

// Ergebnisausgabe
if (h_ReturnValue >= 0)
{
    call api_log_info("Der String wurde in :h_ReturnValue
Strings geteilt.")
    for h_Count = 1 while( h_Count <= h_ReturnValue ) step 1
    {
        call api_log_info("Teilstring: :(apiSplittedString
[:h_Count])")
    }
}
// Fehlerausgabe
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
zurückgeliefert!")
}
```

string_cut_leading_char

This function removes the leading characters from a string. The converted string is stored in the global variable `api_converted_string`.

Note

If no error occurred, then the number of determined values is returned by the function and is NOT stored in the variable `api_single_info`. If an error occurs, then the error number is returned.

Parameters	Description
text	String whose leading characters are to be removed.
Character	Character to be removed (default: space)

Return value	Meaning
>=0	Count of removed characters
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect

In this example, the string "AAAAAABCDEFGF" is truncated of the leading "A"s.

The new string "BCDEFG" is stored in the global variable `api_converted_string` and is displayed.

Note

```
// -----
// Beispiel:
// Entfernen führender Zeichen
// -----

// Variablendeklaration
vars h_ReturnValue           // Fehlercode
vars h_Text = "AAAAAABCDEFGF" // Ursprünglicher String
vars h_Char = "A"            // Abzutrennendes Anfangszeichen

// Funktionsaufruf
h_ReturnValue = api_function ("string_cut_leading_char", h_Text,
                             h_Char)

// Ergebnisausgabe
if (h_ReturnValue >= 0)
{
    call api_log_info("Konvertierter String:
:api_converted_string")
    call api_log_info("Es wurden :h_ReturnValue Zeichen entfernt.
")
}
// Fehlerausgabe
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
zurückgeliefert! ")
}
```

string_fill_leading_char

This function adds leading characters to a string to reach a full string length. The converted string is stored in the global variable `api_converted_string`. If an error occurs, then the error number is returned.

Parameters	Description
text	String to be filled up.
Character	Character to be inserted. (default: space)
length	Target length of the string. (standard: 0)

Return value	Meaning
>=0	Count of inserted characters
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect

This example inserts the characters "0" before the original string "12345", until the string has a length of 10 characters. The new string and the number of inserted characters are displayed. If an error occurs, then the error number is returned.

Note

```
// -----
// Beispiel:
// Anfügen von Zeichen
// -----

// Variablendeklaration
vars h_ReturnValue      // Fehlercode
vars h_Text = "12345"   // Ursprünglicher String
vars h_Char = "0"       // Zeichen, welches eingefügt wird
vars h_Length = 10      // Ziel-Länge des neuen Strings

// Funktionsaufruf
h_ReturnValue = api_function ("string_fill_leading_char", \
                             h_Text, h_Char, h_Length)

// Ergebnisausgabe
if (h_ReturnValue >= 0)
{
    call api_log_info("Konvertierter String:
:api_converted_string")
    call api_log_info("Es wurden :h_ReturnValue Zeichen
eingefügt")
}
// Fehlerausgabe
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
zurückgeliefert! ")
}
```

string_fill_trailing_char

This function appends trailing characters (standard: space) to a string to reach a target length or the string.

The converted string is stored in the global variable `api_converted_string`. This string is one "|" character longer than length, since trailing spaces as fillers are removed by Panther.

Parameters	Description
text	String to be appended.
Character	Character to be inserted. (default: space)
length	Target length of the string. (standard: 0)

Return value	Meaning
>=0	Count of inserted characters
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect

In this example, spaces are appended to the string "AlphaBravoCharlie" until the string has a length of 25 characters. Additionally, a 26th character "|" is appended to prevent JPL from removing the trailing spaces.

Since the function call does not specify a parameter for the character to be inserted, the passed string is filled up with spaces.

Afterwards, the new string and the number of inserted spaces is returned. If an error occurs, then the error number is returned.

Note

```
// -----
// Beispiel:
// Anhängen von Zeichen
// -----

// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_Text = "AntonBertaCaesar" // Ursprünglicher String
vars h_Length = 25          // Ziel-Länge des neuen
Strings

// Funktionsaufruf
// ohne Angabe des Füll-Zeichens, daher wird das Leerzeichen
// verwendet
h_ReturnValue = api_function ("string_fill_trailing_char",
                             h_Text, \
                             "", h_Length)

// Ergebnisausgabe
if (h_ReturnValue >= 0)
{
    call api_log_info("Konvertierter String:
:api_converted_string")
    call api_log_info("Es wurden :h_ReturnValue Zeichen
angehängt. ")
}
// Fehlerausgabe
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
zurückgeliefert! ")
}
```

string_replace

This function replaces all occurrences of one string in a second string by a third one. The converted string is stored in the global variable `api_converted_string`.

Parameters	Description
text	String in which the substitution is to be performed.
old_str	String to be replaced. (standard: space)
new_str	New string

Return value	Meaning
>=0	Count of inserted characters
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-107	Text is not specified

In this example, all occurrences of the string "Alpha" are replaced by "Foxtrott" in the original string.

Afterwards, the new string and the number of substituted occurrences is displayed. If an error occurs, then the error number is returned.

Note

```
// -----
// Beispiel:
// Ersetzen von Zeichen(ketten) innerhalb von Zeichenketten
// -----

// Variablen Deklaration
vars h_ReturnValue                                // Fehlercode
vars h_Text = "Anton Berta Anton Ceasar Anton Dora" // Ursprüngliche
                                                    // Zeichenkette

vars h_TextToReplace = "Anton"                    // Zu ersetzender Text
vars h_TextByReplace = "Friedrich"                // Einzusetzender Text

// Funktionsaufruf
h_ReturnValue = api_function ("string_replace", h_Text, \
                                h_TextToReplace,
                                \
                                h_TextByReplace )

// Ergebnisausgabe
if (h_ReturnValue >= 0)
{
    call api_log_info("Konvertierter String:
:api_converted_string")
    call api_log_info("Es wurden :h_ReturnValue \
Zeichenkette ausgetauscht")
}
// Fehlerausgabe
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
zurückgeliefert! ")
}
```

string_get_position

This function returns the position of one string in another one.

Parameters	Description
text	String which is to be analysed.
sub_str	String to be found (standard: space).
direction	Search direction. (1: forwards, -1: backward, default: 1)
start	Character from which to search (1 = 1. character) (standard: 1 for direction=1;@length(text) for direction=-1)

Return value	Meaning
0	not found
>0	first found position (1. character=1)
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-3	undefined error, contact d.velop support
-107	Text is not specified

In this example, the original string is searched for the string "Charlie". Afterwards, it is displayed if and where the string "Charlie" was found for the first time.

If an error occurs, then the error number is returned.

Note

```
// -----
// Beispiel:
// Suchen innerhalb von Zeichenketten
// -----
// Variablendeklaration
vars h_ReturnValue // Fehlercode
vars h_Text = "Anton Berta Caesar Dora Emil Friedrich Gustav" // Ursprüngliche Zeichenkette
vars h_TextToFind = "Caesar" // Text, der gesucht werden soll
vars h_SearchDirection = 1 // Suchrichtung
vars h_CharWhereToStart // Zeichen, ab dem gesucht wird
// Funktionsaufruf
h_ReturnValue = \
api_function ("string_get_position", h_Text, \
                                         h_TextToFind, \
                                         h_SearchDirection, \
                                         h_CharWhereToStart)

// Ergebnisausgabe
if( h_ReturnValue > 0 )
{
    call api_log_info("Die Zeichenkette wurde an Position \
                      :h_ReturnValue) gefunden. ")
}
else if (h_ReturnValue == 0)
{
    call api_log_info("Die Zeichenkette wurde NICHT gefunden. ")
}
// Fehlerausgabe
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
zurückgeliefert! ")
}
```

string_mask_special_char

This function masks the JAM-special characters of a string.

The masked string is stored in the global variable `api_converted_string`.

Parameters	Description
text	String to be masked.

Return value	Meaning
0	not found
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect

api_str_sub

Note

available from d.3 version 7.0.1

Note

The functions described here are inline-functions which do NOT have to be encapsulated via `call api_function()`. Instead they directly return the result.

Equivalent to native JPL-functions they return '-1' on error.

Returns a part string of the passed string. In contrast to the Substring-operator of JPL this function correctly treats multi-byte-characters on unicode-systems. Thus no characters are split nor counted more than once.

This function expects UTF8-strings on unicode-systems and ANSI-strings on ANSI-systems and thus this function is usable in portable code.

Parameters	Description
<code>inString</code>	the original character string
<code>startIdx</code>	the index of the first character to occur in the output (starting with index 1). negative numbers reference positions relative to the end of the string (-1 is the last character)
<code>charCount</code>	Number of characters to be adopted in the output. if this parameter is not passed, then all characters from <code>startIdx</code> up to the end are returned. Negative numbers reference positions relative to the end of the string here, too.

Return value	Meaning
	the requested part string.
-1	Panther syntax error, details see d.3 logviewer

api_str_sub_utf8

This function behaves equivalent to `api_str_sub()` with the exception that it also expects strings in UTF-8 as the input on ANSI-systems.

api_str_sub_ansi

This function behaves equivalent to `api_str_sub()` with the exception that it also expects strings in ANSI as the input on unicode-systems.

api_str_len

Returns the number of occurrences of a character in a given string.

On unicode-systems the character-number can be lower than the byte number. The rule is

```
api_str_len(X) <= @length(X)
```

This function expects UTF8-strings on unicode-systems and ANSI-strings on ANSI-systems and thus this function is usable in portable code.

Parameters	Description
<code>inString</code>	the character string

Return value	Meaning
>0	the number of characters in the passed string.
-1	Panther syntax error, details see d.3 logviewer

api_str_len_utf8

This function behaves equivalent to `api_str_len()` with the exception that it also expects strings in UTF-8 as the input on ANSI-systems.

api_str_len_ansi

This function behaves like `api_str_len()` with the exception that strings in ANSI are also expected as input on Unicode systems.

api_str_to_utf8

Convert a string to the UTF8-format

Parameters	Description
<code>ansiInString</code>	String to be converted to UTF8.
<code>numCodepage</code>	Numeric ID of the destination-codepage (1250-1258). If this parameter is not specified, then the <code>DEFAULT_CODEPAGE</code> is assumed.

Return value	Meaning
	The UTF-8-representation of the passed string.
-1	Panther syntax error, details see d.3 logviewer

api_str_from_utf8

Note

The functions described here are inline-functions which do NOT have to be encapsulated via `call api_function()`. Instead they directly return the result. Equivalent to native JPL-functions they return '-1' on error.

Convert a UTF-8-string to ANSI.

Parameters	Description
<code>utf8InString</code>	String in the UTF-8-format.
<code>numCodepage</code>	Numeric ID of the destination-codepage (1250-1258). If this parameter is not specified, then the <code>DEFAULT_CODEPAGE</code> is assumed.

Return value	Meaning
	The ANSI-representation of the passed string.
-1	Panther syntax error, details see d.3 logviewer

api_str_is_utf8

Specifies, if a string in the UTF-8-format is valid.

Parameters	Description
<code>inString</code>	The string to be validated.

Return value	Meaning
1	The string is valid in the UTF8-format.
0	The string is invalid in the UTF8-format.

api_str_is_ascii

Specifies, if a string exclusively contains ASCII-characters (byte values between 1 and 127). Strings only containing ASCII-characters can be used as ANSI- and UTF-8-strings without conversion.

Parameters	Description
<code>inString</code>	the string to be validated.

Return value	Meaning
1	The string only contains ASCII-characters.
0	The string does not only contain ASCII-characters.

api_str_cut_fract_utf8

This function removes a split multi-byte-character from the end of a UTF-8-encoded string. This function can be used to convert values which were provided by a third-party system and possibly arbitrarily truncated to valid UTF-8-values again.

Parameters	Description
<code>inString</code>	Original string (this must be a valid UTF-8 except for a possible split character at the end).

Return value	Meaning
	If there is no split character at the end of the entry, then this is returned unchanged. The string in the return of this function can be up to 3 bytes shorter than the entry.
-1	Panther syntax error, details see d.3 logviewer

api_str_hash

Note

available from d.3 version 7.1.1

Note

The functions described here are inline-functions which do NOT have to be encapsulated via call `api_function()`. Instead they directly return the result. Equivalent to native JPL-functions they return '-1' on error.

This function creates hash-values on strings.

Parameters	Description
<code>hash_algorithm</code>	Valid values are: • MD5 • SHA-1 • SHA-224 • SHA-256 • SHA-384 • SHA-512 • RIPEMD-128 • RIPEMD-160 • RIPEMD-256 • RIPEMD-320 Warning MD5 and SHA-1 should no longer be used in security-relevant scenarios.
<code>value</code>	The string for which a hash-value is to be created.

Parameters	Description						
salt_level (0)	This parameters allows to specify that d.3 automatically applies a salting of the hash. <table border="1"> <tr> <td>0</td><td>If this is enabled, only the passed strings is calculated into the string. In this mode, the result with the hash-calculation is compatible with other applications.</td></tr> <tr> <td>1</td><td>A fixed salt implemented in d.3 is calculated into the hash-value. In this mode, all d.3 installations create the same hash value for the same passed string (value).</td></tr> <tr> <td>2</td><td>A salt, which is specific for the respective d.3 installation is calculated into the hash-value. This means in this mode, two d.3 installations return the same result only, if the database of the one installation was copied for the second installation and the same archive ID is used when accessing the d.3 gateway.</td></tr> </table> Note To realize an individual salting-method (for example to calculate a user anme into the password-hash), then the custom-salt can be directly appended to the <code>value</code>-parameter. This method can obviously be combined with the d.3 salt-modes.	0	If this is enabled, only the passed strings is calculated into the string. In this mode, the result with the hash-calculation is compatible with other applications.	1	A fixed salt implemented in d.3 is calculated into the hash-value. In this mode, all d.3 installations create the same hash value for the same passed string (value).	2	A salt, which is specific for the respective d.3 installation is calculated into the hash-value. This means in this mode, two d.3 installations return the same result only, if the database of the one installation was copied for the second installation and the same archive ID is used when accessing the d.3 gateway.
0	If this is enabled, only the passed strings is calculated into the string. In this mode, the result with the hash-calculation is compatible with other applications.						
1	A fixed salt implemented in d.3 is calculated into the hash-value. In this mode, all d.3 installations create the same hash value for the same passed string (value).						
2	A salt, which is specific for the respective d.3 installation is calculated into the hash-value. This means in this mode, two d.3 installations return the same result only, if the database of the one installation was copied for the second installation and the same archive ID is used when accessing the d.3 gateway.						

Note

For additional information on the subject salting see [http://en.wikipedia.org/wiki/Salt_\(cryptography\)](http://en.wikipedia.org/wiki/Salt_(cryptography)).

http://de.wikipedia.org/wiki/Salt_%28Kryptologie%29

Return value	Meaning
	On success, the respective hash-value is returned in hexadecimal encoding (0-9 and a-f) without separators.
-1	Panther syntax error, details see d.3 logviewer
	For example, passing an invalid algorithm ID.

1.4.19. Miscellaneous

restore_from_jukebox

This function restores files from the jukebox.

Parameters	Description
status	Status of the version to be restored. { "Freigabe" (Release), "Archiv" (Archive) }
archiv_index	Index of the version to be restored.
doc_id	ID of the document. If nothing is specified, the ID of the context is used.
user_name	User, in whose name the action shall be performed. If nothing is specified, then the executive user will be used.

Return value	Meaning
0	everything OK
>0	Database error
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-3	undefined error, contact d.velop support
-101	either document ID passed nor set in the context
-126	Invalid status specified
-201	invalid user name specified
-223	Invalid status; "Freigabe" or "Archiv" expected.
-301	invalid document ID specified or user has got no rights to it
-302	Error while determining the user rights

Return value	Meaning
-303	User has got no access permission to document api_error_var[...] = "doc_permission" api_error_value[...] = "<right>"
-310	User does not have the permission to see old archive versions. api_error_var[...] = "doc_permission" api_error_value[...] = "<right>"
-506	There is no release version. api_error_var[...] = "doc_permission" api_error_value[...] = "<right>"
-531	There is no version in the status Archive. api_error_var[...] = "doc_permission" api_error_value[...] = "<right>"
-532	Localisation of the document <\:doc_id> in status <\: status> cannot be determined.
-533	Document is not exclusively on the storage system, but also on the hard disk; restoring is not necessary. api_error_var[...] = "localisation" api_error_value[...] = "<localisation>"
-534	the jukebox did not provide the file
-556	Error while copying the file

restore_from_history

This function restores a deleted document from the history.

Parameters	Description
aktion_id	The value of the parameter has been ignored since d.3 server version 8.0.
doc_id	ID of the document. If nothing is specified, the ID of the context is used.
user_name	User, in whose name the action shall be performed. If nothing is specified, then the executive user will be used.
version_id	Version ID of the version to be restored. Only relevant if the entire document is not deleted. If a -1 is specified, the last version deleted from the Release status is restored, if available.

Return value	Meaning
0	everything OK
>0	Database error
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-101	neither document ID passed nor context set
-535	No history entry found for this document and this action ID
-536	Action was no deletion; document cannot be restored
-537	Error while copying an entry from the history table to the permanent one
-538	Error while deleting an entry from the history table
-539	Error while writing the document flags
<-9500	Error in the customer specific hook function hook_insert_exit_d3_20

send_email

This function allows to send an e-mail.

Warning

Sending to address arrays (see below) does not work with a date in the future (parameter date)

Parameters	Description
recipient	Recipient of the e-mail (d.3 user name or (from 6.2) d.3 group name)
notice	Subject text
Date	Send date (also in the future) optional
doc_id	ID of the document to be sent as a link optional If nothing is specified, the ID of the context is used.
user_name	Sender (d.3 user name); if nothing is specified, then the executing user is used.
body_file	Name and path of a file that contains the body text
mail_format	html: HTML-format, else: text-format (default)
attach	Send document as attachment (0(default) / 1).
doc_status	Attach version in status doc_status.
archive_index	Index for the archive version to be attached.
attach_abh	List of the dependent files (e.g. "T1, P1")
attach_file	List of the files to be attached (e.g. "C:\file1.jpg,c:\file2.txt")
use_recip_array	1: instead of recipient use api_email_recipients as recipient (-> up to 100 recipients possible) default: 0
use_cc_array	1: send a copy of the email e-mail to api_email_cc_recipients default: 0
use_bcc_array	1: send a blind copy to api_email_bcc_recipients default: 0

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-3	undefined error, contact d.velop support
-110	no recipient specified
-201	invalid user name specified
-301	invalid document ID specified or user has got no rights to it
-302	Error while determining the user rights
-303	User has got no access permission to document api_error_var[...] = "doc_permission" api_error_value[...] = "<right>"
-540	SMTP_SUPPORT is not enabled api_error_var[...] = "<smtp_support>" api_error_value[...] = "SMTP_SUPPRT"
-541	The SMTP DLL is not available! api_error_var[...] = "smtp_dll_available" api_error_value[...] = "SMTP_DLL_AVAILABLE"

Return value	Meaning
-542	Error sending the e-mail

In this example, the e-mail is sent from the user "dvelop" to "user1". The email contains html-formatted text and a link to a document from the repository. If an error occurs, then the error number is returned.

Note

```

// -----
// Beispiel:
// Versenden einer Email
// -----

// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_RecipientUser = "dvelop" // Empfänger
vars h_SubjectText = "Dies kommt in den Betreff" // Betreff der Email
vars h_DokId = "D000000153" // zu versendendes
// zu verlinkendes Dokument
vars h_AttachDokId = 0 // Dokument soll verlinkt
werden
vars h_Sender = "user1" // Absender
vars h_BodyFile = "c:\\\\text.html" // Datei mit Text für die Email
vars h_MailFormat = "html" // Formatierten Text
verschicken
varsh_CCRecipientUser = info@d-velop.de // Empfänger einer eMail-Kopie
(CC)
varsh_BCCRecipientUser = "d_velop" // Empfänger versteckter
// eMail- Kopie (BCC)
varsh_AttachFile = "c:\\\\attach.doc" // anzuhängende Datei
varsh_AttachDep = "T1,P1" // anzuhängende abhängige Dateien
vars h_DocStatus = "Be" oder "Pr" oder "Fr" oder "Ar"
vars h_ArchiveIndex

//Empfänger setzen
api_email_cc_recipients[1] = h_CCRecipientUser
api_email_bcc_recipients[1] = h_BCCRecipientUser

// Funktionsaufruf
h_ReturnValue = api_function ("send_email", h_RecipientUser, \
                                h_SubjectText, \
                                "", \
                                h_DokId, \
                                h_Sender, \
                                h_BodyFile, \
                                h_MailFormat, \
                                h_AttachDokId, \
                                h_DocStatus, \
                                h_ArchiveIndex, \
                                h_AttachDep, \
                                h_AttachFile, 0, 1,
1)

// Ergebnisausgabe
if (h_ReturnValue == 0)

```

```

{
    call api_log_info("Die Email wurde erfolgreich verschickt.")
}
// Fehlerausgabe
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
        zurückgeliefert!")
}

```

lock_token_acquire

This function stores a lock token for an object in the database or extends an existing one.

Parameters	Description
object_id	Name (50 character) of the LockToken. This name is freely configurable and is used to differentiate the object_ids in an unambiguous manner. e.g. REDLINE_SYNC
object_name	String (max. 200 characters), with reference to object_name. The content of object_id is defined and specifies what shall be synchronised. e.g. <doku_id>, <status>, <archivindex> The server does not interpret this value.
token	If the client wants to extend a lock, then the existing token is specified here.
object_info	Information text (max 255 characters), which can be stored with the object. This is returned, if another call wants to lock the same object.
ttl	This specifies a customized value for the "time to live". If nothing is specified, then a default of 5 minutes is adopted. If this value is not available, a default of five minutes is assumed.
user_name	User, whose name should be written to the database as the owner of the lock token.

When successfully called the function fills these global variables:

Global variables	
api_lock_token (P)	If a new lock has been successfully created, the respective token will be returned; it can be used to prolong or remove the lock.
api_lock_ttl (P)	Actual period of time which the lock will be valid for. (in minutes)
api_lock_object_info (O)	If the creation of the lock failed because the object was already locked elsewhere, the info text created along with the lock will be returned here.
api_lock_user (O)	If the creation of the lock failed because the object was already locked elsewhere, the user who created the lock will be returned here.

Return value	Meaning
0	everything OK (api_lock_token contains the existing token, api_lock_ttl contains the specified duration)
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-3	undefined error, contact d.velop support
-130	no object_id is passed
-131	no object_name is passed
-255	invalid token is passed
-600	Error while deleting an old token. More details can be found in the logviewer
-601	Object is already locked. (api_lock_user and api_lock_info contain the details)
-602	Error while determining the time to live for the token.
-603	Error while writing the time to live into the DB.
-604	Error while writing the token into the DB.

lock_token_release

Removes a lock for an object in the database.

Parameters	Description
object_id	Name (50 character) of the LockToken. This name is freely configurable and is used to differentiate the object_id's in an unambiguous manner. e.g. REDLINE_SYNC
object_name	String (max. 200 characters), with reference to object_name. The content of object_id is defined and specifies what shall be synchronised. e.g. <doku_id>, <status>, <archivindex> The server does not interpret this value.
token	Specifies the token to be deleted which was previously created using lock_token_acquire.

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-3	undefined error, contact d.velop support
-130	no object_id is passed
-131	no object_name is passed
-132	no token is passed
-605	Error while deleting the token. (DB-error or invalid combination)

start_ext_program

This example executes a command line call (start of an external program).

e.g. notepad.exe C:\\test.txt c:\\test1.txt

Note

After a successful call, api_single_info contains the return parameter value of the command.

Parameters	Description
program	Application to be executed. (e.g. notepad.exe) Command line parameters separated by a space. (e.g. C:\\test.txt c:\\test1.txt
hidden	0: Application is started visible (default) 1: Application is started hidden.

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
138	no program specified
212	invalid Doc ID specified.
226	Invalid value for publish, must be 0 or 1.
614	This document-type is not defined for web-publishing.
615	Only documents with a version in status release can be published.
616	Error while opening the program.

This example executes a command line call (start of an external program).

Note

```
// -----
// Beispiel:
// Externes Programm starten
// -----

// Variablendeklaration
vars h_ReturnValue                                // Fehlercode
vars h_Program = "notepad.exe"                    // zu startende
Anwendung
vars h_Params = "c:\\test.txt"                    // Startparameter

// Funktionsaufruf
h_ReturnValue = api_function ("start_ext_program", h_Program,
                              h_Params)

// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    call api_log_info("Externe Anwendung gestartet. ")
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
                        zurückgeliefert!")
}
```

get_current_datetime

This function returns the current time stamp in the specified format.

Parameters	Description
format	Format string, containing constants and placeholders
hr12	0: 24-hour format (default) 1: 12-hour format

Date format	String	Description	Example:
Year number	%4y	four-digit number	2007
	%2y	2-digits	07
Month	%m	1/2-digits	[1..12]
	%0m	2-digits	[01..12]
	.*m	full name	February
	%3m	Short name	Feb
Day	%d	1/2-digits	[1..12]
	%0d	2-digits	[01..31]
Week day	.*d	full name	Wednesday
	%3d	Short name	Wed
	%,d	Numeric	[1..7]
Day in the year	.*+d	Numeric	[1..366]
Hour	%h	1/2-digits	[1..12] or [0..23]

Date format	String	Description	Example:
	%0h	2-digits	[01..12] or [00..23]
Minute	%M	1/2-digits	[0..59]
	%0M	2-digits	[00..59]
Second	%s	1/2-digits	[0..59]
	%0s	2-digits	[00..59]
Milli-second	%S	1-3-digits	[0..999]
	%0S	3-digits	[000..999]
AM/PM	%p	Time format	AM/PM

Note

For % in the return string you must mask %% in the format string.

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-243	invalid format

This example returns the current time stamp.

Note

```
// -----
// Beispiel: Aktuellen Zeitstempel ausgeben
// -----
// Variablendeklaration
vars h_ReturnValue      // Fehlercode
vars h_Format =
    "Heute ist der %0d. %3m %4y, es ist %0h\:%0M\:%0s.%0S \Uhr"
                                // Format der Ausgabe

// Funktionsaufruf
h_ReturnValue = api_function ("get_current_datetime", h_Format)
// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    call api_log_info(api_single_info)
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
        zurückgeliefert!")
}
```

Show date in format "today is 1.9.2009 13:47:07 h".

Note

```
// -----
// Beispiel: Aktuellen Zeitstempel ausgeben
// -----
// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_Format =
    "Heute ist der %d.%m.%4y %h%\:0M%\:0s, 0 Uhr"
                                // Format der Ausgabe

// Funktionsaufruf
h_ReturnValue = api_function ("get_current_datetime", h_Format)
// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    call api_log_info(api_single_info)
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
        zurückgeliefert!")
}
```

Show date in format "today is 1.9.2009 1:47:07 pm".

Note

```
// -----
// Beispiel:
// Aktuellen Zeitstempel ausgeben
// -----
// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_Format = "Heute ist der \"%d.%m.%4y %h%\:0M%\:0s %p\", 1"
                                // Format der Ausgabe

// Funktionsaufruf
h_ReturnValue = api_function ("get_current_datetime", h_Format)

// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    call api_log_info(api_single_info)
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
        zurückgeliefert!")
}
```

sleep

This function sleeps for a number of milli-seconds without processor load.

Parameters	Description
<code>milli_seconds</code>	Value between 1 and 5000. All other values are ignored.

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect

process_get_id

Returns the value of the process ID in the global variable `api_single_info` (hexadecimal).

This can be useful, for example, if custom logs are written from within a hook function.

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-3	undefined error, contact d.velop support

regular_expression_test

Test of a value to a regular expression.

`regular_expression_test` calls an internal function. The validity check is actually performed by Panther, see

https://docs.prolifics.com/panther/html/eds_html/entry.htm#881493

Parameters	Description
<code>regular_expression</code>	The regular expression for the test.
<code>test_value</code>	The value to be tested.

Return value	Meaning
0	everything OK (Test-result located in <code>api_single_info</code>)
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-127	no regular expression is passed
-128	no test value is passed

In the example, the regular expression `[0-9]\{3,6\}` is checked. This is valid for all numbers with 3 to 6 digits.

Example 1; regular expression is valid

Note

```
// -----  
  
// Example 1:  
  
// Test regular expression  
  
// "api_single_info" will be set to 1 (for valid)  
  
// ----- vars h_ReturnValue  
  
vars h_Format = "[0-9]{3,6}"  
  
vars h_Wert = "1234"  
  
h_ReturnValue = api_function ("regular_expression_test", h_For-  
mat, h_Wert)  
  
call api_log_info("regular_expression_test")  
  
if (h_ReturnValue == 0)  
{  
  call api_log_info(api_single_info)  
}  
  
else  
{  
  call api_log_error("Error number :h_ReturnValue was returned!")  
}
```

Example 2; regular expression is invalid

Note

```
// -----
// Beispiel 2:
// regulären Ausdruck testen
// „api_single_info“ wird auf 0 gesetzt
// (für nicht gültig, weil ein Buchstabe enthalten ist).
// -----
// Variablendeklaration
vars h_ReturnValue                                // Fehlercode
vars h_Format = "[0-9]\\{3,6\\}" , "1a345" // Format der Ausgabe

// Funktionsaufruf
h_ReturnValue = api_function ("regular_expression_test",
h_Format)

// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    call api_log_info(api_single_info)
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
        zurückgeliefert!")
}
```

Example 3; regular expression is invalid

Note

```
// -----
// Beispiel 3:
// regulären Ausdruck testen
// „api_single_info“ wird auf 0 gesetzt
// (für nicht gültig, weil nur 2 Zeichen)
// -----
// Variablendeklaration
vars h_ReturnValue                                // Fehlercode
vars h_Format = "[0-9]\\{3,6\\}" , "12"          // Format der Ausgabe

// Funktionsaufruf
h_ReturnValue = api_function ("regular_expression_test",
h_Format)

// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    call api_log_info(api_single_info)
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
        zurückgeliefert!")
}
```

object_property_set

Sets the property values for objects. Setting a null string for the `value` deletes the property value for the object.

Parameters	Description
<code>property_name</code> (P)	Internal name of the object property
<code>object_id</code> (P)	Object name. (e.g. user name, group name depending on object class id)
<code>object_class_id</code> (P)	Object class ID 1: User 2: Groups 3: Unit of organization 4: Activity profiles 5: Datasets 6: repository fields 7: Document types 8: Document classes 9: Permission Profiles 10: Restriction sets 11: Workflows
<code>property_value</code>	property value

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-3	undefined error, contact d.velop support
-140	no property name is passed
-141	no object id is passed
-142	no object classes id is passed
-240	invalid object classes ID is passed
-241	unknown property name is passed

object_properties_get

Determines the property values for a property or for an object. If both import parameters are specified, exactly the value of the object property `name` for the object `object_id` is returned.

The number of returned results will be written into `api_single_info`.

The following arrays of the context will be filled:

Variable	Description
<code>api_property_name</code>	Property name
<code>api_property_caption</code>	Short description of the property
<code>api_property_description</code>	Detailed description of the property
<code>api_property_value</code>	Value of the property
<code>api_object_id</code>	Object ID of the property (e.g. user name, group name, ...)
<code>api_doc_status_info</code>	Additional information to the status (0/1)=(unverified/ verified, blocked/released).

Variable	Description
api_doc_editor	Editor

Parameters	Description
object_id (O/P)	Object name. (e.g. user name, group name depending on object class id)
object_class_id (O/P)	Object class ID 1: User 2: Groups 3: Unit of organization 4: Activity profiles 5: Datasets 6: repository fields 7: Document types 8: Document classes 9: Permission Profiles 10: Restriction sets 11: Workflows
property_name (O/P)	Internal name of the object property
language (O)	If a language is passed here, then the <code>caption</code> and <code>description</code> is filled in the respective language (if this exists). Else in the first existing language.

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-3	undefined error, contact d.velop support
-143	either property, nor object ID and object class ID are passed
-240	invalid object classes ID is passed
-241	unknown property name is passed

set_signature_data

Note

the function is no longer supported from version 7.1.

A call returns error -671.

Writes the signature data for a document into the database.

Parameters	Description
doc_id	Document ID of the document.
doc_status	Status of the document (B=Processing, P=Verification, F=Release, A=Archive).
archive_index	Respective archive-index

Using the global context variables the signature data set is set before the start:

Variable	Description
api_sig_docextension	File extension of the dependent document to which the signature applies.
api_sig_user	User who applied the signature

Variable	Description
api_sig_version	Which application performed the signature and verification?
api_sig_tstamp	Specifies the point of time, when the user signed the document.
api_sig_algorithm	This value represents the used signature procedure.
api_sig_digest	OID of the HASH procedure.
api_sig_keylength	The used key length in bits.
api_sig_reason	Reason, why the user signed the document.
api_sig_type	Type of signature (empty for advanced and qualified signatures).
api_sig_reached_verif_level (0)	Reached verification depth (1-9)
api_sig_requested_verif_level (0)	Desired verification depth (1-9)
api_sig_content (0)	"a" for attach "d" for detached (default value, if empty) "e" for embedded
api_sig_provider (0)	Provider
api_sig_producer (0)	Producer
api_sig_certissuer	This attribute contains the name of the certifying authority who issued the certificate for the person.
api_sig_certsubject	This attribute contains the name of the person who signed the document. The certificate used for this signature is issued for this person.
api_sig_certvalidfrom	Specifies the point of time from which the certificate of this person is valid.
api_sig_certvalidto	Specifies the point of time until which the certificate of this person is valid.
api_sig_certserial	This is the serial number of the certificate. The combination off CertIssuer and CertSerial is unique.
api_sig_algorithm	Certification algorithm.
api_sig_certrestriction (0)	Restrictions applying to the according certificate as text.

Note

Limitations:

In contrast to the client API function `SetSignatureData` the server API function has only limited support for "basic" signatures as the certificate information are mandatory parameters.

get_objectid_from_synchid

Determines from a synchronization-ID, created by the transport system, the internal ID and type of Master Data-object.

global variables	Description
api_md_object_id	Internal ID of the Master Data-object
api_md_object_type	Type of the Master Data-object

Parameters	Description
synch_id	Synchronization id

Return value	Meaning
0	Everything OK
-5009	Empty or not existing synchronization-ID

1.4.20. d3fc-calls

From d.3 version 6.2 you can execute d3rfc calls using the server API, i.e. from within JPL. This can be used to address d.3 search, d.3 rendition service or other servers supporting d3fc.

It is also supported by the d.3 server itself (e.g. for testing purposes).

Note

However, it has to be noted that 2 server processes are required for this effect and that these are blocked for all other activities for the duration of the call. Such an operation is therefore not recommended in a productive scenario.

The procedure for making such a call is equivalent to a call via `d3remote.ocx` (see d.3 API documentation).

Warning

The d3fc calls may only be used to address other services communicating via d3fc but not the server itself.

You cannot perform any administrative operations (such as creating users, writing to optional fields) using d3fc calls.

The individual functions with a detailed list of parameters is provided below.

Note

4 examples can be found at the end of this chapter.

- `d3fc_user_set`
- `d3fc_password_set`
- `d3fc_remote_server_set`
- `d3fc_port_set`
- `d3fc_server_set`
- `d3fc_timeout_set`
- `d3fc_function_name_set`
- `d3fc_importing_set`
- `d3fc_exporting_set`
- `d3fc_exporting_get`
- `d3fc_table_set_import_file`
- `d3fc_table_set_export_file`
- `d3fc_table_set_headline`
- `d3fc_table_add`
- `d3fc_field_set`
- `d3fc_field_get`
- `d3fc_first`
- `d3fc_next`
- `d3fc_execute`

d3fc_user_set

This function sets the executing user for the d3fc call.

Parameters	Description
<code>user_name</code>	The user to be used. (default: <code>d3_server</code>)

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-102	no user name specified

d3fc_password_set

This functions sets the password of the executing user for the d3fc call.

Parameters	Description
password	Password (default: server_api_client)

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-137	no password specified

d3fc_remote_server_set

This function defines the name / IP of the hosts to be addressed.

Parameters	Description
remote_server	Hostname / IP (Default: Content of the config-variable d3fc_remote_server_name)

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect

d3fc_port_set

Port of the host, which shall be addressed

Parameters	Description
port	Port (Default: Content of the config-variable d3fc_remote_server_tcpip_port)

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect

d3fc_server_set

Server short name of d.3 processes.

Parameters	Description
server	Server short name (Default: Content of the config-variable d3fc_remote_server_kuerzel)

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect

d3fc_timeout_set

This function specifies the time in seconds to be waited for the response of the remote server.

Parameters	Description
timeout	Timeout in seconds (Default: Content of the config-variable d3fc_remote_server_timeout)

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect

d3fc_function_name_set

This function sets the name of the API-call.

Note

This value must be set first, i.e. before all import- and export-parameters, tables and files.

Parameters	Description
function_name	Name of the d.3 function

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-133	no function name specified

d3fc_importing_set

This function sets the import parameters.

Note

This must be called once for each parameter.

Parameters	Description
param_name	Name of the parameter
param_value	Value of the parameter

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-133	no function name specified
-134	no parameter name specified

d3fc_exporting_set

This function sets the export parameters.

Note

This must be called once for each parameter.

Parameters	Description
param_name	Name of the parameter

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-133	no function name specified
-134	no parameter name specified

d3fc_exporting_get

This function queries the export parameters after d3fc_execute.

It returns the values of those parameters which were set prior to d3fc_execute with d3fc_exporting_set.

The value will be stored in the variable api_single_info.

Parameters	Description
param_name	Name of the parameter

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-133	no function name specified
-134	no parameter name specified

d3fc_table_set_import_file

This function sets an import file.

The name must be specified including the path to the file which should be passed when executing the call.

Note

This value can only be set as an alternative to an import table and NOT at the same time!

Parameters	Description
filename	File name

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect

Return value	Meaning
-150	no file name specified.
-250	File does not exist
-606	Import table is already set.

d3fc_table_set_export_file

This function is setting the export file.

The name must include the path to the file, to which the return of the call is to be written.

Note

This value can only be set as an alternative to an export table and NOT at the same time!

Parameters	Description
filename	File name

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-150	no file name specified.
-584	The file already exists!
-611	Export table is already set.

d3fc_table_set_headline

This function defines the table header.

Note

It can only be used as an alternative to an import- or export-file and NOT at the same time.

Parameters	Description
headline	Form: <head1><space><head2><space>...<headN><space><Point>

Note

`doc_id doc_status`. The placeholders of the fields are separated by spaces. At the end, there must be at least one space (" ") and a point ("."). The placeholders contain the field name and sufficient spaces to hold the field content.

The document ID can have a maximum length of 10 characters. Thus, the placeholder must have a length of at least 10 characters as well. The name `doc_id` has got 6 characters; hence, 4 spaces must be added (+ the separating space): The status has got 2 characters.

The name `doc_status` already has 10 characters; hence; no additional spaces are required (except for the separating space):

```
1234567890_12_. doc_id doc_status . head_i_o "I"=Header of the
Import table (Input) "O"=Header of the export table (Output)
```

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-135	no headline specified
-242	invalid value for <code>head_i_o</code> specified (must be I or O)
-607	Import file is already set.
-611	Export file is already set.

d3fc_table_add

Add table row to import table.

Note

The headline must be set already.

Parameters	Description
n/a	

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-135	Headline is not set already.

d3fc_field_set

This function sets a field in the current row of the import table.

Note

The headline must be set already.

`d3fc_table_add` must have been called before.

Parameters	Description
<code>field_name</code>	Name of table column in headline.
<code>field_value</code>	Value of the field

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-135	Headline is not set already.
-136	no <code>field_name</code> specified.
-613	Field does not exist in table.

d3fc_field_get

This function reads a field in the current export table.

Note

The field must have been set in the headline prior to `d3fc_execute`.

`(d3fc_table_headline_set)`

`d3fc_first()` / `d3fc_next()` must have been called before.

Parameters	Description
<code>field_name</code>	Name of table column in headline.

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-136	no <code>field_name</code> specified.
-612	Field does not exist in table.

d3fc_first

Load first table row of the export table.

Parameters	Description
n/a	

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-609	End of table reached. (constant EOT = -609)

d3fc_next

Load next table row of the export table.

Parameters	Description
n/a	

Return value	Meaning
0	everything OK
-1	Panther syntax error, details see d.3 logviewer
-2	function was called incorrect
-609	End of table reached. (constant EOT = -609)

d3fc_execute

Execute an API call.

Note

The return value (e.g. 0002 - Wrong user/password) will be written into the variable `api_single_info`.

Parameters	Description
n/a	

Return value	Meaning
46	Error: Invalid proxy-function while receiving the data.
45	Error: no proxy-support while receiving the data.
25	Error while sending the file <code>api_d3fc_import_file</code> .
24	Error while receiving the data: the sent data have got the length 0
22	Error receiving the data: nothing was sent
21	Error while sending the file <code>api_d3fc_import_file</code> .
20	Error while opening the target file for the received data
0	everything OK
-1	Error during <code>fcCreate</code>
-2	Error during <code>fcGetHeader</code>
-3	Error during <code>fcConnect</code>
-4	Error during <code>fcSend</code>
-10	Timeout during <code>fcReceive</code> . (no response within defined timeout)
-15	Error before sending the file <code>api_d3fc_import_file</code> : File could not be opened or not enough RAM.
-21	Error while opening the passed file.
-22	Error while saving the passed file under the target file name (<code>api_d3fc_export_file</code>).
-609	End of table reached. (constant EOT = -609)

Examples

- `ValidatePasswordForUser`
- `SearchDocumentInIDList`
- `ReceiveNote`
- `SendNote`

Example 1: ValidatePasswordForUser**Note**

```
// -----
// Beispiel 1:
// ValidatePasswordForUser
// -----
// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_User = "armin"       // Benutzer für d3fc-Aufruf
vars h_Password = "frikadelle" // Passwort für d3fc-Aufruf
// Verbindungsoptionen
call api_function ("d3fc_user_set", h_User)
    // Benutzer; default: d3_server
call api_function ("d3fc_password_set", h_Password)
    // Passwort; default: sever_api_client
call api_function ("d3fc_remote_server_set", "host")
    // Hostname; default: d3fc_remote_server_name
call api_function "d3fc_port_set", "3400")
    // Port; default: d3fc_remote_server_tcpip_port
call api_function ("d3fc_server_set", "Y")
    //Kürzel; default: d3fc_remote_server_kuerzel
// Funktionsname
call api_function ("d3fc_function_name_set",
    "ValidatePasswordForUser")
//Funktionsname
// ImportParameter
call api_function ("d3fc_importing_set", "user", h_User)
//zu prüfender Benutzername
call api_function ("d3fc_importing_set", "password", h_Password)
// zu prüfendes Passwort
// Funktionsaufruf
h_ReturnValue = api_function ("d3fc_execute")
// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    if (api_single_info == 0)
        call api_log_info("Kombination aus Benutzer und
            Passwort gültig")
    else
        call api_log_error("Fehlernummer :api_single_info \
            wurde von ValidatePasswordForUser \
            zurückgeliefert!")
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
        zurückgeliefert!")
}
```

Example 2: SearchDocumentInIDList**Note**

```
// -----
// Beispiel 2:
// SearchDocumentInIDList
// -----
// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_counter = 0          // Schleifenzähler
vars h_DokID = "Y000005174" // zu suchendes Dokument
//Verbindungsoptionen
call api_function ("d3fc_user_set", "user")
// Benutzer; default: d3_server
call api_function ("d3fc_password_set", "password")
// Passwort; default: sever_api_client
call api_function ("d3fc_remote_server_set", "host")
// Hostname;
// default: d3fc_remote_server_name
call api_function ("d3fc_port_set", "3400")
// Port;
// default: d3fc_remote_server_tcpip_port
call api_function ("d3fc_server_set", "Y")
// Kürzel;
// default: d3fc_remote_server_kuerzel

//Funktionsname
call api_function ("d3fc_function_name_set",
"SearchDocumentInIDList")
// Funktionsname

//Import-Tabelle
call api_function ("d3fc_table_set_headline", "doc_id      .",
"I")

//Tabellenkopfzeile
call api_function ("d3fc_table_add")
// Tabellenzeile hinzufügen
call api_function ("d3fc_field_set", "doc_id", h_DokID)
//Wert für eine Spalte in dieser Zeile

// ExportParameter
call api_function ("d3fc_exporting_set ", "number")
//Anzahl Treffer

// Export-Tabelle
call api_function ("d3fc_table_set_headline",
"doc_id doc_status .", "O")
// Tabellenkopfzeile
// Funktionsaufruf
h_ReturnValue = api_function ("d3fc_execute")
// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
vars h_numberReturned
vars h_ret
call api_log_info("SearchDocumentInIDList lieferte ReturnCode
\
:api_single_info")
}
```

```
// ExportParameter abfragen
call api_function ("d3fc_exporting_get", "number")
h_numberReturned = api_single_info
call api_log_info(":api_single_info Treffer")
h_Ret = api_function ("d3fc_first")
    // Erste Zeile holen
while (h_Ret != EOT)
    //EOT ist Konstante für Rückgabe -609
{
    for h_counter=1 while h_counter <= h_numberReturned step 1
    {
        call api_log_info(": (h_counter). Treffer:")
        call api_function ("d3fc_field_get", "doc_id")
        // Wert abfragen
        call api_log_info("doc_id=:api_single_info")
        call api_function ("d3fc_field_get", "doc_status")
        call api_log_info("doc_status=:api_single_info")
    }
    h_Ret = api_function ("d3fc_next ")
    // nächste Zeile
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
        zurückgeliefert!")
}
```

Example 3: ReceiveNote**Note**

```
// -----
// Beispiel 3:
// ReceiveNote
// -----
// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_DokID = "Y000005177" // DokuID
vars h_FileName = "c:\\\\notiz.txt" // Speicherort für Notiz
//Verbindungsoptionen
call api_function ("d3fc_user_set", "user")
    // Benutzer; default: d3_server
call api_function ("d3fc_password_set", "password")
    // Passwort; default: sever_api_client
call api_function ("d3fc_remote_server_set", "host")
    // Hostname;
    // default: d3fc_remote_server_name
call api_function ("d3fc_port_set", "3400")
    // Port;
    // default: d3fc_remote_server_tcpip_port
call api_function ("d3fc_server_set", "Y")
    // Kürzel;
    // default: d3fc_remote_server_kuerzel
// Funktionsname
call api_function ("d3fc_function_name_set", "ReceiveNote")
    // Funktionsname
// ImportParameter
call api_function ("d3fc_importing_set", "doc_id", h_DokID)
    // Dokument
// Exportfile
call api_function ("d3fc_table_set_export_file", h_FileName)
    // Dateiname, unter dem die
    // Notiz abgelegt wird
// Funktionsaufruf
h_ReturnValue = api_function ("d3fc_execute")
// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    call api_log_info("Notizdatei wurde unter
        :h_FileName gespeichert.")
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
        zurückgeliefert!")
}
}
```

Example 4: SendNote**Note**

```
// Beispiel 4:
// SendNote
// -----
// Variablendeklaration
vars h_ReturnValue                                // Fehlercode
vars h_DokID = "Y000005177"                      // DokuID
vars h_FileName = "c:\\\\notiz.txt"               // Speicherort für
Notiz
//Verbindungsoptionen
call api_function ("d3fc_user_set", "user")
// Benutzer; default: d3_server
call api_function ("d3fc_password_set", "password")
// Passwort; default: sever_api_client
call api_function ("d3fc_remote_server_set", "host")
// Hostname;
// default: d3fc_remote_server_name
call api_function ("d3fc_port_set", "3400") // Port; default:
// d3fc_remote_server_tcpip_port
call api_function ("d3fc_server_set", "Y") // Kürzel; default:
// d3fc_remote_server_kuerzel
//Funktionsname
call api_function ("d3fc_function_name_set", "SendNote")
//Funktionsname
//ImportParameter
call api_function ("d3fc_importing_set", "doc_id", h_DokID)
//Dokument
//Importfile
call api_function ("d3fc_table_set_import_file", h_FileName)
//Dateiname der Notiz
// Funktionsaufruf
h_ReturnValue = api_function ("d3fc_execute")
// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    call api_log_info("Notizdatei :h_FileName wurde gespeichert.")
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
zurückgeliefert!")
}
```

1.4.21. JAVA

- java_add_param
- java_clear_param
- java_call_static

java_add_param

This function adds a new parameter to the call of the next Java-function.

The called Java-function receives all added parameters as a string-array.

The parameters are stored in the string-array in the order in which they were added via this function.

Note

available since d.3 version 7.2

Parameters	Description
param_value	This string is added as a parameter.

Return value	Meaning
0	everything OK
1	Java-support is not enabled / Java-VM cannot be addressed.
2	An error occurred in the JNI-communication (see logviewer for further information).
3	An error occurred in the d.3 server (see logviewer for further information).

java_clear_param

This function deletes all previously added parameters (see [java_add_param](#)).

Parameters	Description
	none

Return value	Meaning
0	everything OK
1	Java-support is not enabled / Java-VM cannot be addressed.
2	An error occurred in the JNI-communication (see logviewer for further information).
3	An error occurred in the d.3 server (see logviewer for further information).

java_call_static

This function calls a static function of a Java-class. The previously added parameters are passed. The class must be located in the Java CLASSPATH.

The method to be called must have the following signature:

```
static int javaTest(FieldInterface field, String[] params)
```

The is equivalent to the signature of Java-hook functions. The return value of the Java-function is written to the global field `api_single_info`.

Note

Having called this function, all added parameters are deleted automatically. An explicit call of [java_clear_params](#) is not necessary.

Parameters	Description
class_name	The fully qualified class name in JNI notation. Specify the class name including the namespace here. Dots (".") are replaced by slashes ("/"). Example: "com/dvelop/test/JavaHooks". This is equivalent to the Java-class JavaHooks in the package com.dvelop.test.
method_name	The name of the method to be called. This method must be static and have the signature described above. Example: "javaTest".

Return value	Meaning
0	everything OK
1	Java-support is not enabled / Java-VM cannot be addressed.
2	An error occurred in the JNI-communication (see logviewer for further information).
3	An error occurred in the d.3 server (see logviewer for further information).

1.4.22. Creating async jobs

Warning

From d.3 version 8.0.

d3async_job_open

Warning

From d.3 version 8.0.

Opens a async job

With `d3async_job_set_attribute` or `d3async_job_set_lin002_attribute` additional async job parameter can be set.

Then, the async job will be created in the database afterwards via `d3async_job_close`.

Parameters	Description
<code>doc_id</code>	Document ID
<code>job_type</code>	Async job type (e.g. LIN002)
<code>user_name</code>	User for whom the job shall be executed. If empty, then the user name of the context of the hook is used.

Return value	Meaning
0	everything OK
-680	no job type specified.

d3async_job_set_attribute

Sets async job attributes.

Parameters	Description
<code>attr_name</code>	async job attribute name
<code>attr_value</code>	Value for async job attribute

Return value	Meaning
0	everything OK
-681	no attribute name specified

d3async_job_set_lin002_attr

Sets the job attributes specially for LIN002-jobs and thus, can only be called for LIN002-jobs.

Parameters	Description
<code>attr_type</code>	must have one of the following values <code>ernie_job_attr_name</code> : recognition attributes for the superordinate document <code>inherit_field_name_1</code> Inherited attributes of subordinated document to the superordinate document, whereby the attributes of the superordinate document shall not be overwritten <code>inherit_field_name_2</code> Inherited attributes of subordinated document to the superordinate document, whereby the attributes of the superordinate document will be overwritten <code>inherit_field_name_3</code> Inherited attributes of superordinate document to the subordinate document, whereby the attributes of the subordinate document shall not be overwritten <code>inherit_field_name_4</code> Inherited attributes of superordinate document to the subordinate document, whereby the attributes of the subordinate document will be overwritten
<code>attr_name</code>	attribute name (e.g. "dok_dat_feld_1")
<code>attr_value</code>	Value for document property

Return value	Meaning
0	everything OK
-681	no attribute name specified

d3async_job_close

Finishes a job opened with `d3async_job_open`, writes it into the database and set the job status so that the d.3 async process can start it.

`api_single_info` returns the job number on success.

Return value	Meaning
0	Async job was created and released for processing
-682	Async job could not be created

1.5. JPL-Styleguide

1.5.1. File location and name

All hook functions should be combined into one file, if possible. Nothing is worse than several files containing one hook-function each (e.g. `HookInsertEntry_10`).

If possible, the hook file should have the following properties:

- The name should contain the company name or a customer short name.
- The JPL file should be stored in the d.3 server directory in a separate subdirectory:

Note

```
C:\d3\d3server\hooks\d3P\testfirma.jpl
```

1.5.2. Comments

Comments are essential for the readability and maintenance.

The comment itself is prefixed with two slashes `//` in JPL. These two characters can be placed in any column of the JPL code. However, they should be placed on the same level as the code (indent).

An individual comment line could look like this:

Note

```
// Kommentarzeile
```

Multi line comments should be bracketed at the beginning and the end.

Note

```
//-----  
// Komentarzeile  
// Komentarzeile  
//-----
```

1.5.3. Header

The header of a JPL file (external JPI scripts as well as JPL files for hook functions) must contain the following information:

- Name of the hook-function
- When was the function created and by whom
- Rough description
- Change history

The change history should also contain information on editor and date. Furthermore, a number list can be used. Thus, the header could reference the change number 007. In the code, this number could also be included to make finding the alteration easy.

Note

```

//-----
-----
// Für wen?
//   Firma:           Firmen-Name
//   Ansprechpartner: _____
//-----
-----
// Was?
// Die hier enthaltenen Funktionen erfüllen folgende Aufgaben:
// Dieses Script wird periodisch durch den Prozess-Manager
// aufgerufen
// und verschiebt die Wiedervorlagen aus dem WFL-Parkplatz zu den
// entsprechenden Sachbearbeitern
//
// proc Postkorbverteilung(h_DokID)
//   Wurde eine Dokument-ID vorgegeben wird nur diese
// ausgeführt!
//   1.Ermittlung der Vorgänge / Dokumente, die im WFL-
// Parkplatz
//   liegen.
//   2.Wurde durch die Clearingstelle oder für die
// Arbeitsanweisungen
//   schon ein Sachbearbeiter vorgegeben erfolgt hier die
//   WFL-Wiedervorlage.
//   3.Wurde kein Sachbearbeiter vorgegeben erfolgt der Versuch
// der
//   Ermittlung mittels Nummernkreis-Verteilung für alle
// Dokumente
//   im Parkplatz! Es erfolgt für alle ermittelten
// Sachbearbeiter
//   eine direkte WFL-Wiedervorlage.
//   4.Für alle Parkplatz-Dokumente, die nun noch im Workflow
// sind,
//   erfolgt die WFL-Wiedervorlage anhand der
//   NichtklassifizierteVerteilung()
//
// proc NummernKreisVerteilung()
//   Ermittlung der Sachbearbeiter und Weiterleitung
//   Rückgabewert: Sachbearbeiter einer oder mehrere
//
// proc NichtklassifizierteVerteilung()
//   Alle Dokumente / Vorgänge, denen kein Sachbearbeiter
// zugewiesen
//   werden konnte, werden mittels eines "rundum"-Verteilungs-
// Mechanismus auf die anwesenden Sachbearbeiter verteilt.
//   Rückgabewert: ein Sachbearbeiter
//
// proc VerteilungBeiAbwesenheit()
//   ToDo: Der älteste Tages-Stapel eines abwesend gemeldeten
// Benutzers muss hier ermittelt werden. Alle Dokumente
// dieses
//   Sachbearbeiters werden dann mittels der
//   NichtklassifizierteVerteilung() verteilt.

```

```
//      Was passiert hier mit gesperrten Dokumenten die vom
//      selben Sachbearbeiter weiterverarbeitet werden sollen?
//
//  string proc ErmittlungPESNr(h_Sachbearbeiter )
//      Ermittlung der PES_Nr. anhand des Sachbearbeiters.
//
//  string proc ErmittlungBenutzer( h_PESNr )
//      Ermittlung des Benutzers anhand der PES-Nummer
//-----
//-----
// Von wem?
//  d.velop AG / [Kürzel des Bearbeiters]
//  [Datum der ersten Version]
//
//-----
// Historie:
// 21.02.2006  Erste Version für Prototyp
// 06.03.2006  [agoe] Erste Integration von Nummernkreis- und
//              NichtklassifizierteVerteilung
// 06.03.2006  [kmoc] Überarbeitung und Anpassung an neuen
// Aktenplan
// 08.03.2006  [kmoc] Anpassung an Sparten und Produkte!
// 03.04.2006  [agoe] Realisierung der Verteilung bei Abwesenheit
// 04.04.2006  [kmoc] Anpassung an die Matrix und den Workflow
// ohne
//              Parkplatz
//-----
//-----
```

The code itself should also be sufficiently commented.

Recommendation:

- Large logical blocks (e.g. loops , queries) should be commented.
- Alteration numbers from the header could be inserted here.

1.5.4. Variables

- General information
- advanced properties fields and document types

General information

The definition of variables should always be done in the header of the hook or the procedure header.

The declaration of variables or constants within the code is prohibited or should at least be reduced to a minimum.

Variables within the JPL-file will be defined with `h_[VarName]`.

Note

`h_Count` oder `h_SachBearbeiter` oder `h_VersNr`.

Advanced properties fields and document types

The individual hook functions repeatedly refer to document types and their fields. In order to improve the legibility of the code, the following scheme should be obeyed.

The following constants are defined in the header of the hook:

- advanced properties-fields / h_DDF = Dok_Dat_Feld

```
h_DDF_ 'Name' = ,DB-Position'
```

The constants are sorted by 'DB-position'

- Short name of the document type DT = DocType

```
h_DT_ 'NAME' = 'Document Type ID'
```

The constants are sorted by the 'Document Type ID':

Note

```
/-----
// DB-Position der erweiterte Eigenschaften-Felder
//-----
vars h_DDF_MANDANT          = 2
vars h_DDF_BUCHUNGSKREIS   = 3
vars h_DDF_OBJEKT          = 7
vars h_DDF_DOKUTYP         = 25
//-----
// Dokument/Aktenarten
//-----
vars h_DA_AAUF             = "AAUF"
```

The access could look like this:

Note

```
// DB-Zugriff
DBMS SQL SELECT a.dok_dat_feld:h_DDF_RENAME \
FROM firmen_spezifisch a \
WHERE a.kue_dokuart = :+h_DOK_AREMP \
AND a.dok_dat_feld:h_DDF_RECHNR=
:+dok_dat_feld[h_DDF_RECHNR]
```

1.5.5. Debugging

Every procedure contains at least two debugging functions, which are setting the start and the end of this function. Every further call of the debugging function also contains the name of this function.

Note

```

proc FunktionsName
{
    call api_log_debug ("FunktionsName start")
    // Variablendeklaration
    ...
    // Code
    ...
    call api_log_debug ("FunktionsName Irgendeine Meldung")
    // Code
    ...
    call api_log_debug ("FunktionsName end")
}

```

1.5.6. Loops / branches**Note**

Every single level is indented!

- For loop
- While loop
- If query

For loop

Whenever possible, do not use one character variables as counters for loops! Instead, variables in the form `h_Count` should be used.

Note

```

vars h_Count
for h_Count = 0 while(string( h_Count, 1) != "") step 1
{
    Anweisungs-Block
}

```

While loop**Note**

```

vars h_Count = 1
while (string(h_Count,1) != "")
{
    h_Count = h_Count+1
}

```

If query

If more than one condition is to be checked, then each condition should be placed in an individual line. Furthermore, bracketing should be used to support the legibility. The conditions should be vertically aligned.

Note

```

if    (      (x > y) \
          && (z < x) \
          || (x > w))
{
    Anweisungs-Block
}
else
{
    Anweisungs-Block
}

```

1.6. Useful functions

1.6.1. WriteNotizEntry

This function reads the values from the Field 61 and writes them to the notes.

Note

```

// Werte aus 61er-Feld lesen und in die Notizdatei schreiben

proc WriteNotizEntry( h_DokID )
{
    // Pfad bereitstellen UNIX
    // d.3-Archiv-ID in den Pfad einbauen!

    vars h_Pfad      = "/opt/d3/d3server/hooks/d3:
(d3fc_server_kuerzel)/
                        work/"
    vars h_FileName = h_Pfad ## h_DokID ## "_Notiz.txt"
    vars h_Count = 1
    vars h_ReturnValue
    vars h_Flag = 0
    // Lesen der 60-er Felder
    for h_Count = 1 while (dok_dat_feld_61[h_Count] != "") step 1
    {
        // Schreiben in eine tmp-Datei mit Doku-ID als Dateinamen
        h_ReturnValue = \
        api_function ("file_add_line", h_FileName, \
                        \":(dok_dat_feld_61[:h_Count])\\"", "APPEND" )
        h_Flag = 1
    }
    if( h_Flag == 1 )
    {
        // Schreiben der Datei in den Notiz-Eintrag
        h_ReturnValue = api_function ("note_add_file", \
                                      h_FileName, h_DokID, "TEST1"
        )
        // Löschen der Datei
        if (h_ReturnValue == 0)
            call api_function ("file_delete", h_FileName)
    }
}
} // end of WriteNotizEntry

```

1.6.2. AddPrefixAndZeroes

This function adds a specified prefix and zeros to a passed string until the string length `wf_MaxLength` is reached. If the string already begins with the desired prefix, no changes are applied.

Note

```
// Einer Zeichenkette einen Präfix und Nullen anfügen

string proc AddPrefixAndZeros (wf_String, wf_Prefix,
wf_MaxLength)
{
    if (wf_String(1, @length(wf_Prefix)) != wf_Prefix)
    {
        while (@length(wf_String) < wf_MaxLength)
            wf_String = "0" ## wf_String
        wf_String = wf_Prefix ## wf_String
    }
    return wf_String
} // end of AddPrefixAndZeros
```

1.6.3. CutPrefixAndZeros

The function `CutPrefixAndZeros` removes a prefix and following zeros from a string.

Note

```
string proc CutPrefixAndZeros( wf_String, wf_Prefix,
wf_MinLength )
{
    if (wf_String(1, @length(wf_Prefix)) == wf_Prefix)
    {
        wf_String = wf_String(@length(wf_Prefix)+1) // Cut Prefix
        while (wf_String(1,1) == "0" && @length(wf_String)
            > wf_MinLength)

            wf_String = wf_String(2)
    }
    return wf_String
} // end of CutPrefixAndZeros
```

1.6.4. OpenSessionConnection

The following function opens an ODBC connection.

Note

```

proc OpenSessionConnection( h_Umgebung )
{
    vars h_tmpUmgebung, h_Count, h_Session, h_DBReturnCode

    call api_function ("string_to_lower", h_Umgebung)
    h_tmpUmgebung = api_converted_string

    // Ermittlung des ODBC-Namens abhängig von der Umgebung
    for h_Count = 1 while(h_Count <= h_MaxConnectCount) step 1
    {
        if (h_tmpUmgebung == h_ODBC_Name[h_Count] \
            (@length(h_ODBC_Name[h_Count])))
            break
    }
    if (h_Count <= h_MaxConnectCount)
    {
        h_Session = h_SessionName[h_Count]

        call api_log_info ("CONNECTION-STATUS für :h_Session " \
                           ## dm_is_connection( h_Session ))
        if (dm_is_connection( h_Session ) == 0)
        {
            DBMS DECLARE :h_Session CONNECTION FOR USER
:+h_ODBC_USER \
                [h_Count] \
                PASSWORD :+h_ODBC_PW[h_Count] \
                DATASOURCE :h_ODBC_Name[h_Count]
            h_DBReturnCode = @dmretcode
            DBMS WITH CONNECTION :h_Session COMMIT
        }

        if (dm_is_connection( h_Session ) == 1)
        {
            // Test-Abfrage gegen Datenbank!
            vars h_TmpValue
            DBMS WITH CONNECTION :h_Session ALIAS h_TmpValue
            DBMS WITH CONNECTION :h_Session SQL SELECT NAME \
                FROM :(h_TableOwner[h_Count]). \
                VWKI_KDNR_ADR \
                WHERE KDNR = 4711
            h_DBReturnCode = @dmretcode
            DBMS WITH CONNECTION :h_Session ALIAS
            DBMS WITH CONNECTION :h_Session COMMIT
            call api_log_info ("1.Fehlercode: :(@dmengerrcode) \
                               :(@dmengerrmsg) \
                               :(@dmretcode) :(@dmretmsg) !")
            if (h_DBReturnCode != 0)
            {
                DBMS CLOSE CONNECTION :h_Session
                DBMS DECLARE :h_Session CONNECTIN FOR USER \
                    :+h_ODBC_USER[h_Count] \
                    PASSWORD :+h_ODBC_PW[h_Count] \
                    DATASOURCE :h_ODBC_Name[h_Count]
            }
        }
    }
}

```

```

        DBMS WITH CONNECTION :h_Session COMMIT
        call api_log_info (2.Fehlercode: :(@dmengerrcode) \
                           :(@dmengerrmsg) : \
                           :(@dmretcode) :(@dmretmsg) !")

        //return -1005
        // msglib.usr ( DB-ERROR Siehe Log-File
        // für weitere Informationen)
    }
}
return h_Count
}
return -1
} // end of OpenSessionConnection

```

1.6.5. GroupCheckedIn

This function `GroupCheckedIn(h_Group)` determines, if at least one member of a group is checked in.

Note

```

// Prüfen, ob min. 1 Mitglied der Gruppe anwesend

proc GruppeAnwesend( h_Group )
{
    vars h_TmpFlag, h_MaxCount = 0, h_Count = 0, h_ReturnValue

    // Ermittlung aller Mitglieder einer Gruppe
    if (BenutzerIstGruppe (h_Group) == 1)
    {
        h_ReturnValue = api_function ("group_get_members",
h_Group)
        if (h_ReturnValue == 0)
        {
            h_MaxCount = api_single_info

            for h_Count = 1 while(h_Count <= h_MaxCount) step 1
            {
                DBMS ALIAS h_TmpFlag
                DBMS SQL SELECT checkout_user_flag \
                           FROM benutzer \
                           WHERE benutzername = :+api_users[h_Count]
                DBMS ALIAS
                if( h_TmpFlag != 1 )
                    return 1
            }
        }
    }
    return 0
} // end of GruppeAnwesend

```

1.6.6. UserCheckedIn

This function checks, if a user is currently checked in.

Note

```
// Prüfen, ob Benutzer anwesend
proc BenutzerAnwesend (h_SachBe)
{
    call api_function ("user_get_data", h_SachBe)
    if (api_user_checkout_flag == 0)
        return 1
    else
        return 0
} // end of BenutzerAnwesend
```

Note

```
proc BenutzerAnwesend (h_SachBe)
{
    DBMS ALIAS h_TmpFlag
    DBMS SQL SELECT checkout_user_flag \
                FROM benutzer \
                WHERE benutzername = :+h_SachBe
    h_MaxCount = @dmrowcount
    DBMS ALIAS
    if (h_MaxCount != 0 &&& h_TmpFlag != 1)
        return 1
    else
        return 0
} // end of BenutzerAnwesend
```

1.6.7. UserIsGroup

The function `UserIsGroup` determines, if the specified user name is a group.

Note

```
// Prüft, ob h_SachBe eine Benutzergruppe ist.
proc BenutzerIstGruppe (h_SachBe)
{
    vars h_TmpFlag, h_MaxCount = 0
    DBMS ALIAS h_TmpFlag
    DBMS SQL SELECT benutzergruppe \
                FROM benutzergruppen \
                WHERE benutzergruppe = :+h_SachBe
    h_MaxCount = @dmrowcount
    DBMS ALIAS
    return h_MaxCount
} // end of BenutzerIstGruppe
```

1.6.8. DetermineUser

The function `DetermineUser` finds users based on a staff number (PESNr) which was stored in the optional user field 1.

Note

```
// Bestimmung des Benutzers anhand einer PESNr
string proc ErmittlungBenutzer (h_PESNr)
{
    if (h_PESNr != "")
    {
        vars h_MaxCount = 1
        vars h_Benutzer = ""
        DBMS ALIAS h_Benutzer
        DBMS SQL SELECT benutzername benutzer_feld_1 \
                        FROM benutzer \
                        WHERE benutzer_feld_1=:+h_PESNr
        h_MaxCount = @dmrowcount
        DBMS ALIAS
        if (h_MaxCount > 0)
            return h_Benutzer
    }
    return ""
}
} // end of ErmittlungBenutzer
```

1.6.9. fillWmFromDbField

This function determines values for a value set from a database based on specified criteria (table, column, value key). The value set is then filled with values from another column.

Note

```

//*****
// * Funktion zum Ermitteln von Werten (--&gt; Wertemenge)
// * aus einer Datenbank anhand vorgegebener Kriterien
// * (Eingabe: Tabelle, Spalte, Wert (Schlüssel) --&gt;
// * Wertemenge wird mit
// * den Werten anderer Spalte gefüllt)
//*****

proc fillWmFromDbField (dbTable, dbColSrc, dbColTarget,
baseValue, \
                        h_repos_id )
{
  call api_log_info ("+++++++ HOOK: fillWmFromDbField START +++++
+++")
  vars h_dbValue, h_rowCount

  DBMS ALIAS h_dbValue
  DBMS SQL SELECT Count (*) FROM :dbTable WHERE :dbColSrc \
              like ':(baseValue)%'
  h_dbValue = @dmrowcount
  DBMS ALIAS

  if( h_dbValue > h_MAX_AMOUNT_VALUES )
  {
    d3server_value_char_allowed = 'Zu viele Werte'
    d3server_repos_id_allowed = ':h_repos_id'
    return
  }
  else if (h_dbValue == 0)
  {
    d3server_value_char_allowed = ''
    d3server_repos_id_allowed = ':h_repos_id'
    return
  }

  // wenn das Limit nicht überschritten worden ist,
  // dann zeige alle Möglichkeiten in der Wertemenge an
  DBMS ALIAS d3server_value_char_allowed,
d3server_repos_id_allowed
  DBMS SQL SELECT DISTINCT :dbColTarget, ':h_repos_id'
              FROM :dbTable WHERE :dbColSrc like ':(baseValue)%'
  DBMS ALIAS

  call api_log_info ("+++++++ HOOK: fillWmFromDbField ENDE +++++
++")
} // end of fillWmFromDbField

```

1.6.10. getValueFromDBField

This function queries a value from the database based on specified criteria (table, column, value, key). Then, a value from another column is returned.

Note

```

//*****
// * Funktion zum Ermitteln eines Wertes aus einer Datenbank
// * anhand
// * vorgegebener Kriterien (Eingabe: Tabelle, Spalte, Wert
// * (Schlüssel)
// * --> Ausgabe: Wert anderer Spalte)
//
//*****

string proc
getValueFromDBField (dbTable, dbColSrc, dbColTarget, \
                     baseValue )
{
    call
    api_log_info ("+++++++ HOOK: getValueFromDBField START ++++++")

    vars h_dbValue, h_rowCount, dummy

    //
    // Wenn Datensätze doppelt auftreten (Schlüssel): bei der SQL Count
    // Abfrage wird gruppiert (Schlüssel) und die Anzahl der
    //
    // zurückgegebenen Datensätze (aus 2 mach 1) ausgewertet (h_dbValue)
    DBMS ALIAS
    DBMS SQL SELECT :dbColSrc, Count(*) \
                FROM :dbTable \
                WHERE :dbColSrc = :+baseValue \
                GROUP BY :dbColSrc
    h_dbValue = @dmrowcount
    DBMS ALIAS

    if (h_dbValue > 1)
    {
        return h_ERR_TOO_MANY_VALUES
    }

    // Wenn es einen oder keinen Wert gibt
    DBMS ALIAS h_dbValue
    DBMS SQL SELECT DISTINCT :dbColTarget FROM :dbTable WHERE \
                :dbColSrc = :+baseValue
    DBMS ALIAS

    call api_log_info
    ("+++++++ HOOK: getValueFromDBField ENDE ++++++")
    return h_dbValue
}

```

1.6.11. hasDtsReposId

This function determines, if a document type contains a repository field based on the repository ID.

Note

```
//*****
// * Funktion um herauszufinden, ob eine Dokumentart ein
Repository
// * besitzt (anhand der Repository ID)
//
//*****
*

proc hasDtsReposId (h_dts, h_repos_id)
{
call api_log_info "+++++++ HOOK: hasDtsReposId START +++++++
+")
// Ermittle, ob es in der DB einen Eintrag gibt, der besagt,
// ob ein Repository einer Dokumentart zugewiesen ist
vars h_count = 0, retVal = 0
DBMS ALIAS h_count
DBMS SQL SELECT Count(*) \
    FROM fispe_titel_dokuart \
    WHERE kue_dokuart = :h_dts AND repository_id = :h_repos_id
DBMS ALIAS

    // Es gibt eine Zuweisung...
    if (h_count > 0)
    {
        retVal = 1
    }
    // Es gibt KEINE Zuweisung ...
    else
    {
        retVal = 0
    }
call api_log_info ("+++++++ HOOK: hasDtsReposId ENDE +++++++
+")
    return retVal
}
```

1.6.12. AvailableValues

This function provides a value set with all values that already exist in the d.3 database for this `dok_dat_feld` field. Values entered in the field `dok_dat_feld` are integrated into the query (with `LIKE 'xxx%'`).

Note

```

//
*****
// * 20.06.2006
// * Wertemenge, die alle Werte liefert, die bereits zu diesem
//
* dok-dat-feld (Dokuart abhängig) im d3 System vorhanden sind;
//
* Eingegebene Werte in dem Dok_dat_feld werden mit in den Query
// * eingebaut (mit LIKE 'xxx%')
//
*****

proc AvailableValues (h_ReposID, h_User, h_DoctypeShort)
{
    call api_log_info ("+++++++ HOOK: DynWM
                        AvailableValues START ++++++")

    vars h_ddf
    vars h_AnzRow, h_LikeClause = ""
    vars h_MaxCount = 1500

    // ermittle dok_dat_feld zu der repos_id

    DBMS ALIAS h_ddf
    DBMS SQL SELECT DISTINCT doc_field_nr_pref FROM >
                        field_repository WHERE \
                        repository_id = :h_ReposID
        h_AnzRow = @dmrowcount
    DBMS ALIAS

    // ermittle Anzahl der gefundenen Werte
    // (d.h. Größe der Wertemenge)
    // Eingabe im dokdatfeld mit in den Query eingebaut
    // (für Einschränkungen)

    if (dok_dat_feld[:h_ddf] != "")
        h_LikeClause = "and dok_dat_feld:h_ddf LIKE ':
                        (dok_dat_feld[:h_ddf])%'"

    DBMS ALIAS h_AnzRow
    DBMS SQL SELECT count (DISTINCT dok_dat_feld:h_ddf) \
                        FROM firmen_spezifisch \
                        WHERE kue_dokuart = ':h_DoctypeShort' :h_LikeClause
    DBMS ALIAS

        // wenn das Limit überschritten worden ist,
        // dann wird ein Hinweis als
        // Wertemengeninhalte ausgegeben (macht Sinn bei sehr
        // großen Wertemengen,
        // wie z.B. 10000 Einträgen oder mehr)

    if( h_AnzRow > h_MaxCount )
    {

```

```

        call api_log_info ("Mehr als :h_MaxCount Werte gefunden
                           :h_AnzRow")
        d3server_value_char_allowed = "Mehr als
                                       :h_MaxCount Werte gefunden :h_AnzRow"
        d3server_repos_id_allowed   = h_ReposID
        return
    }
    else if( h_AnzRow == 0 )
    {
        call api_log_info ("Keinen Wert gefunden")
        d3server_value_char_allowed
= 'Es sind keine Werte vorhanden.'
        d3server_repos_id_allowed   = h_ReposID
        return
    }

    // wenn das Limit nicht überschritten worden ist,
    // dann zeige alle Möglichkeiten
    // in der Wertemenge an Eingabe im dokdatfeld mit in den
    // Query eingebaut (für Einschränkungen)

    DBMS
    ALIAS d3server_value_char_allowed, d3server_repos_id_allowed
    DBMS SQL SELECT DISTINCT dok_dat_feld:h_ddf, ':h_ReposID' \
           FROM firmen_spezifisch \
           WHERE kue_dokuart = ':h_DoctypeShort' :h_LikeClause
    DBMS ALIAS
    call api_log_info ("+++++++ HOOK: >
                      DynWM AvailableValues ENDE ++++++")
} // end of AvailableValues

```

1.7. Webservice functions

d.3 webservice functions

Note

Unless otherwise stated, all functions return the string OK by success, else an error message occurs.

Initialization	
d3ws_startSession()	Initialization, returns a session ID (actually without usage)
d3ws_stopSession()	Completion (optional, applies to new initialization)

Creation of XML (-> XML-Out-Buffer)	
d3ws_xml_start()	Beginning of a new XML document, deleting of the previous buffer
d3ws_xml_openNode(name)	Opens a new node, that must be closed with CloseNode again (also "<name>")
d3ws_xml_addNode(name, value)	Creates a complete node (also "<name>value</name>")
d3ws_xml_closeNode(name)	Closes a node, that was opened with OpenNode (also "</name>")
d3ws_xml_stop()	Completes the XML-document to be saved now or to be sent

Load, Save, call web-service	
<code>d3ws_xml_saveToFile (filename)</code>	Saves the XML-Out-Buffer in the specified file
<code>d3ws_xml_loadFromFile (filename)</code>	Loads the specified file into the XML-In-Buffer
<code>d3ws_http_addParameter (name, value)</code>	adds a parameter, which is passed at a later call of <code>sendRequest</code> as POST-Parameter
<code>d3ws_http_setTraceFile (filename)</code>	protocols the following call of <code>sendRequest</code> in a file (useful for troubleshooting)
<code>d3ws_http_sendRequest (url)</code>	Sends the XML-Out-Buffer to the respective URL and loads the result of the request into the XML-In-Buffer. GET-parameters can be used in the URL. If previously no further parameter was passed (<code>addParameter</code>), then the XML will be passed nameless via POST. Or else the XML will be passed as parameter <code>"xml="</code> via POST.

XML-Parser (-> XML-In-Buffer)	
Navigation	
<code>d3ws_xml_setRootNode()</code>	Sets the focal point to the root-node of the XML-document.
<code>d3ws_xml_setParentNode()</code>	Sets the focal point to the parent-node of the previously focussed node.
<code>d3ws_xml_setNode (expression)</code>	Sets the focal point to any node (relative to the current focussed node). expression can be: • <code>" / "</code> for the root-node • <code>" . "</code> for the parent-node • <code>"1" - "n"</code> for the "n"-th child-node (i.a. count previously with <code>countNodes</code>) • <code>"name"</code> for a specific child-node any combination of the fields above (as path definition, separated with <code>" / "</code>)
<code>d3ws_xml_setNodeByIndex (i)</code>	Analog to <code>setNode ("i")</code> , but an integer can be passed here (<code>setNode</code> expects a string)
Analysis	
<code>d3ws_xml_countNodes()</code>	Provides the count of available child-nodes below the currently focussed node.
<code>d3ws_xml_getNodeName (i)</code>	Provides the name of the i-node.
Access to included values	
<code>d3ws_xml_getNodeValue (expression)</code>	Provides the content of the node referred by the expression (relative to the currently focussed node). Using the expression as in <code>setNode</code>.

1.8. Additional information sources and imprint

If you want to deepen your knowledge of d.velop software, visit the d.velop academy digital learning platform at <https://dvelopacademy.keelelearning.de/>.

Our E-learning modules let you develop a more in-depth knowledge and specialist expertise at your own speed. A huge number of E-learning modules are free for you to access without registering beforehand.

Visit our Knowledge Base on the d.velop service portal. In the Knowledge Base, you can find all our latest solutions, answers to frequently asked questions and how-to topics for specific tasks. You can find the Knowledge Base at the following address: <https://kb.d-velop.de/>

Find the central imprint at <https://www.d-velop.com/imprint>.