

d.3 Server API - Entwicklerdokumentation





Inhaltsverzeichnis

1	Impressum/Rechtliche Hinweise	7
2	Einleitung	9
2.1	Vereinbarungen	9
2.2	Über die d.3 Server API	10
2.3	Voraussetzungen	10
3	Allgemeine Informationen zur d.3 Server API	11
3.1	Überblick	11
3.1.1	api_hook_user_set (user_name)	11
3.1.2	api_hook_user_reset ()	12
3.1.3	api_hook_user_get ()	12
3.1.4	api_hook_user_master()	12
3.1.5	api_set_dok_dat_feld ()	13
3.1.6	api_set_api_doc_field ()	13
3.1.7	api_clear_document_context()	13
3.1.8	api_clear_user_context()	13
3.2	Funktionsaufruf	14
3.3	Logging	14
3.4	Beispiele	16
3.4.1	Beispiel 1	16
3.4.2	Beispiel 2	17
4	Globale Variablen	19
4.1	Dokumentkontext	19
4.2	Benutzerkontext	21
4.3	Objektkontext	22
4.4	Workflowkontext	22
4.5	Postkorbkontext	23
4.6	Sperrkontext	23
4.7	Anwendungskontext	24
4.8	Sonstige Variablen	24
4.9	Error Handling	25
5	Funktionen	26
5.1	Übersicht über die Funktionen	26
5.1.1	Dokument (Übersicht)	26
5.1.2	Akte (Übersicht)	27
5.1.3	Verknüpfungen (Übersicht)	27
5.1.4	Verzeichnisse (Übersicht)	27
5.1.5	Status (Übersicht)	28
5.1.6	Notizen (Übersicht)	28
5.1.7	Wiedervorlage (Übersicht)	28
5.1.8	Workflow (Übersicht)	28
5.1.9	Dokumentart (Übersicht)	29
5.1.10	Wertemengen (Übersicht)	29
5.1.11	Benutzer (Übersicht)	29
5.1.12	Gruppen (Übersicht)	30
5.1.13	Rollen (Übersicht)	30
5.1.14	Restriktionsmengen (Übersicht)	30
5.1.15	TIFF-Funktionen (Übersicht)	30
5.1.16	Dateizugriffe (Übersicht)	30
5.1.17	Stringkonvertierung (Übersicht)	31



5.1.18 Diverses (Übersicht)	32
5.1.19 D3fc-Calls (Übersicht)	32
5.1.20 JAVA (Übersicht)	33
5.2 Dokument	33
5.2.1 attribute_get_single	33
5.2.2 attribute_get_all	37
5.2.3 attribute_update_single	39
5.2.4 attribute_update_all	46
5.2.5 attribute_get_valid_values	50
5.2.6 attribute_test_value	52
5.2.7 attribute_update_begin	54
5.2.8 attribute_update_execute	55
5.2.9 document_change_type	56
5.2.10 document_delete	57
5.2.11 document_find_ids	60
5.2.12 document_get_permission	62
5.2.13 document_publish_for_web	64
5.2.14 document_dependent_add	65
5.2.15 document_dependent_delete	67
5.2.16 document_register_dependent	68
5.2.17 document_get_file_path	69
5.2.18 document_start_lifetime	70
5.2.19 document_set_cache_days	71
5.2.20 document_send_to_dsearch	72
5.2.21 add_inherit_doc_rights	73
5.2.22 remove_inherit_doc_rights	75
5.3 Akte	76
5.3.1 Akte	76
5.3.2 folder_create	76
5.4 Verknüpfungen	77
5.4.1 link_get_parents	77
5.4.2 link_get_children	79
5.4.3 link_documents	81
5.4.4 link_exists	84
5.4.5 link_delete	85
5.5 Verzeichnisse	87
5.5.1 directory_create	87
5.5.2 directory_delete	88
5.5.3 directory_exist	89
5.5.4 directory_list	90
5.5.5 directory_count_files	92
5.6 Status	93
5.6.1 document_block	93
5.6.2 document_verify	96
5.6.3 document_transfer	98
5.7 Notizen	103
5.7.1 note_add_file	103
5.7.2 note_add_string	105
5.8 Wiedervorlage	107
5.8.1 hold_file_send	107
5.8.2 hold_file_find	112
5.8.3 hold_file_delete	114
5.8.4 hold_file_get_document_info	116
5.9 Workflow	119
5.9.1 workpath_get_document_info	119



5.9.2 workpath_end_document	120
5.9.3 workpath_start_document	122
5.9.4 workpath_go_to_next_step	123
5.10 Dokumentart	125
5.10.1 document_type_get_long	125
5.10.2 document_type_get_short	127
5.11 Wertemengen	129
5.11.1 predefined_values_get	129
5.11.2 predefined_values_test	132
5.12 Benutzer	133
5.12.1 user_get	133
5.12.2 user_is_system	134
5.12.3 user_get_groups	136
5.12.4 user_get_verifier_groups	138
5.12.5 user_get_rolls	140
5.12.6 user_get_doc_types	141
5.12.7 user_get_data	142
5.12.8 user_validate_password	146
5.12.9 user_get_favourites	147
5.12.10 user_get_sid_rights	148
5.12.11 user_checkout	148
5.12.12 user_get_superior	149
5.13 Gruppen	150
5.13.1 group_get	150
5.13.2 group_get_members	151
5.13.3 group_get_verifiers	154
5.13.4 group_get_doc_types	155
5.14 Rollen	157
5.14.1 roll_get	157
5.14.2 roll_get_names	158
5.14.3 roll_get_users	159
5.14.4 roll_get_data	161
5.15 Restriktionsmengen	164
5.15.1 d3set_add_filter	164
5.15.2 d3set_remove_filter	165
5.15.3 d3set_get_filter	165
5.16 Tiff-Funktionen	167
5.16.1 document_render	167
5.16.2 tiff_create	169
5.16.3 tiff_concat	170
5.16.4 document_render_wfl_prot	171
5.17 Dateizugriffe	172
5.17.1 file_add_string	172
5.17.2 file_add_line	174
5.17.3 file_read_line	176
5.17.4 file_concat	176
5.17.5 file_copy	177
5.17.6 file_delete	178
5.17.7 file_compare	179
5.17.8 file_rename	181
5.17.9 file_exist	182
5.18 Stringkonvertierung	183
5.18.1 string_to_lower	183
5.18.2 string_to_upper	184
5.18.3 string_to_number	186



5.18.4 string_to_date	186
5.18.5 string_split	187
5.18.6 string_cut_leading_char	189
5.18.7 string_fill_leading_char	190
5.18.8 string_fill_trailing_char	192
5.18.9 string_replace	193
5.18.10 string_get_position	194
5.18.11 string_mask_special_char	196
5.18.12 api_str_sub	196
5.18.13 api_str_sub_utf8	197
5.18.14 api_str_sub_ansi	197
5.18.15 api_str_len	197
5.18.16 api_str_len_utf8	198
5.18.17 api_str_len_ansi	198
5.18.18 api_str_to_utf8	199
5.18.19 api_str_from_utf8	199
5.18.20 api_str_is_utf8	200
5.18.21 api_str_is_ascii	200
5.18.22 api_str_cut_fract_utf8	201
5.18.23 api_str_hash	202
5.19 Diverses	203
5.19.1 restore_from_jukebox	203
5.19.2 restore_from_history	205
5.19.3 send_email	206
5.19.4 lock_token_acquire	209
5.19.5 lock_token_release	210
5.19.6 start_ext_program	211
5.19.7 get_current_datetime	212
5.19.8 sleep	215
5.19.9 process_get_id	215
5.19.10 regular_expression_test	216
5.19.11 object_property_set	218
5.19.12 object_properties_get	219
5.19.13 set_signature_data	221
5.20 d3fc-Calls	222
5.20.1 d3fc_user_set	223
5.20.2 d3fc_password_set	224
5.20.3 d3fc_remote_server_set	224
5.20.4 d3fc_port_set	224
5.20.5 d3fc_server_set	225
5.20.6 d3fc_timeout_set	225
5.20.7 d3fc_function_name_set	225
5.20.8 d3fc_importing_set	226
5.20.9 d3fc_exporting_set	226
5.20.10 d3fc_exporting_get	227
5.20.11 d3fc_table_set_import_file	227
5.20.12 d3fc_table_set_export_file	228
5.20.13 d3fc_table_set_headline	228
5.20.14 d3fc_table_add	229
5.20.15 d3fc_field_set	230
5.20.16 d3fc_field_get	230
5.20.17 d3fc_first	231
5.20.18 d3fc_next	231
5.20.19 d3fc_execute	232
5.20.20 Beispiele	233



5.21 JAVA	241
5.21.1 java_add_param	241
5.21.2 java_clear_param	241
5.21.3 java_call_static	242
6 JPL-Styleguide	244
6.1 Ablage und Name	244
6.2 Kommentare	244
6.3 Kopf	244
6.4 Variablen	246
6.4.1 Generell	246
6.4.2 Repositoryfelder und Dokumentarten	247
6.5 Debugging	247
6.6 Schleifen und Verzweigungen	248
6.6.1 For-Schleife	248
6.6.2 While-Schleife	249
6.6.3 If-Abfrage	249
7 Nützliche Funktionen	250
7.1 WriteNotizEntry	250
7.2 AddPrefixAndZeroes	250
7.3 CutPrefixAndZeros	251
7.4 OpenSessionConnection	251
7.5 GruppeAnwesend	253
7.6 BenutzerAnwesend	253
7.7 BenutzerIstGruppe	254
7.8 ErmittlungBenutzer	254
7.9 fillWmFromDbField	255
7.10 getValueFromDBField	256
7.11 hasDtsReposId	257
7.12 AvailableValues	258
8 Webservice-Funktionen	260
9 Index	262

1 Impressum/Rechtliche Hinweise

Alle bisherigen Dokumentationen zu d.3 server API verlieren mit der Veröffentlichung dieser Dokumentation ihre Gültigkeit.

Die d.velop AG behält sich vor, Komponenten von Drittanbietern, die in den Produkten ggf. vorhanden sein können durch funktionsadäquate Komponenten anderer Hersteller zu ersetzen. Die d.velop AG behält sich in Ausübung Ihrer jeweils gültigen Releasepolitik vor, Produktfeatures und einzelne Softwareprodukte nicht mehr durch Softwarepflege- und Supportleistungen zu unterstützen. Näheres dazu finden Sie im Supportlebenszyklus des d.velop-Service-Portals unter www.d-velop.com/supportlebenszyklus.

Alle in dieser Dokumentation enthaltenen Informationen und Darstellungen wurden mit größter Sorgfalt durch unsere Qualitätssicherung nach dem allgemeinen Stand der erprobten Technik geprüft. Dennoch sind Fehler nicht auszuschließen. Aus diesem Grund sind die in der vorliegenden Dokumentation enthaltenen Informationen mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden, sofern wir mit Ihnen keine hiervon abweichenden vertraglichen Vereinbarungen getroffen haben. Die d.velop AG übernimmt infolgedessen keine Haftung/Gewährleistung einzig auf Grundlage dieser Dokumentation. Die d.velop AG wird insbesondere keine Haftungen für Schäden übernehmen, die aufgrund der Benutzung dieser Dokumentation oder Teilen davon entstehen, solange diese Dokumentationen nicht Bestandteil einer vertraglichen Beziehung mit Ihnen sind. Sofern eine vertragliche Vereinbarung mit Ihnen besteht, hat diese insbesondere im Hinblick auf das Haftungsregime stets Vorrang. Ansprüche nach dem Produkthaftungsgesetz sowie nach Deliktsrecht bleiben stets unberührt, sofern sie nicht individualvertraglich ausgeschlossen wurden.

Die d.velop AG behält sich vor, ihre Produkte und Dokumentationen fortwährend anzupassen. Ein Anspruch auf Lieferung von Softwareprodukten oder die Erbringung von Leistungen nur auf Grundlage dieser Dokumentation besteht grundsätzlich nicht.

Nach Redaktionsschluss können sich Änderungen bei den Softwareprodukten und Leistungen ergeben haben. Die Informationen und technischen Angaben zu den verschiedenen Softwareprodukten und den damit verbundenen technischen Spezifikationen in dieser Dokumentation stellen ohne eine gesonderte vertragliche Vereinbarung hierzu keine zugesicherte Eigenschaft oder Garantie dar, sei es ausdrücklich oder stillschweigend. Sie stellen insbesondere keine stillschweigende Zusage oder Garantie betreffend die Beschaffenheit, die Handelsfähigkeit, die Eignung für bestimmte Zwecke oder den Nichtverstoß gegen Gesetze und Patente dar. Aussagen über gesetzliche, rechtliche und steuerliche Vorschriften und deren Auswirkungen haben nur für die Bundesrepublik Deutschland Gültigkeit.

Diese Dokumentation unterliegt als Bestandteil der d.velop-Software den jeweils gültigen Lizenzbedingungen der d.velop AG für Partner und Endkunden. Die Dokumentation steht jederzeit zum Download im d.velop-Service-Portal zur Verfügung.

Diese Dokumentation mit ihren Texten, Bildern, Grafiken sowie ihren Arrangements und die beschriebenen Softwareprodukte unterliegen dem Urheberrecht und anderen Gesetzen zum Schutz geistigen Eigentums; es ist ein geschütztes Erzeugnis der d.velop AG.

Die Verwendung der Texte, Bilder, Grafiken sowie deren Arrangements, auch auszugsweise, sind ohne vorherige schriftliche Zustimmung der d.velop AG urheberrechtswidrig und strafbar. Dies gilt insbesondere für die Vervielfältigung, Übersetzung oder die Verwendung in elektronischen Systemen.

Alle verwendeten Hard- und Softwarenamen sind Handelsnamen und/oder Warenzeichen der jeweiligen Hersteller/Inhaber, die diese zur Verfügung gestellt haben.



Sofern Teile oder einzelne Formulierungen dieser Dokumentation der geltenden Rechtslage nicht, nicht mehr oder nicht vollständig entsprechen sollen, bleiben die übrigen Teile der Dokumentation in ihrem Inhalt und ihrer Gültigkeit davon unberührt.

Kontakt

d.velop AG

Schildarpstraße 6-8
48712 Gescher, Deutschland

Fon +49 (0) 2542 9307-0
Fax +49 (0) 2542 9307-6999

www.d-velop.com
info@d-velop.de

Vertreten durch den Vorstand: Christoph Pliete (Vorsitzender), Mario Döenzebrink, Burghardt Garske

Vorsitzender des Aufsichtsrates: Dr. Helmut Bäumer

Handelsregister beim Amtsgericht Coesfeld, Nr. HRB 4903

Umsatzsteueridentifikationsnummer: DE 813062165

Dieses Dokument wurde zuletzt am 01.07.2014 überarbeitet und bezieht sich auf d.3 server API ab Version 7.2.2.

Name des Dokuments: d3serverAPI.pdf (Buildnummer: 20140701)

Bei Fragen zu dieser Dokumentation oder zur Software wenden Sie sich bitte an uns.

Fon +49 (0) 2542 9307-6000
support@d-velop.de

Alle Rechte vorbehalten. Irrtümer vorbehalten.

Copyright © 2014 d.velop AG.

2 Einleitung

2.1 Vereinbarungen

Die in diesem Dokument eingesetzten Schriften und Symbole haben folgende Bedeutung:

Schrift / Symbol	Bedeutung
Extras Optionen	Menüpunkte oder Button Untermenüpunkte werden jeweils mit einem vertikalen Balken voneinander getrennt.
Extras Optionen [Sonstiges]	Der letzte Untermenüpunkt in eckigen [Klammern] steht für eine auszuwählende Registerkarte.
ALT-STRG-X	Taste oder Tastenkombination Bei einer Tastenkombination werden die Tasten jeweils mit einem Bindestrich voneinander getrennt. Die zuerst genannten Tasten müssen gedrückt bleiben, während die letzte betätigt wird.
d.3 smart explorer	Programmnamen erscheinen in dieser Schriftart.



Hinweis:

Bemerkungen mit diesem Zeichen sind besonders zu berücksichtigen, stellen hilfreiche Tipps oder Ausnahmen dar.



Achtung!

Bemerkungen mit diesem Zeichen helfen, Fehler zu vermeiden, die im schlimmsten Fall zu fehlerhaftem Programmverhalten oder dem Verlust von Daten führen können.



Beispiel:

So markierte Abschnitte enthalten Beispiel-Daten zu den erläuterten Funktionen.



Hintergrund-Informationen:

Bemerkungen mit diesem Zeichen enthalten Informationen, die einzelne Punkte näher erläutern und liefern weiterführende Details zum aktuellen Thema.



Dateinamen, Verzeichnis-Pfade oder (Programmier-) Code werden in dieser Schriftart dargestellt.

**Video:**

Unter dem hier aufgeführten Link finden Sie weitere Informationen im Videoformat.

2.2 Über die d.3 Server API



Langfristig sollen Hook-/Workflow-Programmierer so unterstützt werden, dass kein direkter Zugriff auf die d.3 Datenbank mehr nötig ist.

Dazu muss der Server ab Version 5.5.1 Funktionen bereitstellen, um

- dokumentbezogene Informationen (Attribute, Verknüpfungen ...)
- dokumentartbezogene Informationen (Repository ...)
- benutzerbezogene Informationen (Gruppenzugehörigkeit ...)

abfragen / ändern zu können.

Außerdem sollten Helferfunktionen zur Verfügung gestellt werden, die die Konvertierung von Strings / Datumsangaben / numerischen Angaben in das von d.3 bzw. der DB benötigte Format ausführen, sowie Dateizugriffe vereinfachen.

Damit kann eine Entkopplung der einzelnen Hooks von der d.3 Datenstruktur erfolgen. Der Programmierer bekommt eine Garantie für Langzeitstabilität, Support etc. und die d.3 Daten werden vor Fehlern besser geschützt und Zugriffe können protokolliert und damit besser debugged werden.

Die Funktionen sollen in Bezug auf Namen und Aufrufparameter ähnlich sein und somit eine einheitliche Schnittstelle zwischen Hook und d.3 bilden.

Die Funktionen sollen alle über eine zentrale Codestelle geführt werden, damit hier Standard-Loggings, Kontrollen etc. ausgeführt werden können.

2.3 Voraussetzungen



Dieses Handbuch wendet sich an d.3 Programmentwickler.
Grundlegende Kenntnisse über Programmierung und das d.3 System sind notwendige Voraussetzungen für das Verständnis dieses Handbuchs.



3 Allgemeine Informationen zur d.3 Server API

3.1 Überblick

Um die Übergabe von Werten möglichst einfach zu gestalten sind verschiedene Kontexte definiert, d.h. es gibt einen Satz an globalen Variablen, die gesetzt / gelesen werden können.

Vor dem Aufruf einer Funktion, kann beispielsweise die DocId in Form der Variablen `api_doc_id` und damit der Dokumentenkontext gesetzt werden. Die nachfolgenden Funktionen arbeiten dann alle im Kontext dieser DocId. Werden nun Operationen wie Attributänderungen etc. vorgenommen, beziehen sich diese immer auf diese DocId.

Gleiches gilt für den Benutzer. Auch dieser kann global eingestellt werden, so dass er nicht bei jedem Funktionsaufruf vorgegeben werden muss. Wird kein Benutzerkontext gewählt, so wird der Hook mit der Berechtigung des Standardbenutzers ausgeführt. Der Standardbenutzer hängt von der Konfigurationsvariablen `SET_API_USER_MASTER` ab. Ist diese Variable "1", so wird der "Master" als Default-Benutzer verwendet, d.h. der Hook hat alle Rechte, sofern dort nicht explizit ein anderer Benutzer gewählt wird, dessen Rechte genutzt werden. Ist diese Variable ="0", so wird der Benutzer genutzt, der den d.3-API-Call ausgeführt hat, der wiederum den Hook auslöste.

Für den Hook-Benutzer gibt es folgende Funktionen:

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

Alternativ zum Setzen des Kontexts können diese Angaben auch beim Funktionsaufruf direkt gemacht werden, so dass der aktuelle Kontext für diesen Funktionsaufruf ignoriert wird.

Um die Werte zwischen den globalen `dok_dat_feldern` und den `api_doc_field's` auszutauschen gibt es folgende Funktionen:

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

3.1.1 `api_hook_user_set (user_name)`

Diese Funktion setzt einen Benutzer als ausführenden Benutzer. Danach sollte ein `api_hook_user_reset` durchgeführt werden.

api_hook_user_set (user_name)	
Parameter	<code>user_name</code> Name des Benutzers, in dessen Namen nun gearbeitet wird

api_hook_user_set (user_name)

Rückgabe	0 fehlerfrei -1 user_name wurde nicht angegeben -2 user_name ist kein gültiger d.3 Benutzername
----------	---

3.1.2 api_hook_user_reset ()

Diese Funktion setzt wieder den Default-Benutzer als ausführenden Benutzer. Dies ist immer nach einem api_hook_user_set aufzurufen.

api_hook_user_reset ()

Parameter	keine
Rückgabe	Name des Benutzers, der zuvor als Benutzer verwendet wurde.

3.1.3 api_hook_user_get ()

Die Funktion ermittelt den ausführenden Benutzer.

api_hook_user_get ()

Parameter	keine
Rückgabe	Name des ausführenden Benutzers



```
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
```

3.1.4 api_hook_user_master()

Diese Funktion setzt den "Master" als ausführenden Benutzer.

api_hook_user_master()

Parameter	keine
Rückgabe	keine



seit d.3 Version 6.0 HF2

3.1.5 api_set_dok_dat_feld ()

Setzt die globalen `dok_dat_felder` mit den Werten der `api_doc_fields` (z. B. für `attribute_get_valid_values`)

api_set_dok_dat_feld ()

Parameter	keine
Rückgabe	0 fehlerfrei

3.1.6 api_set_api_doc_field ()

Setzt die `api_doc_fields` mit den Werten der globalen `dok_dat_felder` (z.B. für `attribute_get_valid_values`)

api_set_api_doc_field ()

Parameter	keine
Rückgabe	0 fehlerfrei

3.1.7 api_clear_document_context()

leert alle dokumentbezogenen Felder

api_clear_document_context()

Parameter	keine
Rückgabe	0 fehlerfrei

3.1.8 api_clear_user_context()

leert alle benutzerbezogenen Felder

api_clear_user_context()

Parameter	keine
Rückgabe	0 fehlerfrei

3.2 Funktionsaufruf

Um eine Server-Api-Funktion zu verwenden, ist folgende Funktion aufzurufen:

api_function(name, param1, param2, param3, param4 ...)	
Parameter	name: Name der Funktion, die verwendet werden soll param1..n: Parameter, die für die Funktion benötigt werden
Rückgabe	0: alles in Ordnung, keine Fehler >0: Datenbankfehler -1: Panther Syntax Fehler, Details siehe LogViewer -2: Funktion wurde falsch aufgerufen -3: unbekannter Fehler, d.velop Support kontaktieren



sonst: siehe die Beschreibung der jeweiligen Funktion

Dort sind auch die Informationen erläutert, die die globalen Fehlervariablen liefern können, neben den Übergabeparametern, die sie immer enthalten. Die Werte sind an die der API angelehnt.

Übergabeparameter	
api_error_text	Fehlertext
api_error_var[1..n]	Variablenname
api_error_value[1..n]	Wert der Variablen n ist in der Variablen MAX_ERROR_VAR definiert

3.3 Logging

Für das Logging steht allgemein die Funktion `api_log` zur Verfügung.

api_log(log_level, log_message)	
Parameter	log_level: Ausgabelevel, typische d.3 Ausgabelevel 0..9 definierte Konstanten: LOG_DEBUG 9 LOG_INFO 6 LOG_ERROR 0 log_message auszugebender Text
Rückgabe	0: fehlerfrei

Für bestimmte Loglevel sind weitere Funktionen definiert, die stattdessen benutzt werden können und die auf die oben beschriebene allgemeine Logfunktion führen:

Logfunktion	
api_log_debug(log_message)	Debug-Logging
api_log_info(log_message)	Info-Logging
api_log_error(log_message)	Fehler_Logging

3.4 Beispiele

3.4.1 Beispiel 1

Der Beispielhook sendet nach einer erfolgreichen Archivierung eines Dokumentes eine Wiedervorlage an einen User.



```
-----  
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
```

3.4.2 Beispiel 2

Diese Hookfunktion schreibt während der Archivierung des Dokumentes den Inhalt von `h_Worte` in das Feld 63.



```
proc InsertEntry_10 (h_User, h_DocTypeShort)
{
    // Variablen Deklaration
    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
```



```
proc InsertEntry_10 (h_User, h_DocTypeShort)
{
    // Variablen Deklaration
    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
```

4 Globale Variablen

4.1 Dokumentkontext

Variable	Beschreibung
api_doc_id	Dokumenten ID
api_doc_texts[1..4]	Bemerkungstext der einzelnen Zeilen
api_doc_type_short	Dokumentartkürzel
api_doc_field [1..89]	firmenspezifische Felder
api_doc_field_60..69[1..2000]	firmenspezifische Mehrfachattributfelder
api_doc_date_import	Importdatum
api_doc_last_alter_no	letzte Änderungsnummer
api_doc_publ_alter_no	Änderungsnummer in Freigabe
api_doc_status	Status={"Bearbeitung", "Prüfung", "Freigabe", "Archiv"} (in Version 5.5.1.x: doc_state)
api_doc_status_info	Zusatzinfo zum Status für doc_status="Prüfung" = { 0 = ungeprüft, 1 = geprüft}, für doc_status="Freigabe" = { 0 = gesperrt, 1 = freigegeben}
api_doc_editor	Bearbeiter
api_doc_no	Dokumentnummer
api_doc_var_no	Variantennummer
api_doc_text	

Variable	Beschreibung
	<p>Bemerkungstext</p> <div style="border: 2px solid red; padding: 10px;">  <p>Das globale Feld <code>api_doc_text</code> wird nicht mehr unterstützt. Auf Nicht-Unicode-Systemen kann dieses Feld zwar noch wie bisher verwendet werden (die Abwärtskompatibilität bleibt erhalten) aber auf Unicode-Systemen kann auf dieses Feld weder lesend noch schreibend zugegriffen werden.</p> </div> <div style="border: 2px solid blue; padding: 10px; margin-top: 10px;">  <p>Verwenden Sie stattdessen die Variable <code>api_doc_texts[1..4]</code>.</p> </div>
<code>api_doc_ids[MAX_DOC_IDS]</code>	Dokument-ID-Array
<code>api_doc_owner</code>	Besitzer
<code>api_doc_file_name</code>	Dateiname
<code>api_doc_file_ext</code>	Dateierweiterung
<code>api_doc_encrypt</code>	Verschlüsselung (0/1)
<code>api_doc_cnt_access</code>	Anzahl Zugriffe
<code>api_doc_date_last_access</code>	Datum des letzten Zugriffs
<code>api_doc_user_last_access</code>	Benutzer des letzten Zugriffs
<code>api_doc_last_update_file</code>	Zeitstempel der letzten Änderung
<code>api_doc_web_published</code>	webveröffentlicht (0/1)
<code>api_doc_workflow</code>	in einem Leitweg befindlich (0/1)
<code>api_links[MAX_LINKS]</code>	über-/untergeordnete Dokumente
<code>api_link_creators[MAX_LINKS]</code>	Erzeuger der Verknüpfung (a: Aktenplan, m: manuell, l: a00-Datei) (für <code>link_get_children</code> und <code>link_get_parents</code>)
<code>api_link_tstamps[MAX_LINKS]</code>	Zeitstempel, wann die Verknüpfung angelegt wurde (für <code>link_get_children</code> und <code>link_get_parents</code>)
<code>api_doc_types[MAX_DOC_TYPES]</code>	Dokumentarten
<code>api_classes[MAX_CLASSES]</code>	Array mit allen zugeordneten Klassen

Variable	Beschreibung
api_class_right[MAX_CLASSES]	<p>Array mit den zugeordneten Rechten für die Klasse mit gleichem Index in <code>api_classes</code> (0=verweigert, 1=erteilt, 2=keine Angabe)</p> <p>Position 1: Lesen Position 2: Schreiben Position 3: Prüfen Position 4: Freigeben</p>
api_color_code	<p>Farbcod des Dokuments 0: nicht gesetzt 1-24: Farbcod</p> <div style="border: 1px solid blue; padding: 5px; margin-top: 10px;">  neue Variable ab Version 7.2 </div>

4.2 Benutzerkontext

Variable	Beschreibung
api_users[MAX_USERS]	Benutzernamen
api_groups[MAX_GROUPS]	Gruppennamen
api_rolls[MAX_ROLLS]	Benutzernamen
api_user_name	Benutzername
api_user_right_level	Rechtestufe (1..5)
api_user_right_archive	Recht zur Ansicht von Archivversionen
api_user_right_link	Recht zur Aktenverknüpfung
api_user_right_flags	Rechteflags Position 1: Verknüpfungen erstellen Position 2: Webzugriff erlaubt Position 3: Benutzer deaktiviert Position 4: Benutzer gesperrt
api_user_field[10]	benutzerbezogene Felder
api_user_checkout_flag	Benutzer ist abwesend (0/1: nein/ja)
api_user_checkout_text	Abwesenheitstext des Benutzers
api_user_realname	Name
api_user_email	eMail Adresse

Variable	Beschreibung
api_user_phone	Telefonnummer
api_user_plant	Werk
api_user_department	Abteilung
api_user_tstamp_current_login	Zeitstempel des aktuellen Logins
api_user_tstamp_last_login	Zeitstempel des letzten Logins
api_user_tstamp_hf_info	Zeitstempel der Information über neue Wiedervorlagen
api_user_ldap_dn	LDAP Name
api_user_change_count	Änderungszahl
api_user_pw_valid_until	Datum, bis zu dem das aktuelle Paßwort gilt
api_user_tstamp_login_failed	Zeitstempel des Loginversuchs mit falschem Paßwort
api_user_count_login_failed	Anzahl der Anmeldeversuche mit falschem Passwort
api_roll_id	ID der Rolle
api_roll_name	Name der Rolle

4.3 Objektkontext

Variable	Beschreibung
api_property_name	Eigenschaftsname
api_property_caption	Kurzbeschreibung der Eigenschaft
api_property_description	Ausführliche Beschreibung der Eigenschaft
api_property_value	Wert der Eigenschaft
api_object_id	Objekt-ID der Eigenschaft (z.B. Benutzername, Gruppenname, ...)

4.4 Workflowkontext

Variable	Beschreibung
api_wfl_header_id	ID des Leitweges
api_wfl_step_header_id	ID des Leitwegschrittes
api_wfl_step_exit_string	wfl_step_exit_string
api_wfl_step_tstamp_entry	Zeitstempel des Erreichens des Schrittes

Variable	Beschreibung
api_wfl_header_name	LeitwegName
api_wfl_step_recipient	StepEmpfänger
api_wfl_step_type	StepTyp
api_wfl_step_action_type	Aktionstyp
api_wfl_step_action_string_1	Option1 der Aktion
api_wfl_step_action_string_2	Option2 der Aktion

4.5 Postkorbkontext

Variable	Beschreibung
api_hold_file_date	Datum der Wiedervorlage
api_hold_file_recipient	Empfänger
api_hold_file_notice	Betrefftext
api_hold_file_informed	Empfänger benachrichtigt (0/1)
api_hold_file_tstamp_sent	Zeitstempel "Gesendet"
api_hold_file_tstamp_ack	Zeitstempel "Quittiert"
api_hold_file_date_ack_until	Datum "Quittieren bis"
api_hold_file_sender	Sender
api_hold_file_date_remember	Datum "Erinnern am"
api_hold_file_deleted_sender	vom Sender gelöscht (0/1)
api_hold_file_chain_id	Ketten ID
api_hold_file_type	Typ (W=Workflow)

4.6 Sperrkontext

Variable	Beschreibung
api_lock_token	wurde erfolgreich eine neue Sperre angelegt, wird das zugehörige Token zurückgeliefert; damit kann die Sperre verlängert oder aufgehoben werden
api_lock_ttl	tatsächliche Gültigkeitsdauer der Sperre in Minuten
api_lock_object_info	schlug das Anlegen der Sperre fehl, da das Objekt schon anderweitig gesperrt ist, wird hier der mit der Sperre angelegte Info-Text zurückgeliefert

Variable	Beschreibung
api_lock_user	schlug das Anlegen der Sperre fehl, da das Objekt schon anderweitig gesperrt ist, wird hier der User zurückgeliefert, der die Sperre angelegt hat

4.7 Anwendungskontext

In die globale Variable `d3server_letzte_api_applikat` wird die Applikations-Id der Anwendung geschrieben, die den Hook angestossen hat.

Dabei sind die ersten drei Zeichen wichtig. Eine Funktion zur Ermittlung der Applikationsid könnte folgendermaßen aussehen:

```
$_
string proc d3GetLetzteApiApplikation( h_FullFlag )
{
    vars h_TmpLetzteApiApplikation = ""

    if ( h_FullFlag != 1 )
        h_TmpLetzteApiApplikation = d3server_letzte_api_applikat(1,3)
    else
        h_TmpLetzteApiApplikation = d3server_letzte_api_applikat

    return h_TmpLetzteApiApplikation
}
```

Sie können in einem Hook per IF-Anweisung darauf reagieren. Wenn Sie den Hook aus dem d.3 smart explorer anstossen, wäre der Wert "200" und beim d.3 import "220".



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

if(applikationsid==220)// d.3 Import
{
}
```

4.8 Sonstige Variablen

Sonstige Variablen	Beschreibung
api_converted_string	nimmt den konvertierten String der Funktionen <code>string_...</code> auf

Sonstige Variablen	Beschreibung
api_splitted_string [MAX_SPLIT]	enthält die Teilstings der Funktion <code>string_split</code>
api_single_info	enthält einen einzelnen Rückgabestring (z.B. einzelnes Attribut)
api_data_type	Datentyp bei Wertemengen
api_list[]	Array zur Aufnahme der Einträge von <code>directory_list</code> ab d.3 Version 6.0.1 Hotfix 1
MAX_DOC_IDS	Größe des DokID-Arrays
RIGHT_ANY	iegliches Recht
RIGHT_READ	Leserecht
RIGHT_WRITE	Schreibrecht
RIGHT_VERIFY	Prüfungsrecht
RIGHT_RELEASE	Freigaberecht

4.9 Error Handling

Variable	Beschreibung
MAX_ERROR_VAR	Größe der Fehler-Arrays
api_error_text	Fehlerbeschreibung
api_error_var[MAX_ERROR_VAR]	Fehlervariablen-Array
api_error_value[MAX_ERROR_VAR]	Fehlervariablenwerte-Array
LOG_ERROR = 0	Fehler
LOG_INFO = 6	Info
LOG_DEBUG = 9	Debug
additional_info_text	Dieser Text wird als zusätzlicher Exportparameter bei dem aktuellen API-Call an den Client übergeben.

5 Funktionen

5.1 Übersicht über die Funktionen

- Dokument
- Akte
- Verknüpfungen
- Verzeichnisse
- Status
- Notizen
- Wiedervorlage
- Workflow
- Dokumentart
- Wertemengen
- Benutzer
- Gruppen
- Rollen
- TIFF-Funktionen
- Dateizugriffe
- Stringkonvertierung
- D3fc-Calls
- Diverses

5.1.1 Dokument (Übersicht)

Dokumentfunktionen	Beschreibung
attribute_get_single	einzelnes Attribut
attribute_get_all	alles über Kontext
attribute_update_single	einzelnes Attribut
attribute_update_all	alles über Kontext
attribute_test_value	Wert auf Gültigkeit prüfen
attribute_update_begin	schreibt die aktuellen Attribute des Dokuments in die api_doc_fields
attribute_update_execute	verändert die api_doc_fields nach Aufruf von attribute_update_begin
document_change_type	Dokumentart wechseln
document_get_valid_values	erlaubte Kenndaten abfragen
document_delete	Dokument löschen
document_find_ids	Dokumente zu Kontext suchen

Dokumentfunktionen	Beschreibung
document_get_permission	Abfrageberechtigungen eines Benutzers auf ein Dokument
document_publish_for_web	markiert ein Dokument als (Nicht-) Web-Veröffentlicht
document_dependent_add	legt ein abhängiges Dokument ab bzw. überschreibt ein existierendes
document_dependent_delete	löscht ein abhängiges Dokument
document_register_dependent	registriert die abhängigen Dateien eines Dokuments
document_get_file_path	ermittelt zu vorgegebenen Parametern den kompletten Pfad im d.3 -Dokbaum unter dem das Dokument stehen müsste
document_start_lifetime	setzt die Lebensdauer eines Dokuments
document_set_cache_days	setzt die Anzahl an Tagen - heute beginnend - , die das Dokument im Dokumentenbaum (Fileserver) verbleibt; die Anzahl an Tagen ist nur relevant, wenn das Dokument auf den Sekundärspeicher ausgelagert ist/wurde
document_send_to_dsearch	sendet OCR-Informationen einer Dokumentversion an d.3 search
add_inherit_doc_rights	Funktion zur Vererbung von Rechten auf ein Dokument
remove_inherit_doc_rights	Funktion zum Löschen von vererbten Rechten auf ein Dokument

5.1.2 Akte (Übersicht)

Aktenfunktionen	Beschreibung
folder_create	Akte anlegen

5.1.3 Verknüpfungen (Übersicht)

Verknüpfungsfunktionen	Beschreibung
link_get_parents	übergeordnete Dokumente abfragen
link_get_children	untergeordnete Dokumente abfragen
link_documents	zwei Dokumente miteinander verknüpfen
link_delete	Verknüpfung löschen
link_exists	Testen, ob zwei Dokumente verknüpft sind

5.1.4 Verzeichnisse (Übersicht)

Verzeichnisfunktionen	Beschreibung
directory_create	Verzeichnis anlegen

Verzeichnisfunktionen	Beschreibung
directory_delete	Verzeichnis löschen
directory_exist	ermittelt, ob ein Verzeichnis existiert
directory_list	Inhalt eines Verzeichnisses auflisten (maximal 100 Einträge)
directory_count_files	Anzahl Dateien in einem Verzeichnis nach Suchmuster bestimmen; auch Unterordner können eingebunden werden

5.1.5 Status (Übersicht)

Statusfunktionen	Beschreibung
document_block	Dokument sperren
document_verify	Dokument prüfen
document_transfer	Statustransfer durchführen

5.1.6 Notizen (Übersicht)

Notizfunktionen	Beschreibung
note_add_file	Dateiinhalt an Notizdatei anhängen
note_add_string	Text an Notizdatei anhängen

5.1.7 Wiedervorlage (Übersicht)

Wiedervorlagefunktionen	Beschreibung
hold_file_send	Dokument an Wiedervorlage senden
hold_file_find	Dokument in der Wiedervorlage finden
hold_file_delete	Dokument aus Wiedervorlage löschen
hold_file_get_document_info	Informationen zu einer Wiedervorlage

5.1.8 Workflow (Übersicht)

Workflowfunktionen	Beschreibung
workpath_get_document_info	Infos zu aktuellem Leitweg abfragen
workpath_end_document	Dokument aus Leitweg entfernen
workpath_start_document	Dokument in Leitweg stellen
workpath_go_to_next_step	Dokument im Workflow weiterschieben
document_render_wfl_prot	

Workflowfunktionen	Beschreibung
	vertifft die Workflow-Protokolle zu einer Dokument-ID und erzeugt abhängige w1-Datei

5.1.9 Dokumentart (Übersicht)

Dokumentartfunktionen	Beschreibung
document_type_get_long	Dokumentart-Langtext bestimmen
document_type_get_short	Dokumentart-Kürzel bestimmen

5.1.10 Wertemengen (Übersicht)

Wertemengenfunktionen	Beschreibung
predefined_values_get	Wertemenge abfragen
predefined_values_test	Werte gegen Wertemenge prüfen
d3set_add_filter	fügt einen Filter zu einer bestehenden Restriktionsmenge hinzu

5.1.11 Benutzer (Übersicht)

Benutzerfunktionen	Beschreibung
user_get	alle Benutzernamen
user_get_groups	Gruppen, in denen der Benutzer Mitglied.
user_get_verifier_groups	Gruppen, in denen der Benutzer Prüfer.
user_get_rolls	Rollen, die dem Benutzer zugeordnet sind.
user_get_data	Daten in die globalen Variablen laden.
user_get_doc_types	Dokumentarten, auf die der Benutzer Zugriff hat.
user_validate_password	testet, ob ein Benutzer mit diesem Passwort in d.3 bekannt ist (nicht LDAP)
user_get_favourites	liefert die Dokument-IDs der Favoriten des Benutzers im Array api_doc_ids zurück
user_get_sid_rights	ermittelt zu einem Benutzer die benutzerspezifischen SID-Rechte/Eigenschaften
user_checkinout	Funktion zur An- bzw. Abwesenheitsmeldung von d.3 Benutzern
user_get_superior	Funktion zur Ermittlung der Vorgesetzten eines Benutzers

5.1.12 Gruppen (Übersicht)

Gruppenfunktionen	Beschreibung
group_get	Gruppennamen abfragen
group_get_members	Mitglieder einer Gruppe abfragen
group_get_verifiers	Prüfer einer Gruppe abfragen
group_get_doc_types	Dokumentarten einer Gruppe abfragen

5.1.13 Rollen (Übersicht)

Rollenfunktionen	Beschreibung
roll_get	alle Rollen-Ids abfragen
roll_get_names	alle Rollen-Namen abfragen
roll_get_users	alle Rollen-Mitglieder abfragen
roll_get_data	Daten der Rolle (ID, Name, Klassen, Rechte)

5.1.14 Restriktionsmengen (Übersicht)

d3set_remove_filter	entfernt einen Filter von einer bestehenden Restriktionsmenge
d3set_get_filter	ermittelt den Filter einer bestehenden Restriktionsmenge

5.1.15 TIFF-Funktionen (Übersicht)

TIFF-Funktionen	Beschreibung
document_render	Tiff oder PDF Datei erzeugen
tiff_create	Tiff-Datei erzeugen
tiff_concat	Tiff-Datei anhängen
document_render_wfl_prot	vertifft die Workflow-Protokolle zu einer Dokument-ID und erzeugt abhängige w1-Datei

5.1.16 Dateizugriffe (Übersicht)

Dateizugriffsfunktionen	Beschreibung
file_add_string	String an Datei anhängen
file_add_line	Zeile an Datei anhängen

Dateizugriffsfunktionen	Beschreibung
file_read_line	Zeile aus einer Datei lesen
file_concat	Dateien aneinander hängen
file_copy	Datei kopieren
file_delete	Datei löschen
file_compare	Dateien vergleichen
file_rename	Datei umbenennen
file_exist	existiert die Datei

5.1.17 Stringkonvertierung (Übersicht)

Stringkonvertierungsfunktionen	Beschreibung
string_to_lower	in Kleinbuchstaben wandeln
string_to_upper	in Großbuchstaben wandeln
string_to_number	numerisch aufbereiten
string_to_date	Datum aufbereiten
string_split	String an einem Separator aufsplitten
string_cut_leading_char	führende Zeichen entfernen
string_fill_leading_char	führende Zeichen einfügen
string_fill_trailing_char	anhängende Zeichen einfügen
string_replace	ersetzt Zeichen in einem String
string_get_position	sucht die Postion eines Strings im anderen
string_mask_special_char	maskiert die JAM-Sonderzeichen eines Strings
api_str_sub	liefert einen Teilstring der übergebenen Zeichenfolge
api_str_sub_utf8	liefert einen Teilstring der übergebenen Zeichenfolge
api_str_sub_ansi	liefert einen Teilstring der übergebenen Zeichenfolge
api_str_len	liefert die Anzahl an Zeichen in einem gegebenem String zurück
api_str_len_utf8	liefert die Anzahl an Zeichen in einem gegebenem String zurück
api_str_len_ansi	liefert die Anzahl an Zeichen in einem gegebenem String zurück
api_str_to_utf8	konvertiert eine Zeichenfolge in das UTF8-Format
api_str_from_utf8	konvertiert eine UTF8-Zeichenfolge nach ANSI
api_str_is_utf8	gibt an, ob eine Zeichenfolge im UTF8-Format gültig ist

Stringkonvertierungsfunktionen	Beschreibung
api_str_is_ascii	gibt an, ob eine Zeichenfolge im ASCII-Format gültig ist
api_str_cut_fract_utf8	entfernt ein durchtrenntes Multi-Byte-Zeichen vom Ende einer UTF8-kodierten Zeichenfolge
api_str_hash	erstellt Hash-Werten auf Strings

5.1.18 Diverses (Übersicht)

Diverse Funktionen	Beschreibung
restore_from_jukebox	von Jukebox wiedereinlagern
restore_from_history	gelöschtes Dokument wiederherstellen
send_email	eMail versenden
lock_token_acquire	legt eine Sperre für ein Objekt in der DB ab oder verlängert sie
lock_token_release	gibt eine Sperre für ein Objekt in der DB frei
start_ext_program	führt einen Kommandozeilenufruf durch (Start eines externen Programms)
get_current_datetime	liefert den aktuellen Zeitstempel im gewünschten Format zurück
regular_expression_test	testet einen Wert gegen einen regulären Ausdruck
object_property_set	setzt Eigenschaftswerte für Objekte. Durch das Setzen eines Leerstrings
object_properties_get	ermittelt die Eigenschaftswerte zu einer Eigenschaft oder zu einem Objekt.
set_signature_data	schreibt zu einem Dokument die Signaturinformationen in die Datenbank.

5.1.19 D3fc-Calls (Übersicht)

D3fc-Call-Funktionen	Beschreibung
d3fc_user_set	mit dieser Funktion setzen Sie einen ausführenden Benutzer für den d3fc-Call
d3fc_password_set	Passwort des ausführenden Benutzers für den d3fc-Call setzen
d3fc_remote_server_set	Name / IP des Hosts, der angesprochen werden soll
d3fc_port_set	Port des Hosts, der angesprochen werden soll
d3fc_server_set	Serverkürzel bei d.3 Prozessen
d3fc_timeout_set	Zeit in Sekunden, die auf eine Antwort des Remoteservers gewartet werden soll
d3fc_function_name_set	Name des Api-Calls

D3fc-Call-Funktionen	Beschreibung
d3fc_importing_set	ImportParameter setzen
3fc_exporting_set	ExportParameter setzen
d3fc_exporting_get	ExportParameter abfragen nach d3fc_execute
d3fc_table_set_import_file	Import-Datei setzen
d3fc_table_set_export_file	Export-Datei setzen
d3fc_table_set_headline	Tabellenkopf setzen
d3fc_table_add	Tabellenzeile zur Import-Tabelle hinzufügen
d3fc_field_set	Feld in der aktuellen Tabellenzeile der Importtabelle setzen
d3fc_field_get	Feld in der aktuellen Tabellenzeile der Export-Tabelle lesen
d3fc_first	erste Tabellenzeile der Export-Tabelle laden
d3fc_next	nächste Tabellenzeile der Export-Tabelle laden
d3fc_execute	einen API-Call ausführen

5.1.20 JAVA (Übersicht)

JAVA Funktionen	Beschreibung
java_add_param	Hinzufügen eines Parameters
java_clear_param	Löschen eines Parameters
java_call_static	Aufruf einer statischen Methode

5.2 Dokument

5.2.1 attribute_get_single

Funktion zum Abfragen eines einzelnen Attributes. Der Wert des Attributes wird in das globale Feld `api_single_info` geschrieben.

Parameter	Beschreibung
db_pos	Position des firmenspezifischen Feldes, das abgefragt wird oder Name des allgemeinen Attributes: <code>doc_date_import</code> : Importdatum <code>doc_last_alter_no</code> : Nummer der letzten Änderung <code>doc_publ_alter_no</code> : Nummer der Version in Freigabe <code>doc_status</code> : Dokumentstatus <code>doc_status_info</code> : Zusatzinfo zum Status (0/1), ungeprüft / geprüft für Prüfung, gesperrt / freigegeben für Freigabe (ab 5.5.1 Hotfix 14 bzw. 6.0) <code>doc_editor</code> : Bearbeiter

Parameter	Beschreibung
	<p>doc_type_short: Kürzel der Dokumentart doc_no: Dokumentnummer (zeich_nn) doc_var_no: Variantennummer doc_text_1: Bemerkungstext doc_owner: Besitzer doc_file_name: Dateiname doc_file_ext: Dateierweiterung doc_encrypt: Verschlüsselt ? (0/1) doc_cnt_access: Anzahl der Zugriffe doc_date_last_access: Datum des letzten Zugriffs doc_user_last_access: Benutzer des letzten Zugriffs doc_last_update_file: letzte Dateiänderung doc_web_published: webveröffentlicht ? (0/1) doc_workflow: im Leitweg ? (0/1) doc_type_type: a für Akte und d für Dokument doc_color_code: Farocode des Dokuments (0: nicht gesetzt; 1-24: Farocode).</p>
	<div style="border: 1px solid #0070C0; padding: 10px; background-color: #F0F8FF;">  neues Attribut doc_color_code ab Version 7.2. </div>
	Pflichtfeld
row_no	Zeilennummer bei Mehrfachattributfeldern optional; Defaultwert ist 0
doc_id	Dokument-ID, für die das Attribut abgefragt wird; wird nichts angegeben, wird die ID aus dem Kontext verwendet Pflichtfeld
user_name	Name des Benutzers, der das Attribut abfragt Optional

Rückgabewert	Bedeutung
0	alles OK
> 0	Datenbankfehler
-1	Panther Syntax Fehler, Details siehe LogViewer
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler, d.velop Support kontaktieren
-101	weder Dokument ID übergeben, noch im Kontext gesetzt
-103	DB-Position nicht vorgegeben

Rückgabewert	Bedeutung
-201	ungültigen Benutzernamen vorgegeben
-202	ungültige DB-Position vorgegeben
-203	Dokument <\:doc_id> der Dokumentart <\:doc_type_short> enthält nicht das Feld <\:db-pos> api_error_var[...] = "doc_type_short" api_error-value[...] = Ermitteltes Dokumentartkürzel zur DokumentID
-204	ungültige Zeilennummer vorgegeben
-301	ungültige Dokument ID vorgegeben oder Benutzer hat keine Rechte darauf
-302	Fehler beim Ermitteln der Benutzerberechtigungen
-303	Benutzer hat kein Zugriffsrecht auf das Dokument



Die Funktion ist ab d.3 Version 5.5.1 verfügbar.

Im

- Beispiel (A) wird abgefragt, welcher Wert auf der Datenbankposition 2 gespeichert worden ist
- Beispiel (B) wird ein Mehrfachattribut abgefragt
- Beispiel (C) wird abgefragt, welches Importdatum das Dokument hat



```
// -----
// Beispiel A:
// Abfragen eines einfachen Attributes
// -----
// Variablendeclaration
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!" )  
}
```



```
// -----  
// Beispiel B:  
// hier wird abgefragt welchen Wert das Mehrfach-Attribut  
// in einer Zeile hat  
// -----  
  
// Variablen-deklaration  
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!")  
}
```



```
// -----  
// Beispiel C:  
// hier wird abgefragt welches Importdatum das Dokument hat  
// -----  
  
// Variablen-deklaration  
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!")
    }
}

```

5.2.2 attribute_get_all

Funktion, die alle Attribute abfragt. Die Werte werden in die globalen Variablen des Dokumentkontextes geschrieben:

Globale Variable	Beschreibung
api_doc_field[1..89]	Attributfelder 1-89 (für 60er-Felder ist hier der erste Wert enthalten)
api_doc_field_60..69[1..2000]	60er Attribute mit allen Zeilen
api_doc_date_import	Importdatum
api_doc_last.Alter_no	Nummer der letzten Änderung
api_doc_publ.Alter_no	Nummer der Version in der Freigabe
api_doc_status	Dokumentstatus
api_doc_status_info	Zusatzinfo zum Status: (0/1) ungeprüft / geprüft für Prüfung, gesperrt / freigegeben für Freigabe (ab 5.5.1 Hotfix 14 bzw. 6.0)
api_doc_editor	Bearbeiter
api_doc_type_short	Kürzel der Dokumentart
api_doc_no	Dokumentnummer (zeich_nr)
api_doc_var_no	Variantennummer
api_doc_texts[1..4]	Bemerkungstext in einzelnen Zeilen
api_doc_owner	Besitzer
api_doc_file_name	Dateiname

Globale Variable	Beschreibung
api_doc_file_ext	Dateierweiterung
api_doc_encrypt	Verschlüsselt ? (0/1)
api_doc_cnt_access	Anzahl Zugriffe
api_doc_date_last_access	Datum des letzten Zugriffs
api_doc_user_last_access	Benutzer des letzten Zugriffs
api_doc_last_update_file	letzte Dateiänderung
api_doc_web_published	webveröffentlicht ? (0/1)
api_doc_workflow	im Leitweg ? (0/1)
api_color_code	Farocode des Dokuments (0: nicht gesetzt; 1-24: Farocode)
	 neue Variable ab Version 7.2

Parameter	Beschreibung
doc_id	Dokument-ID, für die das Attribut abgefragt wird; wird nichts angegeben, wird die ID aus dem Kontext verwendet. (Pflichtfeld: entweder als Parameter oder im Kontext: api_doc_id)
user_name	Name des Benutzers, der das Attribut abfragt Optional

Rückgabewert	Bedeutung
0	alles OK
> 0	Datenbankfehler
-1	Panther Syntax Fehler, Details siehe LogViewer
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler, d.velop Support kontaktieren
-101	weder Dokument ID übergeben noch im Kontext gesetzt
-201	ungültigen Benutzernamen vorgegeben
-301	ungültige Dokument ID vorgegeben oder Benutzer hat keine Rechte darauf
-302	Fehler beim Ermitteln der Benutzerberechtigungen

Rückgabewert	Bedeutung
-303	Benutzer hat kein Zugriffsrecht auf das Dokument



verfügbar ab d.3 Version 5.5.1

In diesem Beispiel werden alle Attributwerte des Dokumentes mit der ID „D000000153“ in die entsprechenden globalen Variablen geschrieben und teilweise ausgegeben. Tritt ein Fehler auf, wird dieser ausgegeben.



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

// Variablen-deklaration
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!")
}
```

5.2.3 attribute_update_single

Ändern eines einzelnen Attributes. Bei mehr als drei bis vier Aufrufen dieser Funktion hintereinander sollte attribute_update_all verwendet werden, um nicht bei jedem einzelnen Aufruf eine Validierung und Berechtigungskontrolle durchzuführen.

Parameter	Beschreibung
db_pos	<p>Position des firmenspezifischen Feldes, das abgefragt wird oder Name des allgemeinen Attributes:</p> <p>doc_text_1: Bemerkungstext (Pflichtfeld) doc_color_code: Farocode des Dokuments (0: nicht gesetzt; 1-24: Farocode).</p> <div style="border: 1px solid blue; padding: 10px; margin-top: 10px;">  neues Attribut doc_color_code ab Version 7.2 </div>
	Pflichtfeld
new_value	Wert, den das Attribut erhalten soll wird bei row_no = -1 ignoriert, ansonsten Pflichtfeld
row_no	<p>Zeilennummer bei Mehrfachattributfeldern 0 = als nächste Zeile anhängen (default) -1 = bei "normalen" Attributfeldern: Wert aus dem entsprechenden api_doc_field verwenden und new_value ignorieren; bei Mehrfachattributfeldern: alle Zeilen des Mehrfachattributfeldes löschen</p> <p>Optional</p>
doc_id	Dokument-ID, für die das Attribut abgefragt wird; wird nichts angegeben, wird die ID aus dem Kontext verwendet Pflichtfeld – entweder als Parameter oder im Kontext: api_doc_id
user_name	Name des Benutzers, der das Attribut abfragt optional

Rückgabewert	Bedeutung
0	alles OK
> 0	Datenbankfehler
-1	Panther Syntax Fehler, Details siehe LogViewer
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler, bitte d.velop Support kontaktieren
-101	weder Dokument ID übergeben noch im Kontext gesetzt
-103	DB-Position nicht vorgegeben
-117	kein neuer Wert für das Pflichtfeld <\:db_pos> in der Dokumentart <\:doc_type_short> angegeben:

Rückgabewert	Bedeutung
	<pre> api_error_var[...] = "doc_type_short" api_error_value[...] = Kürzel der Dokumentart api_error_var[...] = "modifiable" api_error_value[...] = Kürzel der Dokumentart api_error_var[...] = "modifiable" api_error_value[...] = Feld ist modifizierbar (0:nein; 1: ja) api_error_var[...] = "doc_flags" api_error_value[...] = Feldeigenschaften api_error_var[...] = "data_type" api_error_value[...] = Datentyp des Attributes </pre>
-201	ungültigen Benutzernamen vorgegeben
-202	ungültige DB-Position vorgegeben
-203	Dokument <\:doc_id>s Dokumentart <\:doc_type_short> enthält nicht das Feld <\:db_pos> <pre> api_error_var[...] = "doc_type_short" api_error_value[...] = Ermitteltes Dokumentartkürzel zur DokumentID </pre>
-204	ungültige Zeilennummer vorgegeben
-206	ungültigen Datentyp vorgegeben <pre> api_error_var[...] = "doc_type_short" api_error_value[...] = Kürzel der Dokumentart api_error_var[...] = "modifiable" api_error_value[...] = Feld ist modifizierbar (0:nein; 1: ja) api_error_var[...] = "doc_flags" api_error_value[...] = Feldeigenschaften api_error_var[...] = "data_type" api_error_value[...] = Datentyp des Attributes </pre>
-215	ungültigen Wert gegenüber Wertevorrat für das Feld <\:db_pos> der Dokumentart <\:doc_type_short> angegeben <pre> api_error_var[...] = "doc_type_short" api_error_value[...] = Kürzel der Dokumentart api_error_var[...] = "modifiable" api_error_value[...] = Feld ist modifizierbar (0:nein; 1: ja) api_error_var[...] = "doc_flags" api_error_value[...] = Feldeigenschaften api_error_var[...] = "data_type" api_error_value[...] = Datentyp des Attributes </pre>
-216	ungültigen Wert gegenüber Plausibilitätskriterien für das Feld <\:db_pos> der Dokumentart <\:doc_type_short> angegeben

Rückgabewert	Bedeutung
	<pre>api_error_var[...] = "doc_type_short" api_error_value[...] = Kürzel der Dokumentart api_error_var[...] = "modifiable" api_error_value[...] = Feld ist modifizierbar (0:nein; 1: ja) api_error_var[...] = "doc_flags " api_error_value[...] = Feldeigenschaften api_error_var[...] = "data_type " api_error_value[...] = Datentyp des Attributes</pre>
-301	ungültige Dokument ID vorgegeben oder Benutzer hat keine Rechte darauf
-302	Fehler beim Ermitteln der Benutzerberechtigungen
-303	Benutzer hat kein Zugriffsrecht auf das Dokument
-304	Benutzer hat kein Recht, die Attribute des Dokumentes zu ändern <pre>api_error_var[...] = "doc_permission" api_error_value[...] = Berechtigungen auf das Dokument</pre>
-501	Fehler beim Bestimmen der Dokumentart des Dokumentes <\:doc_id>
-515	Feld <\:db_pos> darf nicht geändert werden in Dokumentart <\:doc_type_short>" <pre>api_error_var[...] = "doc_type_short" api_error_value[...] = Kürzel der Dokumentart api_error_var[...] = "modifiable" api_error_value[...] = Feld ist modifizierbar (0:nein; 1: ja) api_error_var[...] = "doc_flags " api_error_value[...] = Feldeigenschaften api_error_var[...] = "data_type " api_error_value[...] = Datentyp des Attributes</pre>
-516	Fehler beim Ändern des Attributes; nur @dmrowcount Zeilen in der DB geändert <pre>api_error_var[...] = "doc_type_short" api_error_value[...] = Kürzel der Dokumentart api_error_var[...] = "modifiable" api_error_value[...] = Feld ist modifizierbar (0:nein; 1: ja) api_error_var[...] = "doc_flags " api_error_value[...] = Feldeigenschaften api_error_var[...] = "data_type " api_error_value[...] = Datentyp des Attributes</pre>
< -9500	Fehler in kundenspezifischer Hook-Funktion hook_upd_attrib <pre>api_error_var[...] = "hook_return_code" api_error_value[...] = Rückgabewert des Hooks</pre>



verfügbar ab d.3 Version 5.5.1.



Im Beispiel wird das Attribut mit der Datenbankposition 2 des Dokumentes mit der ID „Y000050257“ NICHT auf den Wert „317“ aktualisiert. Da der Parameter `h_AttributeValue` den Wert -1 hat, wird der Wert „299“ aus `api_doc_field[h_DB_PosRef]` zugewiesen. Tritt ein Fehler auf, wird dieser ausgegeben.



```
// -----
// Beispiel:
// Aktualisierung eines Attributes
// -----
// Variablendeclaration
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!")
}
```

Dieses Beispiel hängt nacheinander die Werte angehängt1, angehängt2 und angehängt3 an das Mehrfach-Attributfeld 60 an. Um die Verständlichkeit dieses Beispiels zu erhöhen, erfolgt eine Fehlerbehandlung nur für den letzten Aufruf, obwohl sie für jeden Aufruf erfolgen sollte.

Bei mehr als drei bis vier Aufrufen dieser Funktion hintereinander sollte attribute_update_all verwendet werden, um nicht bei jedem einzelnen Aufruf eine Validierung und Berechtigungskontrolle durchzuführen.



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



// Variablendeclaration
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!")
}
```



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

// Variablen-deklaration
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!")
}

```

5.2.4 attribute_update_all

Funktion zur Änderung aller Attribute im Kontext (Bemerkungsfeld = `api_doc_texts[1..4]`; Attribute = `api_doc_field[...]` und `api_doc_field_6x[...]`)

Parameter	Beschreibung
doc_id	Dokument-ID, für die die Attribute geändert werden; wird nichts angegeben, wird die ID aus dem Kontext verwendet Pflichtfeld
user_name	Name des Benutzers, der das Attribut ändert Optional

Rückgabewert	Bedeutung
0	alles OK
> 0	Datenbankfehler
-1	Panther Syntax Fehler, Details siehe LogViewer
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler, d.velop Support kontaktieren
-101	weder Dokument ID übergeben, noch im Kontext gesetzt
-117	kein neuer Wert für das Pflichtfeld <\:db_pos> in der Dokumentart <\:doc_type_short> angegeben api_error_var[...] = "doc_type_short" api_error_value[...] = Kürzel der Dokumentart api_error_var[...] = "db-position" api_error_value[...] = <Position> - fehlender Wert für Pflichtfeld
-201	ungültigen Benutzernamen vorgegeben
-215	ungültigen Wert gegenüber Wertevorrat für das Feld <\:db_pos> der Dokumentart <\:doc_type_short> angegeben api_error_var[...] = "doc_type_short" api_error_value[...] = Kürzel der Dokumentart api_error_var[...] = "db-position" api_error_value[...] = <Position> - ungültiger Wert für Wertevorrat
-216	ungültigen Wert gegenüber Plausibilitätskriterien für das Feld <\:db_pos> der Dokumentart <\:doc_type_short> angegeben api_error_var[...] = "doc_type_short" api_error_value[...] = Kürzel der Dokumentart api_error_var[...] = "db-position" api_error_value[...] = <Position> - ungültiger Wert laut Plausibilitätskontrolle
-301	ungültige Dokument ID vorgegeben oder Benutzer hat keine Rechte darauf

Rückgabewert	Bedeutung
-302	Fehler beim Ermitteln der Benutzerberechtigungen
-303	Benutzer hat kein Zugriffsrecht auf das Dokument
-304	Benutzer hat kein Recht, Attribute des Dokumentes zu ändern api_error_var[...] = "doc_permission" api_error_value[...] = Berechtigungen auf das Dokument
-501	Fehler beim Bestimmen der Dokumentart des Dokumentes <\ :doc_id>
-516	Fehler beim Ändern des Attributes; nur @dmrowcount Zeilen in der DB geändert api_error_var[...] = "doc_type_short" api_error_value[...] = Kürzel der Dokumentart
<- 9500	Fehler in kundenspezifischer Hook-Funktion hook_upd_attrib ... api_error_var[...] = "hook_return_code" api_error_value[...] = Rückgabewert des Hooks



verfügbar ab d.3 Version 5.5.1 Hotfix 1

Das erste Beispiel ändert die Attributfelder 1 und 60 und überschreibt sie mit neuen Werten. Alle übrigen Attributwerte gehen jedoch verloren, da die entsprechenden globalen Variablen nach Leerung des Dokumenten-Kontextes leer sind. Die Attributfelder werden daher entsprechend mit einem leeren Attributwert überschrieben.



```

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


// Variablendeclaration
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!")  
}
```

Im zweiten Beispiel werden nach der Leerung des Dokumenten-Kontextes zunächst die Dokumentattribute in die globalen Variablen gelesen. Erst anschließend wird die Funktion aufgerufen. So bleiben die ursprünglichen Attributwerte erhalten.



```
// -----  
// Beispiel B:  
// Aktualisierung der Attributfelder 1 und 60 OHNE  
// gleichzeitigem Verlust anderer Attributwerte  
// -----  
  
// Variablen-deklaration  
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! " )
}

```

5.2.5 attribute_get_valid_values

Funktion zur Abfrage erlaubter Kenndaten. Das globale Array d3server_field_no_allowed[] erhält die DB-Position des jeweiligen Feldes, die Werte werden je nach Datentyp in eines der Arrays d3server_value_char_allowed[], d3server_value_date_allowed[], d3server_value_num_allowed[] geschrieben. Die Anzahl der gelieferten Werte wird in api_single_info angegeben.

Parameter	Beschreibung
doc_type_short	Kürzel der Dokumentart, zu der das Attribut gehört, dessen Werte abgefragt werden sollen Pflichtfeld
db_pos	Position des Feldes, dessen Werte abgefragt werden sollen, in der Dokumentart Pflichtfeld
user_name	Benutzer, in dessen Namen die Notiz geschrieben werden soll; wird nichts angegeben, wird der ausführende Benutzer verwendet optional
row_no	Zeilennummer bei 60er-Feld für den Fall, dass die Werte über Hookfunktionen ermittelt werden optional - wird an Wertemengen-Hook übergeben

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe LogViewer
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler, bitte d.velop Support kontaktieren
-104	Dokumentartkürzel nicht vorgegeben
-201	ungültigen Benutzernamen vorgegeben
-303	Benutzer hat kein Zugriffsrecht
-517	Fehler beim Laden der Werte für das Attribut
-518	Fehler beim Ermitteln des Index der Dokumentart
-519	für Attribut bzw. alle Attribute ist keine Werteauswahl möglich



verfügbar ab d.3 Version 5.5.1 Hotfix 1

Dieses Beispiel gibt alle gültigen Werte der Wertemenge an Position 42 der Dokumentenart mit dem Dokumentenkürzel „DACAD“ aus. Tritt ein Fehler auf, wird dieser ausgegeben.



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


// Variablendeclaration
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!")
}
```

5.2.6 attribute_test_value

Funktion prüft einen Wert auf Gültigkeit und liefert 0 zurück, wenn der Wert gültig ist.

Parameter	Beschreibung
doc_type_short	Kürzel der Dokumentart, zu der das Attribut gehört, dessen Werte abgefragt werden sollen. (Pflichtfeld)
db_pos	Position des Feldes, dessen Werte abgefragt werden sollen, in der Dokumentart (Pflichtfeld)
new_value	Wert, dessen Gültigkeit geprüft werden soll (Pflichtfeld)
user_name	Benutzer, in dessen Namen die Notiz geschrieben werden soll; wird nichts angegeben, wird der ausführende Benutzer verwendet (optional)
row_no	Zeilennummer bei 60er-Feld für den Fall, daß die Werte über Hookfunktion ermittelt werden (optional – notwendig für 60er-Felder)

Rückgabewert	Bedeutung
0	alles OK
> 0	Datenbankfehler
-1	Panther Syntax Fehler, Details siehe LogViewer
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler, bitte d.velop Support kontaktieren
-103	DB-Position nicht vorgegeben
-104	Dokumentartkürzel nicht vorgegeben
-117	ein neuer Wert für das Pflichtfeld <\:db_pos> in der Dokumentart <\:doc_type_short> angegeben api_error_var[...] = "doc_flags" api_error_value[...] = Feldeigenschaften api_error_var[...] = "data_type" api_error_value[...] = Datentyp des Attributes
-201	ungültigen Benutzernamen vorgegeben
-203	Dokumentart <\:doc_type_short> enthält nicht das Feld <\:db_pos> .
-215	ungültigen Wert gegenüber Wertevorrat für das Feld <\:db_pos> der Dokumentart <\:doc_type_short> angegeben

Rückgabewert	Bedeutung
	<pre>api_error_var[...] = "doc_flags" api_error_value[...] = Feldeigenschaften api_error_var[...] = "data_type" api_error_value[...] = Datentyp des Attributes api_error_var[...] = "new_value" api_error_value[...] = ggf. aufbereiteter Wert</pre>
-216	ungültigen Wert gegenüber Plausibilitätskriterien für das Feld <\:db_pos> der Dokumentart <\:doc_type_short> angegeben <pre>api_error_var[...] = "doc_flags" api_error_value[...] = Feldeigenschaften api_error_var[...] = "data_type" api_error_value[...] = Datentyp des Attributes api_error_var[...] = "new_value" api_error_value[...] = ggf. aufbereiteter Wert</pre>
-303	Benutzer hat kein Zugriffsrecht auf das Dokument



verfügbar ab d.3 Version 5.5.1 Hotfix 1

In diesem Beispiel wird überprüft, ob der Testwert „Meier“ als Wert für das Attribut mit der Datenbankposition 5 der Dokumentenart „DER“ gültig ist. Tritt ein Fehler auf, wird dieser ausgegeben.



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


// Variablendeclaration
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!")
    }
}

```

5.2.7 attribute_update_begin



neu in d.3 Version 7.0

Mit `attribute_update_begin` werden die aktuellen Attribute des Dokuments in die `api_doc_fields` geschrieben.

Im Anschluss können die `api_doc_fields` verändert werden.

Die `dok_dat_felder` werden nicht gefüllt und Änderungen an den `dok_dat_feldern` werden auch nicht weiter beachtet beim Aufruf von `attribute_update_execute`. Mit dieser Funktion wird der übliche Vorgang der Aktualisierung durchgeführt. Dabei wird der Zustand der `api_doc_fields` auf Gültigkeit geprüft und dann in die Datenbank geschrieben. Für die Update-Hooks gilt dasselbe wie bei anderen Attributaktualisierungen.

Allerdings sind zwei Dinge zu beachten:

- die `dok_dat_felder` sind nicht mit den Werten des Dokuments belegt. Es muss also unbedingt auf die `api_doc_fields` zurückgegriffen werden. Um erkennen zu können, ob gerade diese Form der Aktualisierung aktiv ist, kann auf `api_attribute_update_active == 1` abgefragt werden.
- um ein gegenseitiges Überschreiben der `api_doc_fields` zu verhindern, ist ein rekursiver Aufruf dieser Funktion nicht möglich; es wird eine entsprechende Fehlermeldung im Log erscheinen.



Ist `SERVER_API_NO_HOOKS` aktiviert, werden die Update-Hooks nicht aufgerufen.

Parameter	Beschreibung
doc_id	Dokument-ID des Dokuments, dessen Attribute geändert werden sollen
user	Benutzer, der die Attributaktualisierung durchführt

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-101	keine doc_id angegeben
-304	Benutzer hat kein Recht die Attribute zu ändern
-640	zuviele rekursive Attributupdates
-641	unbekannte doc_id vorgegeben
-650	es wird bereits eine Attributaktualisierung über Server-API ausgeführt (Paar attribute_update_begin/_execute)

5.2.8 attribute_update_execute



neu in d.3 Version 7.0

Beschreibung: siehe attribute_update_begin

Parameter: keine

Voraussetzung zur Ausführung ist ein vorhergehendes erfolgreiches attribute_update_begin
(Handle ist intern bekannt, es werden doc_id und user von attribute_update_begin genommen und im d.3 logviewer auch ausgegeben)

Rückgabewert	Bedeutung
0	alles OK
>0	Datenbank-Fehler
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-203	Dokumentart enthält das Feld nicht (Näheres siehe d.3 logviewer)
-216	Plausibilitätsprüfung nicht bestanden (Näheres siehe d.3 logviewer)
-224	ein Attributwert zu lang (Näheres siehe d.3 logviewer)

Rückgabewert	Bedeutung
-651	es ist keine Attributaktualisierung aktiv, die hier ausgeführt werden könnte

5.2.9 document_change_type

Funktion zum Wechseln der Dokumentart

Parameter	Beschreibung
document_type_short	Kürzel der neuen Dokumentart Pflichtfeld
doc_id	ID des Dokumentes, dessen Dokumentart geändert werden soll Pflichtfeld
user_name	Name des ändernden Benutzers; ist nichts angegeben, wird der ausführende Benutzer verwendet optional

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler, d.velop Support kontaktieren
-101	weder Dokument ID übergeben noch im Kontext gesetzt
-104	Dokumentatkürzel nicht vorgegeben
-201	ungültigen Benutzernamen vorgegeben
-205	ungültiges Dokumentatkürzel vorgegeben
-301	ungültige Dokument ID vorgegeben oder Benutzer hat keine Rechte darauf
-302	Fehler beim Ermitteln der Benutzerberechtigungen
-303	Benutzer hat kein Zugriffsrecht auf das Dokument
-304	Benutzer hat kein Recht, die Attribute des Dokumentes zu ändern api_error_var[...] = "doc_permission" api_error_value[...] = Berechtigungen auf das Dokument
-505	Fehler beim Speichern der Attribute
< -9500	Fehler in kundenspezifischer Hook-Funktion hook_upd_attrib api_error_var[...] = "hook_return_code" api_error_value[...] = Rückgabewert des Hooks



verfügbar ab d.3 Version 5.5.1

Im folgenden Beispiel wird die Dokumentart des Dokumentes mit der ID „D000000513“ zu der Dokumentart mit dem Kürzel „DER“ geändert, soweit kein Fehler auftritt.



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

// Variablen-deklaration
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!")
}
```

5.2.10 document_delete

Funktion löscht ein Dokument aus dem d.3 System.

Parameter	Beschreibung
reason	Löschergrund, bei Freigabe/Archiv-Versionen nötig Pflichtfeld bei Status Freigabe/Archiv, anderenfalls optional
del_from_each_status	aus allen Status löschen 0=nur die aktuelle Version 1=alle Versionen

Parameter	Beschreibung
	optional, default: 0
del_file_always	<p>Nutzdatei unabhängig vom Status löschen 0=nur Nutzdateien in Bearbeitung/Prüfung löschen 1=Nutzdatei in jedem Fall löschen</p> <p>optional, default:0</p>
doc_id	ID des Dokuments, das gelöscht werden soll; wird nichts übergeben, wird die ID des Kontexts genutzt Pflichtfeld, falls nicht im Kontext (<code>api_doc_id</code>) angegeben
user_name	Benutzer, in dessen Namen das Dokument gelöscht werden soll; wird nichts angegeben, wird der ausführende Benutzer verwendet optional
del_privileged	ist dieser auf "1" gesetzt, wird die entsprechende Lizenz überprüft und das Recht des ausführenden Benutzers, privilegiert zu löschen, getestet; ist dies nicht gegeben, kommen die Fehler 319 oder 320
 neu in Version 7.0	

Rückgabewert	Bedeutung
0	alles OK
> 0	Datenbankfehler
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler, d.velop Support kontaktieren
-101	weder Dokument ID übergegeben noch im Kontext gesetzt
-120	Löschergrund nicht angegeben
-201	ungültigen Benutzernamen vorgegeben
-301	ungültige Dokument ID vorgegeben oder Benutzer hat keine Rechte darauf
-302	Fehler beim Ermitteln der Benutzerberechtigungen
-303	Benutzer hat kein Zugriffsrecht auf das Dokument
-311	

Rückgabewert	Bedeutung
	Benutzer hat kein Recht das Dokument zu löschen api_error_var[...] = "doc_permission" api_error_value[...] = Berechtigung
-319	keine Lizenz, um privilegiert zu löschen
-320	Benutzer hat kein Recht, privilegiert zu löschen
<-9500	Fehler in kundenspezifischer Hook-Funktion hook_delete api_error_var[...] = "hook_return_code" api_error_value[...] = Rückgabewert des Hooks



verfügbar ab d.3 Version 5.5.1 Hotfix 1

Dieses Beispiel löscht das Dokument mit der ID „Y000050256“ aus dem Archiv, falls es in Bearbeitung oder in Prüfung ist. Tritt ein Fehler auf, wird dieser ausgegeben.



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


// Deklaration von Variablen
vars h_ReturnValue          // Fehlercode
vars h_DocIdRef = "Y000050256" // zu lösches 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!")
    }
}

```

5.2.11 document_find_ids

Funktion sucht Dokumente zum Kontext. KEINE Volltextsuche; kein Aufruf der Such-Hooks, keine Wertemengen-Prüfung; keine Case-Konvertierung.

In `api_single_info` wird die Anzahl der gefundenen Dokument-IDs eingetragen. Diese werden im Array `api_doc_ids[]` hinterlegt.

Berücksichtigung folgender Kontext-Variablen:

Kontext-Variable	Beschreibung
<code>api_doc_field[1..89]</code>	Attributfelder 1-89
<code>api_doc_type_short</code>	Kürzel der Dokumentart
<code>api_doc_status</code>	aktueller Status des Dokuments
<code>api_doc_editor</code>	Bearbeiter des Dokuments



Die `api_error_var[...]` / `api_error_value[...]` enthalten die berücksichtigten Suchkriterien.

Parameter	Beschreibung
<code>user_name</code>	Benutzer, in dessen Namen gesucht werden soll; wird nichts angegeben, wird der ausführende Benutzer verwendet optional

Rückgabewert	Bedeutung
0	alles OK
> 0	Datenbankfehler
-1	Panther Syntax Fehler, Details siehe d.3 logviewer



verfügbar ab d.3 Version 5.5.1 Hotfix 1

Das Beispiel sucht nach Dokumenten-IDs anhand der globalen Variablen `api_doc_type_short` und `api_doc_field[2]`. Damit nur diese beiden Variablen für die Suche herangezogen werden, wurde zuvor der Dokumenten-Kontext geleert. Anschließend werden die Gesamtzahl und die IDs der gefundenen Dokumente ausgegeben. Tritt ein Fehler auf, wird dieser ausgegeben.



```
// Suche nach Dokumenten
// -----
// Variablen-deklaration
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!")
}
```



5.2.12 document_get_permission

Die Funktion gibt die Abfrageberechtigungen eines Benutzers auf ein Dokument an.

Die Angabe wird in `api_single_info` geschrieben in der Form, die beim API-Call

`SearchDocument (doc_permission)` beschrieben ist: jeweils 1:Ja, 0:Nein

a1a2a3a4a5a6a7a8-b1b2b3b4b5b6b7b8-c1c2-d1d2
mit:

Parameter	Bedeutung
a1	Zugriff auf Dokumentart erlaubt
a2	Notizdatei vorhanden
a3	Dokumentverknüpfungen vorhanden
a4	Nutzdatei vorhanden
a5	Nutzdatei verschlüsselt
a6	Version in Freigabe existiert (1: freigegeben, 2: gesperrt)
a7	neueste Version in Prüfung (1: geprüft, 2: ungeprüft)
a8	mindestens eine Version im Status Archiv
b1	Recht, die Nutzdatei zu visualisieren
b2	Recht, die Kenndaten zu ändern
b3	Recht, die Nutzdaten zu ändern
b4	Recht, das Dokument zu löschen
b5	Recht zum Statustransfer
b6	Recht, das Dokument zu prüfen
b7	Recht, das Dokument freizugeben / zu sperren
b8	Recht, beide Prüfschritte in einem Schritt durchzuführen (Prüfung + Freigabe)
c1	Recht, eine ältere Version in Freigabe zu visualisieren
c2	Recht, eine ältere Version in Freigabe freizugeben / zu sperren
d1	Recht, Versionen im Status Archiv zu sehen
d2	Dokument befindet sich im Leitweg

Parameter	Beschreibung
doc_id	

Parameter	Beschreibung
	ID des Dokuments, dessen Berechtigungen abgefragt werden; wird nichts übergeben, wird die ID des Kontextes verwendet Pflichtfeld – entweder als Parameter oder im Kontext als api_doc_id
user_name	Benutzer, dessen Berechtigungen abgefragt werden sollen; wird nichts angegeben, wird der ausführende Benutzer verwendet optional

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer



verfügbar ab d.3 Version 5.5.1 Hotfix 14, 6.0

Dieses Beispiel fragt die Berechtigungen für das Dokument „Y000050257“ ab und gibt diese codiert aus. Tritt ein Fehler auf, wird dieser ausgegeben.

```

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


// Variablenklärung
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!")
}
```

5.2.13 document_publish_for_web

Markiert ein Dokument als (nicht-) web-veröffentlicht.

Parameter	Beschreibung
publish	0: Dokument nicht web-veröffentlicht setzen 1: Dokument als webveröffentlicht setzen (default)
doc_id	Dokument das veröffentlicht werden soll
user_name	Benutzer, der die Veröffentlichung vornimmt

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer



verfügbar ab d.3 Version 6.2

Dieses Beispiel markiert das Dokument mit der ID "Y000005174" als "web veröffentlicht". Im Fehlerfall wird die Fehlernummer zurückgegeben.



```

// -----
// Beispiel 1:
// Dokument markieren als „web-veröffentlicht“
// -----
// Variablendeclaration
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!")
}

```

Dieses Beispiel zeigt, wie das Dokument mit der ID "Y000005174" als nicht web veröffentlicht gesetzt wird.

Im Fehlerfall wird die Fehlernummer zurückgegeben.



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

// Variablendeclaration
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!")
}
```

5.2.14 document_dependent_add



neu in Version 7.1:
die Funktion kann keine Signaturdatei (.s1) mehr zum Dokument ablegen; es wird dann der Fehler -670 zurückgegeben

Legt ein abhängiges Dokument ab bzw. überschreibt ein existierendes.

Parameter	Beschreibung
filename	Name der neuen abhängigen Datei
doc_status	Status (B, P, F) zu dem das abhängige Dokument abgelegt werden soll Default: neueste Version
doc_ext	Erweiterung der abhängigen Datei (default: Erweiterung von filename)
doc_id	Dokument zu dem das abhängige gehören soll
user_name	Name des ausführenden Benutzers

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-101	keine DokumentID vorgegeben
-150	filename nicht angegeben
-214	ungültigen Status angegeben (muss B, P oder F sein)
-250	Datei nicht gefunden
-313	Benutzer darf Nutzdaten nicht ändern
-350	Zugriff auf Datei verweigert
-555	Fehler beim Umbenennen der Datei
-556	Fehler beim Kopieren der Datei
-617	Fehler beim Abfragen des aktuellen Status aus der DB
-670	die Funktion kann keine Signaturdatei (.s1) mehr zum Dokument ablegen



verfügbar ab d.3 Version 6.2

Legt ein abhängiges Dokument ab bzw. überschreibt ein existierendes.



```

// -----
// Beispiel:
// Neues abhängiges Dokument anfügen
// -----
// Variablendeclaration
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!")
    }
```

5.2.15 document_dependent_delete

Löscht ein abhängiges Dokument.

Parameter	Beschreibung
doc_status	Status (B, P, F), in dem das zu löschen abhängige Dokument vorliegt default: neueste Version
doc_ext	Erweiterung der abhängigen Datei
doc_id	Dokument-ID des abhängigen Dokuments
user_name	Name des ausführenden Benutzers

Rückgabewert	Bedeutung
0	alles OK
>0	Datenbankfehler
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-101	keine DokumentID vorgegeben
-139	keine doc_ext angegeben
-214	ungültigen Status angegeben (muss B, P oder F sein)
-313	Benutzer darf Nutzdaten nicht ändern
-350	Zugriff auf Datei verweigert
-617	Fehler beim Abfragen des aktuellen Status aus der DB



verfügbar ab d.3 Version 6.2

Löscht ein abhängiges Dokument.



```
// -----
// Beispiel:
// Neues abhängiges Dokument löschen
// -----
// Variablendeclaration
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!")
}
```

5.2.16 document_register_dependent

Registriert die abhängigen Dateien eines Dokuments.

Parameter	Beschreibung
doc_id	Stammdokument für das die abhängigen Dokumente registriert werden sollen
archive_index	Index der Archiv-Version erforderlich für status = "Archiv"
status	Status des Dokuments: "Bearbeitung", "Prüfung", "Freigabe" oder "Archiv"

Parameter	Beschreibung
user_group	User-/Gruppenname erforderlich für status = Bearbeitung => B. /Gr. status = Prüfung => Gr.

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-101	weder Dokument ID übergeben noch im Kontext gesetzt
-112	für Status Bearbeitung keinen Benutzer- oder Gruppennamen vorgegeben
-113	für Status Prüfung keinen Gruppennamen vorgegeben
-115	für Status Archiv keinen Index der Archiv-Version vorgegeben
-126	weder Status übergeben noch im Kontext gesetzt



verfügbar ab d.3 Version 6.1.1

5.2.17 document_get_file_path

Über diese Funktion wird der komplette Pfad eines Dokuments im d.3-Dokumentenbaum ermittelt. Die Variable `api_single_info` enthält den gefundenen Pfad.

Parameter	Beschreibung
doc_id	Stammdokument für das der Pfad ermittelt werden soll.
status	Status des Dokuments: "Bearbeitung", "Prüfung", "Freigabe" oder "Archiv"
archive_index	Index der Archiv-Version erforderlich für <code>status = "Archiv"</code>
user_group	wird dieser Parameter übergeben, hat er Vorrang vor dem tatsächlichen Bearbeiter des unter <code>doc_id</code> angegebenen Dokuments; nur relevant für Status Bearbeitung

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-115	für Status Archiv keinen Index der Archiv-Version vorgegeben

Rückgabewert	Bedeutung
-126	weder Status übergeben noch im Kontext gesetzt



verfügbar ab d.3 Version 6.1.1

5.2.18 document_start_lifetime

Setzt die Lebensdauer eines Dokuments.

Parameter	Beschreibung
doc_id	Dokument ID des Dokuments
overwrite_old_date	Überschreiben des aktuellen Wertes (default: 1)
life_time_days	wenn nicht die für die Dokumentart konfigurierte Lebensdauer eingetragen werden soll, kann man hier die Anzahl Tage angeben (default 0 - konfigurierte Lebensdauer)

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-101	keine doc_id angegeben
-621	für die Dokumentart ist nicht vorgesehen die Lebenszeit zu ändern und overwrite_old_date mit 0 übergeben
-622	Datenbank-Fehler; siehe Logviewer für Einzelheiten

Setzt die Lebensdauer des Dokuments:



```
// -----
// Beispiel:
// Lebensdauer eines Dokumentes auf ca. 10 Jahre setzen
// -----


// Variablendeclaration
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 Tagen
```

```

// 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!")
}

```

5.2.19 document_set_cache_days

Setzt für eine Dokumentversion die Tage, die es "von heute an" im d.3 Dokumentbaum verbleiben soll. Diese Anzahl Tage ist letztlich nur relevant, wenn das Dokument auf Sekundärspeicher ausgelagert wird/worden ist.

Parameter	Beschreibung
doc_id	Dokument ID des Dokuments
status	Status des Dokuments: "Freigabe" oder "Archiv" (Default "Fr")
archive_index	Index der Archiv-Version erforderlich für status = "Archiv" Pflichtfeld bei Status "Archiv", wird bei "Freigabe" nicht beachtet
days_in_cache	Tage, die die Dokumentversion von heute an im Cache verbleiben soll Pflichtfeld

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-115	für Status Archiv keinen Index der Archiv-Version vorgegeben
-223	ungültigen Status angegeben (muss F oder A sein)
-623	

Rückgabewert	Bedeutung
	Anzahl Tage, die die Dokumentversion im d.3 Dokumentbaum verbleiben soll, nicht vorgegeben
-624	Dokument-Version existiert nicht oder ist bereits aus dem d.3 Dokumentbaum gelöscht



Diese Funktion ist ab d.3 Version 6.3 verfügbar.

5.2.20 document_send_to_dsearch



neu in d.3 Version 7.0

Sendet OCR-Informationen einer Dokumentversion an d.3 search.

Parameter	Beschreibung
doc_id (P)	Dokument, das in d.3 search aktualisiert werden soll
ocr_file	OCR-Datei, die die OCR-Informationen enthält; wenn nur die Attribute übermittelt werden sollen oder use_existent_ocr_file aktiviert wird, braucht dieser Parameter nicht gefüllt zu sein.
version	Version des Dokumentes, die verändert werden soll; wenn nicht vorgegeben, dann wird die aktuelle Version benutzt; für Notizdateien muss hier die Konstante <code>not</code> übergeben werden
dsearch_corpus	Corpus an den das Dokument übergeben werden soll; muss gesetzt werden, wenn ein vorhandenes Dokument an einen anderen/neuen Corpus übergeben werden soll
use_existent_ocr_file	Schalter um bereits vorhandene OCR-Datei nutzen zu können: kann gesetzt werden, um vorhandene Dokumente neu an d.3 search zu übergeben (Recovery, neue Volltext-Engine, ..) default: 0
doc_status	Status der Dokumentversion, die aktualisiert werden soll; wenn nicht gefüllt, wird die aktuellste Dokumentversion genutzt
archive_index	Archivindex einer Archivversion; muss nur mitgegeben werden, wenn doc_status "Archiv" ist
user_name	Name des ausführenden Benutzers

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-101	keine doc_id angegeben
-313	Benutzer darf Nutzdaten nicht ändern
-660	kein DSEARCH_SUPPORT.
-661	Fehler beim Anlegen des OCR-Unterverzeichnisses
-662	Fehler beim Kopieren der OCR-Datei
-663	d.3 search meldet Fehler, Fehlernummer bitte dem d.3 logviewer entnehmen

5.2.21 add_inherit_doc_rights



neu in d.3 Version 7.0

Funktion zur Vererbung von Rechten auf ein Dokument.

Parameter	Beschreibung
doc_id	Dokument, auf das die Rechte gesetzt werden soll
granter	Benutzer, der das Rechte vererbt
grantee	Benutzer, dem das Recht vererbt wird
right_flags	Position 1: 1=Lese-Rechte, 2=Schreib-Rechte (impliziert auch Lese-Rechte) Position 2: 1=Ablauf nach Zeitstempel, 2=Ablauf nach Quittierung Position 3: 1=rekursiv (wirkt sich auf alle Dokumente auf, die mit dem Dokument als Kind verknüpft sind)
tstamp_expire	Zeitstempel für Ablaufzeitpunkt (Format DD.MM.YYYY – hh:mm:ss); leer, wenn right_flags[2] = 2

Rückgabewert	Bedeutung
0	alles OK
3	fehlerhafte oder fehlende Parameter
120	Benutzer (granter bzw. grantee) existieren nicht

Rückgabewert	Bedeutung
450	granter darf keine Rechte vererben
451	granter hat nicht die Rechte, die vererbt werden sollen
452	grantee hat bereits die Rechte auf das Dokument
9000	Datenbank-Fehler



```

//-----
-----//
// NAME
// rechtevererben.jpl
//
// BESCHREIBUNG:
//
// Diese JPL-Datei dient als Beispiel für die Funktion
// add_inherit_doc_rights. In diesem Beispiel wird dem Benutzer
// grantee vom Benutzer Adminis1 das Schreibrecht auf das Dokument
// P000000792 vererbt.
//
//-----
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"

// Die Variablen h_ReadWriteFlag, h_DateFlag, h_Rekursive und
// h_InheritDate
// bilden das Recht. Es wird das Schreibrecht rekursiv auf dieses
// Dokument
// vererbt. Das Recht läuft am 01.01.9999 ab

```

```

h_ReadWriteFlag = 2
h_DateFlag = 1
h_Rekursive = 1
h_InheritDate = "01.01.9999"

h_ReturnValue = api_function( "add_inherit_doc_rights", wf_DokuID,
h_granter, \
h_grantee, h_ReadWriteFlag ## h_DateFlag \
## h_Rekursive, h_InheritDate ## " 00\:00\:00" )

if (h_ReturnValue == 0)
{
call api_log_info("Rechte wurden vererbt.")
}
else
{
call api_log_error("Rechte wurden nicht vererbt. Fehler
:h_ReturnValue")
}

SET_API_USER_MASTER = tmpMasterAPI

```

5.2.22 remove_inherit_doc_rights



neu in d.3 Version 7.0

Funktion zum Löschen von vererbten Rechten auf ein Dokument.

Parameter	Beschreibung
doc_id	Dokument, von dem die Rechte gelöscht werden sollen
user_name	Benutzer, unter dem das Recht gelöscht werden soll
grantee	Benutzer, dem das Recht entzogen werden soll

Rückgabewert	Bedeutung
0	alles OK
456	granter darf das Recht nicht löschen

5.3 Akte

5.3.1 Akte

5.3.2 folder_create

Funktion zum Anlegen einer Akte.

Attribute der neuen Akte müssen in den

`api_doc_field's`

`api_doc_texts[1..4]`

`api_doc_status`

`api_doc_editor`

`api_doc_nound api_doc_type_short`

stehen.

Bei synchronem Anlegen enthält bei Erfolg `api_doc_id` die Dokument ID der neuen Akte.

Parameter	Beschreibung
<code>synchron</code>	Akte direkt anlegen (0: asynchronous, 1: synchronous) optional – default: 0

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-104	Dokumentart nicht vorgegeben in <code>api_doc_type_short</code>
-548	Fehler beim Anlegen der Akte; der Logviewer enthält ggf. nähere Informationen
-566	Fehler beim Bestimmen des temporären Dateinamens
-567	Fehler beim Anlegen der JPL-Datei
-568	Fehler beim Anlegen der Nutzdatei



verfügbar ab d.3 Version 5.5.1 Hotfix 14, 6.0

In diesem Beispiel werden zunächst in globalen Variablen Informationen zum Anlegen einer Akte hinterlegt. Im Anschluss daran wird die Funktion zur Aktenanlage aufgerufen. Tritt dabei ein Fehler auf, wird dieser ausgegeben.



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

// Variablendeclaration
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"

// synchrone 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!")
}
```

5.4 Verknüpfungen

5.4.1 link_get_parents

Funktion zum Abfragen der übergeordneten Dokumente. Die IDs der übergeordneten Dokumente werden in das globale Array `api_links[]` geschrieben, der `link_creator` in `api_link_creators[]`, der Zeitstempel in `api_link_tstamps[]`. Die Variable `api_single_info` enthält die Anzahl der gefundenen Verknüpfungen.



`link_get_children` schreibt ebenfalls in `api_links[]`.

Parameter	Beschreibung
<code>doc_id</code>	ID des Dokuments, zu dem übergeordnete Dokumente abgefragt werden; wird nichts übergeben, wird die ID des Kontexts verwendet Pflichtfeld – entweder als Parameter oder im Kontext als <code>api_doc_id</code>
<code>user_name</code>	Benutzer, in dessen Namen die Verknüpfung durchgeführt werden soll; wird nichts angegeben, wird der ausführende Benutzer verwendet optional

Rückgabewert	Bedeutung
0	alles OK
> 0	Datenbankfehler
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler; d.velop Support kontaktieren
-101	weder Dokument ID übergeben, noch im Kontext gesetzt
-201	ungültigen Benutzernamen vorgegeben
-301	ungültige Dokument ID vorgegeben oder Benutzer hat keine Rechte darauf
-302	Fehler beim Ermitteln der Benutzerberechtigungen
-303	Benutzer hat kein Zugriffsrecht auf das Dokument



verfügbar ab d.3 Version 5.5.1

Dieses Beispiel fragt die übergeordneten Dokumente von „Y000050257“ ab und gibt diese aus. Tritt ein Fehler auf, wird dieser ausgegeben.



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

```

// Variablendeclaration
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!")
}

```

5.4.2 link_get_children

Funktion zum Abfragen der untergeordneten Dokumente. Die IDs der untergeordneten Dokumente werden in das globale Array `api_links[]` geschrieben, der `link_creator` in `api_link_creators[]`, der Zeitstempel in `api_link_tstamps[]`. Die Variable `api_single_info` enthält die Anzahl der gefundenen Verknüpfungen.



`link_get_parents` schreibt ebenfalls in `api_links[]`.

Parameter	Beschreibung
<code>doc_id</code>	ID des Dokuments, zu dem untergeordnete Dokumente abgefragt werden; wird nichts übergeben, wird die ID des Kontext verwendet Pflichtfeld – entweder als Parameter oder im Kontext als <code>api_doc_id</code>
<code>user_name</code>	Benutzer, in dessen Namen die Verknüpfung durchgeführt werden soll; wird nichts angegeben, wird der ausführende Benutzer verwendet optional

Rückgabewert	Bedeutung
0	alles OK
> 0	Datenbankfehler
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler; d.velop Support kontaktieren
-101	weder Dokument ID übergeben, noch im Kontext gesetzt
-201	ungültigen Benutzernamen vorgegeben
-301	ungültige Dokument ID vorgegeben oder Benutzer hat keine Rechte darauf
-302	Fehler beim Ermitteln der Benutzerberechtigungen
-303	Benutzer hat kein Zugriffsrecht auf das Dokument



verfügbar ab d.3 Version 5.5.1

Dieses Beispiel fragt die untergeordneten Dokumente von „Y000050501“ ab und gibt diese aus.

Tritt ein Fehler auf, wird dieses ausgegeben.



```
// Beispiel:
// Abfrage untergeordneter Dokumente
// -----
//
// Variablendeclaration
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!")
}

```

5.4.3 link_documents

Funktion verknüpft zwei Dokumente miteinander.

Falls ein Job angelegt wird, wird die Jobnummer in `api_single_info` zurückgegeben, wenn die `doc_id_parent` nicht vorgegeben wird.

Parameter	Beschreibung
<code>doc_id_parent</code>	ID des übergeordneten Dokumentes. Wird nichts angegeben, wird ein async Job zur Verknüpfung angelegt mit den Erkennungsattributen, die in den <code>api_doc_field's</code> eingetragen sind. Für die Pflege der <code>inherit_field_name_1..2 / inherit_field_value_1..2</code> und die Vererbung ist der Benutzer selbst verantwortlich. optional
<code>doc_id_child</code>	ID des untergeordneten Dokumentes. Wird nichts angegeben, wird die Dokument ID aus dem Kontext verwendet. Pflichtfeld – entweder als Parameter oder im Kontext als <code>api_doc_id</code>
<code>user_name</code>	Benutzer, in dessen Namen die Verknüpfung durchgeführt werden soll. Wird nichts angegeben, wird der ausführende Benutzer verwendet. optional
<code>folder_definition</code>	Akte anlegen, falls nicht vorhanden optional – default: 0
<code>use_folderplan</code>	Wenn <code>doc_id_parent</code> nicht angegeben ist 0: LIN002-JOB anlegen mit den Erkennungsattributen in den <code>api_doc_field's</code> und <code>api_doc_type_short</code> und den Vererbungsattributen in <code>ernie_inherit_field_name_1..4</code> 1: Aktenplan für das Dokument verwenden optional – default: 0

Rückgabewert	Bedeutung
0	alles OK
> 0	Datenbankfehler
-1	Panther Syntax Fehler, Details siehe LogViewer
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler; bitte d.velop Support kontaktieren
-101	weder Dokument ID übergeben noch im Kontext gesetzt
-201	ungültigen Benutzernamen vorgegeben
-219	Vater und Kind Dokument ID sind identisch
-301	ungültige Dokument ID vorgegeben oder Benutzer hat keine Rechte darauf
-302	Fehler beim Ermitteln der Benutzerberechtigungen
-303	Benutzer hat kein Zugriffsrecht auf das Dokument
-312	Benutzer hat kein Recht, Dokumente zu verknüpfen / lösen api_error_var[...] = "right_level" api_error_value[...] = "<Rechtestufe>" api_error_var[...] = "right_link" api_error_value[...] = "<Recht zum Aktenverknüpfen>"
-520	die Dokumente sind bereits auf diese Weise verknüpft
-521	die Dokumente sind bereits in umgekehrter Hierarchie verknüpft
<-9500	Fehler in kundenspezifischer Hook-Funktion hook_link api_error_var[...] = "hook_return_code" api_error_value[...] = Rückgabewert des Hooks



Verfügbar ab d.3 Version 5.5.1 Hotfix 14, 6.0

In dem ersten Beispiel wird eine Verlinkung der Datei „Y000050256“ durch Angabe der übergeordneten Datei „Y00050501“ erzeugt.

In dem zweiten Beispiel wird die Datei „Y000050256“ dagegen anhand des Aktenplanes vorgenommen.



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

```

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!")
}

```



```

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

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

// Angabe von Erkennungs-Attributten
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!")
}
```

5.4.4 link_exists

Testen, ob zwei Dokumente verknüpft sind.

Parameter	Beschreibung
doc_id_parent	ID des übergeordneten Dokuments
doc_id_child	ID des untergeordneten Dokuments
test_vice-versa	1: auch auf umgekehrte Verknüpfung prüfen (default=0)

Rückgabewert	Bedeutung
0	Dokumente sind nicht miteinander verknüpft
1	Dokumente sind miteinander verknüpft
2	Dokumente sind in der anderen Richtung miteinander verknüpft
>2	Datenbank-Fehler
-1	Panther Syntax Fehler, Details siehe LogViewer
-2	Funktion wurde falsch aufgerufen
-101	weder Dokument ID übergeben, noch im Kontext gesetzt für Vater bzw. Kind
-219	Vater und Kind Dokument ID sind identisch



verfügbar ab d.3 Version 6.2

Testen, ob zwei Dokumente verknüpft sind:

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

// Variablenklärung
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!")
}

```

5.4.5 link_delete

Funktion löscht die Verknüpfung zwischen über- und untergeordnetem Dokument.

Parameter	Beschreibung
doc_id_parent	ID des übergeordneten Dokuments Pflichtfeld – entweder als Parameter oder im Kontext als api_doc_id
doc_id_child	ID des untergeordneten Dokuments Wird die doc_id aus dem Kontext genommen, weil sie nicht als Parameter angegeben worden ist, kommt es zu einer Fehlermeldung. Pflichtfeld – entweder als Parameter oder im Kontext als api_doc_id)
user_name	Benutzer, in dessen Namen die Verknüpfung durchgeführt werden soll; wird nichts angegeben, wird der ausführende Benutzer verwendet optional

Rückgabewert	Bedeutung
0	alles OK
> 0	Datenbankfehler
-1	Panther Syntax Fehler, Details siehe LogViewer
-2	Funktion wurde falsch aufgerufen

Rückgabewert	Bedeutung
-3	unbekannter Fehler, dvelop support kontaktieren
-101	weder Dokument ID übergeben, noch im Kontext gesetzt für Vater bzw. Kind
-201	ungültigen Benutzernamen vorgegeben
-219	Vater und Kind Dokument ID sind identisch
-301	ungültige Dokument ID vorgegeben oder Benutzer hat keine Rechte darauf
-302	Fehler beim Ermitteln der Benutzerberechtigungen
-303	Benutzer hat kein Zugriffsrecht auf das Dokument
-312	Benutzer hat kein Recht, Dokumente zu verknüpfen / lösen api_error_var[...] = "right_level" api_error_value[...] = "<Rechtestufe>" api_error_var[...] = "right_link" api_error_value[...] = "<Recht zum Aktenverknüpfen>"
-522	die Dokumente sind nicht auf diese Weise verknüpft
<-9500	Fehler in kundenspezifischer Hook-Funktion hook_unlink api_error_var[...] = "hook_return_code" api_error_value[...] = Rückgabewert des Hooks



verfügbar ab d.3 Version 5.5.1 Hotfix 1



Bei der Funktion link_delete kommt es zu einer Fehlermeldung (-219), wenn beide doc_ids (also doc_id_parent und doc_id_child) nicht übergeben werden.
In diesem Fall würde jeweils die api_doc_id genommen.

Damit wären doc_id_parent und doc_id_child identisch.

In diesem Beispiel wird die Verknüpfung von den Dokumenten Y000050256 und Y00050501 gelöst. Tritt dabei ein Fehler auf, wird dieser angegeben.



```
/ -----
// Beispiel:
// Lösen einer Verknüpfung
// -----
// Variablenklärung
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!")
}

```

5.5 Verzeichnisse

5.5.1 directory_create

Legt ein Verzeichnis auf der Festplatte an.

Parameter	Beschreibung
directory	Verzeichnis (Pflichtfeld)

Rückgabewert	Bedeutung
0	Verzeichnis angelegt
-1	Panther Syntax Fehler, Details siehe LogViewer
-2	Funktion wurde falsch aufgerufen
-160	Verzeichnis nicht vorgegeben
-581	Verzeichnis konnte nicht angelegt werden



verfügbar ab d.3 Version 6.0.1 Hotfix 1

Dieses Beispiel legt das Verzeichnis D:\api_test\dir1 an.



```

// -----
// Beispiel:
// Anlegen eines Verzeichnisses
// ----

// Variablen Deklaration
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!")
}

```

5.5.2 directory_delete

Funktion löscht ein Verzeichnis von der Festplatte.

Parameter	Beschreibung
directory	Verzeichnis Pflichtfeld

Rückgabewert	Bedeutung
0	Verzeichnis gelöscht
-1	Panther Syntax Fehler, Details siehe LogViewer
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler, d.velop support kontaktieren
-160	Verzeichnis nicht vorgegeben
-580	Verzeichnis konnte nicht gelöscht werden



verfügbar ab d.3 Version 6.0.1 Hotfix 1

In diesem Beispiel wird das Verzeichnis D:\api_test\dir1 gelöscht. Tritt ein Fehler auf, wird dieser ausgegeben.



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

// Variablen Deklaration
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!")
}
```

5.5.3 directory_exist

Funktion ermittelt, ob ein Verzeichnis existiert.

Parameter	Beschreibung
directory	Verzeichnis Pflichtfeld

Rückgabewert	Bedeutung
1	Verzeichnis existiert
0	Verzeichnis existiert nicht
-1	Panther Syntax Fehler, Details siehe LogViewer

Rückgabewert	Bedeutung
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler, d.velop Support kontaktieren
-160	Verzeichnis nicht vorgegeben



verfügbar ab d.3 Version 6.0.1 Hotfix 1

Dieses Beispiel prüft, ob das Verzeichnis D:\api_test\dir1 existiert und gibt das Ergebnis aus. Tritt ein Fehler auf, wird dieser ausgegeben.



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

// Variablendeclaration
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!")
}
elseif (h_ReturnValue == 0)
{
    call api_log_info("Ordner existiert nicht!")
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
                        zurückgeliefert!")
}
```

5.5.4 directory_list

Funktion listet den Inhalt eines Verzeichnisses auf (maximal 100 Einträge). Die ermittelten Einträge werden in api_list geschrieben, die Anzahl in api_single_info.

Parameter	Beschreibung
directory	Verzeichnis Pflichtfeld
pattern	Suchmuster optional – default = *, also alles
mode	Modus (Angabe als String): 1 nur Dateien (default) 2 nur Verzeichnisse 3 alles optional – default = 1

Rückgabewert	Bedeutung
0	fehlerfrei ausgeführt
-1	Panther Syntax Fehler, Details siehe LogViewer
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler, d.velop support kontaktieren
-160	Verzeichnis nicht vorgegeben
-580	Fehler beim Listen des Verzeichnisinhaltes



verfügbar ab d.3 Version 6.0.1 Hotfix 1

Dieses Beispiel listet die Unterverzeichnisse des Verzeichnisses D:\api_test auf. Falls ein Fehler auftritt, wird dieser ausgegeben.



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


// Variablendeclaration
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!")
}

```

5.5.5 directory_count_files

Funktion bestimmt die Anzahl der Dateien in einem Verzeichnis nach Suchmuster. Auch Unterordner können eingebunden werden.

Parameter	Beschreibung
directory	Verzeichnis Pflichtfeld
pattern	Suchmuster optional – default = *, also alles
mode	Modus (Angabe als String): 0 nur Verzeichnisse 1 auch Unterverzeichnisse

Rückgabewert	Bedeutung
0	fehlerfrei ausgeführt
-1	Panther Syntax Fehler, Details siehe LogViewer
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler, dvelop support kontaktieren
-160	Verzeichnis nicht vorgegeben
-583	Fehler beim Zählen der Dateien



verfügbar ab d.3 Version 6.2

Anzahl der Dateien in einem Ordner.



Der Pfad zum Verzeichnis ":" und "\" muss mit einem weiteren "\" maskiert werden.



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

// Variablendeclaration
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!")
}
```

5.6 Status

5.6.1 document_block

Diese Funktion sperrt oder entsperrt ein Dokument.

Parameter	Beschreibung
block	0: Dokument entsperren 1: Dokument sperren (default) optional
user_name	

Parameter	Beschreibung
	Benutzer, der (ent)sperrt; falls nichts angegeben ist, wird der Benutzer aus dem Kontext verwendet optional
doc_id	Dokument, das transferiert werden soll; falls nichts angegeben ist, wird die ID aus dem Kontext verwendet. Pflichtfeld – entweder als Parameter oder im Kontext als api_dok_id

Rückgabewert	Bedeutung
0	alles OK
> 0	Datenbankfehler
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler, d.velop support kontaktieren
-101	weder Dokument ID übergeben, noch im Kontext gesetzt
-201	ungültigen Benutzernamen vorgegeben
-301	ungültige Dokument ID vorgegeben oder Benutzer hat keine Rechte darauf
-302	Fehler beim Ermitteln der Benutzerberechtigungen
-303	Benutzer hat kein Zugriffsrecht auf das Dokument
-306	Benutzer hat kein Recht, dieses Dokument zu sperren / entsperren
-506	es existiert keine freigegebene Version api_error_var[...] = "doc_permission" api_error_value[...] = "<Berechtigungen>"
-507	Version in Freigabe ist bereits gesperrt api_error_var[...] = "doc_permission" api_error_value[...] = "<Berechtigungen>"
-508	Version in Freigabe ist nicht gesperrt api_error_var[...] = "doc_permission" api_error_value[...] = "<Berechtigungen>"
< -9500	Fehler in kundenspezifischer Hook-Funktion hook_release_... api_error_var[...] = "hook_return_code" api_error_value[...] = Rückgabewert des Hooks



verfügbar ab d.3 Version 5.5.1

Das erste Beispiel sperrt das Dokument mit der ID Y000050257 und gibt ggf. eine Fehlermeldung aus.
Das zweite Beispiel entsperrt das Dokument mit der ID Y000050257 und gibt ggf. eine Fehlermeldung aus.



```
// -----
// Beispiel A:
// Sperren eines Dokumentes
// -----

// Variablen-deklaration
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!")
}
```



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

// Variablen-deklaration
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 Variablen-deklaration
if (h_ReturnValue == 0)
{
    call api_log_info("Dokument erfolgreich entsperrt!")
```

```

    }
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
                       zurückgeliefert!")
}

```

5.6.2 document_verify

Funktion transferiert ein Dokument aus dem Status „Prüfung, nicht geprüft“ in den Status „Prüfung, geprüft“.

Parameter	Beschreibung
user_name	Benutzer, der die Prüfung ausführen soll; falls nichts angegeben ist, wird der Benutzer aus dem Kontext verwendet optional
doc_id	Dokument, das transferiert werden soll; falls nichts angegeben ist, wird die ID aus dem Kontext verwendet. Pflichtfeld – entweder als Parameter oder im Kontext als api_dok_id

Rückgabewert	Bedeutung
0	alles OK
> 0	Datenbankfehler
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler, d.velop support kontaktieren
-101	weder Dokument ID übergeben, noch im Kontext gesetzt
-201	ungültigen Benutzernamen vorgegeben
-301	ungültige Dokument ID vorgegeben oder Benutzer hat keine Rechte darauf
-302	Fehler beim Ermitteln der Benutzerberechtigungen
-303	Benutzer hat kein Zugriffsrecht auf das Dokument
-307	Benutzer hat kein Recht, dieses Dokument zu prüfen api_error_var[...] = "doc_permission" api_error_value[...] = "<Berechtigungen>"
-509	Version im Status Prüfung ist bereits geprüft api_error_var[...] = "doc_permission" api_error_value[...] = "<Berechtigungen>"
-510	

Rückgabewert	Bedeutung
	es existiert keine Version im Status Prüfung api_error_var[...] = "doc_permission" api_error_value[...] = "<Berechtigungen>"
< -9500	Fehler in kundenspezifischer Hook-Funktion hook_release_... api_error_var[...] = "hook_return_code" api_error_value[...] = Rückgabewert des Hooks



verfügbar ab d.3 Version 5.5.1

Dieses Beispiel prüft das Dokument mit der ID Y000050257. Tritt ein Fehler auf, wird dieser ausgegeben.



```
// -----
// Beispiel:
// Prüfung eines Dokumentes
// -----

// Variablendeclaration
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
    }
    elseif (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! ")
}
```

5.6.3 document transfer



neu in Version 7.1:

Die Funktion darf signierte Dokumente nur noch von Status **Freigabe** in den Status **Archiv** transferieren.

Funktion führt den Statustransfer durch. Es wird jeweils die aktuelle Version verwendet, es sei denn, es ist mindestens eine Archiv-Version vorhanden und ein `archiv_index` angegeben.

Parameter	Beschreibung
destination	Zielstatus {"Bearbeitung", "Prüfung", "Freigabe", "Archiv} Pflichtfeld
user_group_group	User-/Gruppenname erforderlich für destination = Bearbeitung => Benutzer oder Gruppe destination = Prüfung => Gruppe Pflichtfeld, wenn im Parameter destination „Bearbeitung“ angegeben wird
	<p> geändert für Version 7.0: kein Pflichtfeld mehr, wenn im Parameter destination „Prüfung“ angegeben wird</p>
change_remark	Änderungstext: muss angegeben werden, wenn ein Dokument aus der Bearbeitung in die Prüfung überführt werden soll und sich die Nutzdatei gegenüber einer älteren Version im Status Freigabe oder Archiv geändert hat Wird dieser beim Statustransfer B->P angegeben, wird das Dokument nun mit neuer Versionsnummer weitergepflegt. Soll die Versionsnummer gleich bleiben, darf kein Änderungstext angegeben werden. optional
asynchronous	gibt an, ob der Transfer sofort ausgeführt oder aber ein Job dafür angelegt werden soll, welcher der d3 async dann später durchführt (0: sofort (default); 1: später durch d3 async) optional – default: 0

Parameter	Beschreibung
archive_index	Index der Archiv-Version, die zurückgeholt werden soll optional
user_name	Benutzer, der den Transfer ausführen soll; falls nichts angegeben ist, wird der Benutzer aus dem Kontext verwendet optional
doc_id	Dokument, das transferiert werden soll; falls nichts angegeben ist, wird die ID aus dem Kontext verwendet Pflichtfeld – entweder als Parameter oder im Kontext als api_dok_id

Rückgabewert	Bedeutung
0	alles OK
> 0	Datenbankfehler
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler, d.velop support kontaktieren
-101	weder Dokument ID übergeben, noch im Kontext gesetzt
-112	Benutzer- oder Gruppennamen nicht vorgegeben api_error_var[...] = "doc_permission" api_error_value[...] = "<Berechtigungen>" api_error_var[...] = "state" api_error_value[...] = "<Quellstatus>" api_error_var[...] = "editor" api_error_value[...] = "<Quellbearbeiter>"
-113	Gruppenname nicht vorgegeben api_error_var[...] = "doc_permission" api_error_value[...] = "<Berechtigungen>" api_error_var[...] = "state" api_error_value[...] = "<Quellstatus>" api_error_var[...] = "editor" api_error_value[...] = "<Quellbearbeiter>"
<p> neu in d.3 Version 7.0: Fehler -113 tritt nicht mehr im Status „Prüfung“ auf.</p>	
-114	Zielstatus nicht vorgegeben
-115	Archivindex nicht vorgegeben api_error_var[...] = "doc_permission" api_error_value[...] = "<Berechtigungen>"

Rückgabewert	Bedeutung
	<pre>api_error_var[...] = "state" api_error_value[...] = "<Quellstatus>" api_error_var[...] = "editor" api_error_value[...] = "<Quellbearbeiter>"</pre>
-116	Änderungstext nicht vorgegeben <pre>api_error_var[...] = "doc_permission" api_error_value[...] = "<Berechtigungen>" api_error_var[...] = "state" api_error_value[...] = "<Quellstatus>" api_error_var[...] = "editor" api_error_value[...] = "<Quellbearbeiter>"</pre>
-201	ungültigen Benutzernamen vorgegeben
-213	ungültigen Gruppenname vorgegeben <pre>api_error_var[...] = "doc_permission" api_error_value[...] = "<Berechtigungen>" api_error_var[...] = "state" api_error_value[...] = "<Quellstatus>" api_error_var[...] = "editor" api_error_value[...] = "<Quellbearbeiter>"</pre>
-214	ungültigen Zielstatus vorgegeben <pre>api_error_var[...] = "doc_permission" api_error_value[...] = "<Berechtigungen>" api_error_var[...] = "state" api_error_value[...] = "<Quellstatus>" api_error_var[...] = "editor" api_error_value[...] = "<Quellbearbeiter>"</pre>
-301	ungültige Dokument ID vorgegeben oder Benutzer hat keine Rechte darauf
-302	Fehler beim Ermitteln der Benutzerberechtigungen
-303	Benutzer hat kein Zugriffsrecht auf das Dokument
-308	Benutzer hat kein Recht, dieses Dokument freizugeben <pre>api_error_var[...] = "doc_permission" api_error_value[...] = "<Berechtigungen>"</pre>
-309	Benutzer hat kein Recht, dieses Dokument zu transferieren <pre>api_error_var[...] = "doc_permission" api_error_value[...] = "<Berechtigungen>"</pre>
-310	Benutzer hat kein Recht, alte Versionen im Status Archiv zu sehen <pre>api_error_var[...] = "doc_permission" api_error_value[...] = "<Berechtigungen>" api_error_var[...] = "state" api_error_value[...] = "<Quellstatus>" api_error_var[...] = "editor" api_error_value[...] = "<Quellbearbeiter>"</pre>

Rückgabewert	Bedeutung
-511	dieser Transfer ist nicht erlaubt api_error_var[...] = "doc_permission" api_error_value[...] = "<Berechtigungen>" api_error_var[...] = "state" api_error_value[...] = "<Quellstatus>" api_error_var[...] = "editor" api_error_value[...] = "<Quellbearbeiter>"
-512	Es gibt eine neuere Version; um diese zu überschreiben muss sie erst in die Bearbeitung des Benutzers <\:user_group> übertragen werden api_error_var[...] = "doc_permission" api_error_value[...] = "<Berechtigungen>"
-513	Fehler <\:job_error> beim Anlegen des Jobs
-514	Fehler beim Vergleichen der Nutzdateien der alten und der neuen Version api_error_var[...] = "doc_permission" api_error_value[...] = "<Berechtigungen>" api_error_var[...] = "state" api_error_value[...] = "<Quellstatus>" api_error_var[...] = "editor" api_error_value[...] = "<Quellbearbeiter>"
< -9500	Fehler in kundenspezifischer Hook-Funktion hook_transfer_... api_error_var[...] = "hook_return_code" api_error_value[...] = Rückgabewert des Hooks



verfügbar ab d.3 Version 5.5.1

Im ersten Beispiel wird das Dokument mit der angegebenen Dokumenten-ID in den Status „Bearbeitung“ für den Benutzer „dvelop“ geholt.

Im zweiten Beispiel wird das angegebene Dokument in die Prüfung für die Gruppe „test“ verschoben.

Im dritten Beispiel wird das angegebene Dokument freigegeben.



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


// Variablendeclaration
vars h_ReturnValue          // Fehlercode
vars h_DocIdRef = "Y000050257" // Dokumenten-ID des betroffenen
                                // 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!")
}
```



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

// Variablenklärung
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!")
}
```

```
        verschoben! " )
    }
else
{
    call api_log_error("Fehlernummer \
                        :h_ReturnValue wurde zurückgeliefert! ")
}
```



```
// -----
// Beispiel C:
// Ein Dokument freigeben
// -----  
  
// Variablendeclaration
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! ")
}
```

5.7 Notizen

5.7.1 note_add_file

Diese Funktion fügt den Inhalt einer Datei an die Notizdatei des Dokuments an.

Parameter	Beschreibung
file_name	Name der Datei, deren Inhalt an die Notiz gehängt werden soll Pflichtfeld
doc_id	ID des Dokuments, dessen Notiz verändert werden soll; wird nichts angegeben, wird die ID des Kontexts benutzt Pflichtfeld – entweder als Parameter oder im Kontext: <code>api_doc_id</code>
user_name	Benutzer, in dessen Namen die Notiz geschrieben werden soll; wird nichts angegeben, wird der ausführende Benutzer verwendet optional

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler, d.velop support kontaktieren
-101	weder Dokument ID übergeben, noch im Kontext gesetzt
-150	Dateiname nicht vorgegeben
-201	ungültigen Benutzernamen vorgegeben
-250	Datei nicht gefunden
-301	ungültige Dokument ID vorgegeben oder Benutzer hat keine Rechte darauf
-302	Fehler beim Ermitteln der Benutzerberechtigungen
-303	Benutzer hat kein Zugriffsrecht auf das Dokument
-401	Datei konnte nicht ver-/ entschlüsselt werden
-402	ent- / verschlüsselte Datei konnte nicht korrekt umgenannt werden
-403	ent-/ verschlüsselte temporäre Datei konnte nicht gelöscht werden
< -9500	Fehler in kundenspezifischer Hook-Funktion <code>hook_transfer_...</code> <code>api_error_var[...] = "hook_return_code"</code> <code>api_error_value[...] = Rückgabewert des Hooks</code>



verfügbar ab d.3 Version 5.5.1

Dieses Beispiel fügt den Inhalt der Datei `C:\temp\note.txt` an die Notizdatei des Dokumentes mit der ID `Y000050257` an. Tritt dabei ein Fehler auf, wird dieser ausgegeben.



```

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

// Variablen Deklaration
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!")
}

```

5.7.2 note_add_string

Diese Funktion fügt eine Zeile an die Notizdatei des Dokuments an.

Parameter	Beschreibung
line	Zeile, die an die Notiz gehängt werden soll Pflichtfeld
doc_id	ID des Dokuments, dessen Notiz verändert werden soll; wird nichts angegeben, wird die ID des Kontexts benutzt Pflichtfeld – entweder als Parameter oder im Kontext: api_doc_id
user_name	Benutzer, in dessen Namen die Notiz geschrieben werden soll; wird nichts angegeben, wird der ausführende Benutzer verwendet optional

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen

Rückgabewert	Bedeutung
-3	unbekannter Fehler, d.velop support kontaktieren
-101	weder Dokument ID übergeben, noch im Kontext gesetzt
-111	keine Zeile zum Einfügen angegeben
-201	ungültigen Benutzernamen vorgegeben
-301	ungültige Dokument ID vorgegeben oder Benutzer hat keine Rechte darauf
-302	Fehler beim Ermitteln der Benutzerberechtigungen
-303	Benutzer hat kein Zugriffsrecht auf das Dokument
-401	Datei konnte nicht ver-/ entschlüsselt werden
-402	ent- / verschlüsselte Datei konnte nicht korrekt umgenannt werden
-403	ent- / verschlüsselte temporäre Datei konnte nicht gelöscht werden
< -9500	Fehler in kundenspezifischer Hook-Funktion hook_transfer_... api_error_var[...] = "hook_return_code" api_error_value[...] = Rückgabewert des Hooks



verfügbar ab d.3 Version 5.5.1

In diesem Beispiel wird an die Notizdatei für das Dokument mit der ID Y000050257 die angegebene Zeile gehängt. Tritt ein Fehler auf, wird dieser ausgegeben.

```


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


// Variablenklärung
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!")
}

```

5.8 Wiedervorlage

5.8.1 hold_file_send

Diese Funktion sendet ein Dokument an den Postkorb eines d.3 Benutzers. Alle in recipient übergebenen Benutzer / Gruppen werden in api_splitted_string geschrieben.

Parameter	Beschreibung
recipient	Empfänger der Wiedervorlage (d.3 Benutzer / Gruppe); mehrere Empfänger müssen durch Semikolon (;) getrennt werden Pflichtfeld
notice	Betrefftext optional
doc_id	ID des Dokuments, das gesendet werden soll; wird nichts angegeben, wird die ID des Kontexts benutzt Pflichtfeld – entweder als Parameter oder im Kontext: api_dok_id
date_acknowledge	Datum, bis zu dem quittiert werden muss optional
date_remember	Datum, an dem eine Erinnerung erfolgen soll optional
expand_groups	Gruppen auflösen und Benutzer einzeln benachrichtigen optional – default: 0
ignore_checkout	auch versenden, wenn der Empfänger abwesend ist optional – default: 0 0 = nicht versenden, wenn der Empfänger abwesend ist. 1 = auch versenden, wenn der Empfänger abwesend ist.
date_activate	Datum, an dem die Wiedervorlage gesendet wird optional
type	Typ der Wiedervorlage [W=Workflow] optional
sender	Absender; wird nichts angegeben, wird der ausführende Benutzer verwendet optional
chain_id	ID der Wiedervorlage-Kette optional

Parameter	Beschreibung
remove_immediately	1 = Nach dem Senden der Wiedervorlage wird sofort das loesch_flag_sender gesetzt, so dass kein Eintrag bei den gesendeten Wiedervorlagen erscheint.
inherit_class_rule	1 = Ablauf des Rechtes nach Zeitstempel (in diesem Fall muss inherit_class_tstamp gesetzt werden); 2 = Ablauf nach Quittierung aus Postkorb
inherit_class_tstamp	Ablauf-Zeitpunkt der vererbten Rechtes als Zeitstempel der Form DD.MM.YYYY - hh:mm:ss
inherit_class_right	1 = Lese-Rechte 2 = Schreib-Rechte (impliziert auch Lese-Rechte)
check_write_access	prüft, ob der Empfänger Schreib-Rechte auf das Dokument hat
	 neuer Parameter in d.3 Version 7.0

Rückgabewert	Bedeutung
0	alles OK
> 0	Datenbankfehler
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler, dvelop support kontaktieren
-101	weder Dokument ID übergeben noch im Kontext gesetzt
-110	Empfänger nicht vorgegeben
-201	ungültigen Benutzernamen vorgegeben api_error_var[...] = "sender" api_error_value[...] = "<Sendername> - invalid"
-207	ungültigen Empfänger api_error_var[...] = "recipient" api_error_value[...] = "<Empfängername> - invalid"
-208	ungültiges Datum vorgegeben api_error_var[...] = "date_activate_holdfile" api_error_value[...] = Aktivierungsdatum api_error_var[...] = "date_acknowledge_until"

Rückgabewert	Bedeutung
	<pre>api_error_value[...] = Zu quittieren bis-Datum api_error_var[...] = "date_remember_at" api_error_value[...] = Erinnerungsdatum</pre>
-209	ungültige Ketten ID vorgegeben
-212	ungültige Dokument ID vorgegeben
-244	ungültige Regel oder Recht vorgegeben
-301	ungültige Dokument ID vorgegeben oder Benutzer hat keine Rechte darauf <code>api_error_var[...] = "recipient" / "sender"</code> <code>api_error_value[...] = "<Empfänger-/Sendername> - no access right ?"</code>
-302	Fehler beim Ermitteln der Benutzerberechtigungen
-303	Benutzer hat kein Zugriffsrecht auf das Dokument <code>api_error_var[...] = "recipient" / "sender"</code> <code>api_error_value[...] = "<Empfänger-/Sendername> - has no access right for document"</code>
-315	kein Gruppenmitglied hat Zugriffsrecht auf das zu versendende Dokument
-316	Empfänger hat kein Zugriffsrecht; Sender kann Zugriffsrecht aber vererben
<div style="border: 1px solid #0070C0; padding: 10px; text-align: center;">  neuer Rückgabewert in d.3 Version 7.0 </div>	
-317	Sender hat kein Schreibrecht, kann aber Rechte vererben
<div style="border: 1px solid #0070C0; padding: 10px; text-align: center;">  neuer Rückgabewert in d.3 Version 7.0. </div>	
-318	Empfänger hat kein Schreibrecht, kann aber Rechte erben
<div style="border: 1px solid #0070C0; padding: 10px; text-align: center;">  neuer Rückgabewert in d.3 Version 7.0. </div>	
-450	Sender hat kein Recht zur Rechtevererbung
-451	der User besitzt nicht die Rechte, die er vererben will

Rückgabewert	Bedeutung
-452	der Empfänger hat bereits Rechte auf das Dokument
-453	dem User ist keine Dokumentklasse oder Berechtigungsprofil zugewiesen
-454	der Empfänger muss ein Recht auf die zu vererbende Dokumentart besitzen
-455	die temporär erstellte Dokumentklasse wurde dem Benutzer bereits zugewiesen
-503	Empfänger ist abwesend api_error_var[...] = "recipient" api_error_value[...] = "<Empfängername> - checked out" api_error_var[...] = "checkout_text" api_error_value[...] = Abwesenheitstext



verfügbar ab d.3 Version 5.5.1

In dem ersten Beispiel wird das Dokument mit der ID „Y000050257“ in den Postkorb der beiden Benutzer „dvelop“ und „test“ geschoben – auch falls diese abwesend sein sollten.

Als Betreff wird „Test von hold_file_send“ angegeben. Ein Datum, bis zu dem quittiert werden muss bzw. an dem eine Erinnerung erfolgen soll, wird jeweils nicht angegeben. Da u.a. kein Absender als Parameter angegeben ist, wird der ausführende Benutzer als Absender verwendet.

In dem zweiten Beispiel wird der Benutzer „dvelop“ das Dokument „Y000050527“ an den Benutzer „test“ weiterleiten. Dies wird aber nur gelingen, wenn der Benutzer „dvelop“ der bisher letzte Benutzer in der Wiedervorlagekette mit der ID „9“ ist und die Kette mit der ID „9“ das Dokument mit der ID „Y000050257“ betrifft. Bei einem Aufruf dieser Form sollte als Empfänger nur ein Benutzer oder nur eine Gruppe eingetragen sein, da das Senden an einen weiteren Empfänger auf jeden Fall fehlschlagen würde.



```
// -----
// Beispiel A:
// Verschieben eines Dokumentes in den Postkorb
// -----


// Variablendeclaration
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!")
}

```



```

// -----
// Beispiel B:
// Weiterleitung
// -----

// Variablendeclaration
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!")
}

```



`hold_file_send` sollte nicht in einem Entry-Hook aufgerufen werden, wenn noch eine E-Mail Versendung folgt.

5.8.2 `hold_file_find`

Funktion sucht die Benutzer, in deren Wiedervorlage das Dokument liegt.

Die Benutzernamen werden in `api_users` abgelegt.

Die Anzahl der gefundenen Benutzer werden in `api_single_info` abgelegt.

Parameter	Beschreibung
<code>recipient</code>	Empfänger der Wiedervorlage optional
<code>doc_id</code>	ID des Dokumentes, das gesucht werden soll; wird nichts angegeben, wird die ID des Kontexts benutzt Pflichtfeld – entweder als Parameter oder im Kontext: <code>api_dok_id</code>
<code>user_name</code>	Benutzer, in dessen Namen gesucht werden soll; wird nichts angegeben, wird der ausführende Benutzer verwendet optional

Rückgabewert	Bedeutung
0	alles OK
> 0	Datenbankfehler
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler, d.velop support kontaktieren
-101	weder Dokument ID übergeben, noch im Kontext gesetzt
-201	ungültigen Benutzernamen vorgegeben
-301	ungültige Dokument ID vorgegeben oder Benutzer hat keine Rechte darauf

Rückgabewert	Bedeutung
-302	Fehler beim Ermitteln der Benutzerberechtigungen
-303	Benutzer hat kein Zugriffsrecht auf das Dokument



verfügbar ab d.3 Version 5.5.1 Hotfix 1

In dem ersten Beispiel werden alle Postkörbe, bzw. deren Benutzer abgefragt und ausgegeben, in denen das Dokument mit der ID Y000050257 liegt. Tritt ein Fehler auf, wird dieser ausgegeben.

In dem zweiten Beispiel wird geprüft, ob das Dokument mit der ID Y000050257 in dem Postkorb des Benutzers „dvelop“ liegt und das Ergebnis ausgegeben. Tritt ein Fehler auf, wird dieser ausgegeben.



```
// -----
// Beispiel A:
// Abfrage von Postkörben
// -----

// Variablendeclaration
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!")
}
```



```

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

// Variablen Deklaration
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!")
}

```

5.8.3 hold_file_delete

Funktion quittiert empfangene Wiedervorlagen und löscht gesendete Wiedervorlagen.



Der für die Funktion `hold_file_delete` benötigte Parameter `chain_id` kann mit der Funktion `hold_file_get_document_info` ermittelt werden.

Parameter	Beschreibung
<code>chain_id</code>	

Parameter	Beschreibung
	ID der Wiedervorlagen-Kette, zu der der Eintrag gehört Pflichtfeld
sent_received	gesendete (1) oder empfangene (2) Wiedervorlage verarbeiten (optional – default = 2)
workflow_only	0/1: Workflow-Wiedervorlage quittieren (nur für sent_received = 2) optional
recipient	Empfänger (nur für sent_received = 1 oder Empfänger ist eine Gruppe) optional
doc_id	ID des Dokumentes, das gelöscht/quittiert werden soll; wird nichts angegeben, wird die ID des Kontexts benutzt optional
user_name	Benutzer, in dessen Namen gelöscht/quittiert werden soll; wird nichts angegeben, wird der ausführende Benutzer verwendet optional

Rückgabewert	Bedeutung
0	alles OK
> 0	Datenbankfehler
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler, d.velop support kontaktieren
-101	weder Dokument ID übergeben, noch im Kontext gesetzt
-110	keinen Empfänger angegeben
-124	keine KettenID angegeben
-528	Fehler beim Löschen aus der Liste des Senders
-529	Fehler beim Löschen aus der Liste des Empfängers



verfügbar ab d.3 Version 5.5.1 Hotfix 11, 6.0

In diesem Beispiel wird der Empfang des Dokumentes „Y000050257“ in der Wiedervorlagen-Kette mit der ID „8“ quittiert. Tritt ein Fehler auf, wird dieser ausgegeben.



```
// -----
// Beispiel:
// Quittierung eines Dokumentes
```



```
// -----  
  
// Variablen-deklaration  
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!")  
}
```

5.8.4 hold_file_get_document_info

Funktion fragt Informationen zu einer Wiedervorlage ab. Die Daten werden in die globalen Variablen geschrieben:

Globale Variable	Beschreibung
api_hold_file_date	Datum der Wiedervorlage
api_hold_file_recipient	Empfänger
api_hold_file_notice	Betrefftext
api_hold_file_informed	Empfänger benachrichtigt (0/1)
api_hold_file_tstamp_sent	Zeitstempel "Gesendet"
api_hold_file_tstamp_ack	Zeitstempel "Quittiert"
api_hold_file_date_ack_until	Datum "Quittieren bis"
api_hold_file_sender	Sender
api_hold_file_date_remember	Datum "Erinnern am"
api_hold_file_deleted_sender	vom Sender gelöscht (0/1)
api_hold_file_chain_id	Ketten ID

Globale Variable	Beschreibung
api_hold_file_type	Typ (W=Workflow)

Parameter	Beschreibung
sender	Sender Pflichtfeld
recipient	Empfänger Pflichtfeld
doc_id	ID des Dokumentes; wird nichts angegeben, wird die ID des Kontexts benutzt Pflichtfeld – entweder als Parameter oder aus dem Kontext: api_doc_id
user_name	Benutzer, in dessen Namen die Abfrage durchgeführt werden soll; wird nichts angegeben, wird der ausführende Benutzer verwendet optional

Rückgabewert	Bedeutung
0	alles OK
> 0	Datenbankfehler
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler, dvelop support kontaktieren
-101	weder Dokument ID übergeben, noch im Kontext gesetzt
-110	keinen Empfänger angegeben
-125	keinen Sender angegeben
-201	ungültigen Benutzernamen vorgegeben
-301	ungültige Dokument ID vorgegeben oder Benutzer hat keine Rechte darauf
-302	Fehler beim Ermitteln der Benutzerberechtigungen
-303	Benutzer hat kein Zugriffsrecht auf das Dokument
-530	keine Wiedervorlage mit diesem Sender und Empfänger



verfügbar ab d.3 Version 5.5.1 Hotfix 11, 6.0



Die Variable `api_hold_file_chain_id` war lange Zeit nur fünfstellig. Wenn diese nun im System (Datenbank) aber sechsstellig wird, kann mit der Rückgabe nichts angefangen werden, da die ID dann falsch ist.

Wir empfehlen, mindestens Hotfix 15 zu d.3 Version 6.2.1 bzw. mindestens Hotfix 4 zu d.3 Version 6.3.0 zu installieren, wenn z.B. ein anschließendes `hold_file_delete` auf Fehler läuft. Hinweis könnte dementsprechend auch bei `hold_file_delete` hinterlegt werden.

In diesem Beispiel werden Informationen für die Wiedervorlage des Dokumentes mit der angegebenen Dokumenten-ID in die globalen Variablen geschrieben und ausgegeben. Tritt ein Fehler auf, wird dieser ausgegeben.



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


// Variablen Deklaration
vars h_ReturnValue                      // Fehlercode
vars h_SenderUserName = "dvelop"          // Name des Senders
vars h_RecipientUserName = "user1"         // Name des Empfängers
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!")
}
```

5.9 Workflow

5.9.1 workpath_get_document_info

Funktion fragt Infos zu aktuellem Leitweg ab. Die Daten werden in die globalen Variablen geschrieben:

Globale Variable	Beschreibung
api_wfl_header_id	HeaderID
api_wfl_header_name	Workflowname
api_wfl_step_id	StepID
api_wfl_step_exit_string	StepExitString
api_wfl_step_tstamp_entry	TStampEntry
api_wfl_step_type	ElementTyp (1:Schritt, 2: Aktion)
api_wfl_step_recipient	Empfänger
api_wfl_step_action_type	Aktionstyp (api_wfl_step_type=2) (1: Statustransfer, 2:Wiedervorlage, 3: JPL-Hook)
api_wfl_step_action_string_1	für api_wfl_step_action_type=1: Bearbeitung; Prüfung (=Prüfung, ungeprüft); Prüfung,1 (=Prüfung, geprüft); Freigabe für api_wfl_step_action_type=2: Empfänger der Wiedervorlage für api_wfl_step_action_type=3: Name der JPL-Procedure
api_wfl_step_action_string_2	für api_wfl_step_action_type=1: Bearbeiter /Gruppe für api_wfl_step_action_type=2: Betrefftext der Wiedervorlage

Parameter	Beschreibung
doc_id	ID des Dokumentes, das gesucht werden soll; wird nichts angegeben, wird die ID des Kontexts benutzt
user_name	Benutzer, in dessen Namen gesucht werden soll; wird nichts angegeben, wird der ausführende Benutzer verwendet

Rückgabewert	Bedeutung
0	alles OK
> 0	Datenbankfehler
-1	Panther Syntax Fehler, Details siehe d.3 logviewer

Rückgabewert	Bedeutung
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler, d.velop support kontaktieren
-101	weder Dokument ID übergeben, noch im Kontext gesetzt
-201	ungültigen Benutzernamen vorgegeben
-220	ungültige Leitweg-ID vorgegeben; inkonsistente DB-Einträge, bitte d.velop Support kontaktieren
-301	ungültige Dokument ID vorgegeben oder Benutzer hat keine Rechte darauf
-302	Fehler beim Ermitteln der Benutzerberechtigungen
-303	Benutzer hat kein Zugriffsrecht auf das Dokument
-523	das Dokument ist gerade nicht in einem Leitweg aktiv
-524	der Workflow über Leitwege wird nicht unterstützt für dieses Archiv



verfügbar ab d.3 Version 5.5.1 Hotfix 11, 6.0

5.9.2 workpath_end_document

Funktion entfernt Dokument aus dem Leitweg.



neuer Parameter mit Version 6.3.1 Hotfix 1

delete_jobs

JPL Server-API-Funktion `workpath_end_document` entfernt nun optional alle zugehörigen Workflowjobs, die sich ggf. noch in der asynchronen Jobqueue befinden konnten, nachdem ein Dokument erfolgreich aus einem Leitweg entfernt wurde.

Parameter	Beschreibung
doc_id	ID des betroffenen Dokumentes; wird nichts angegeben, wird die ID des Kontexts benutzt
user_name	Benutzer, in dessen Namen gesucht werden soll; wird nichts angegeben, wird der ausführende Benutzer verwendet
delete_jobs	0/1: alle zugehörigen Workflowjobs, die sich noch in der asynchronen Jobqueue befinden, entfernen (default: 0)

Rückgabewert	Bedeutung
0	alles OK

Rückgabewert	Bedeutung
> 0	Datenbankfehler
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler, dvelop support kontaktieren
-101	weder Dokument ID übergeben, noch im Kontext gesetzt
-201	ungültigen Benutzernamen vorgegeben
-301	ungültige Dokument ID vorgegeben oder Benutzer hat keine Rechte darauf
-302	Fehler beim Ermitteln der Benutzerberechtigungen
-303	Benutzer hat kein Zugriffsrecht auf das Dokument
-523	das Dokument ist gerade nicht in einem Leitweg aktiv
-524	der Workflow über Leitwege wird nicht unterstützt für dieses Archiv



verfügbar ab d.3 Version 5.5.1 Hotfix 11, 6.0

Das Dokument mit der angegebenen Dokumenten-ID wird in diesem Beispiel aus dem Leitweg entfernt. Tritt ein Fehler auf, wird dieser ausgegeben.



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


// Variablendeclaration
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! ")
}

```

5.9.3 workpath_start_document

Funktion stellt Dokument in den Leitweg.

Parameter	Beschreibung
wfl_id	ID des Leitweges, in den das Dokument gestellt werden soll
doc_id	ID des betroffenen Dokumentes; wird nichts angegeben, wird die ID des Kontexts benutzt
user_name	Benutzer, in dessen Namen gesucht werden soll; wird nichts angegeben, wird der ausführende Benutzer verwendet

Rückgabewert	Bedeutung
0	alles OK
> 0	Datenbankfehler
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler, d.velop support kontaktieren
-101	weder Dokument ID übergeben, noch im Kontext gesetzt
-122	keine Leitweg ID angegeben
-201	ungültigen Benutzernamen vorgegeben
-220	ungültige Leitweg ID vorgegeben
-301	ungültige Dokument ID vorgegeben oder Benutzer hat keine Rechte darauf
-302	Fehler beim Ermitteln der Benutzerberechtigungen
-303	Benutzer hat kein Zugriffsrecht auf das Dokument
-524	der Workflow über Leitwege wird nicht unterstützt für dieses Archiv
-525	dieser Leitweg kann nicht manuell gestartet werden
-526	das Dokument ist bereits in einem Leitweg aktiv



verfügbar ab d.3 Version 5.5.1 Hotfix 11, 6.0

Für das Dokument mit der Dokumenten-ID “D000000436” wird in diesem Beispiel durch einen Funktionsaufruf der angegebene Workflow gestartet. Tritt ein Fehler auf, wird dieser ausgegeben.



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

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

// Funktionsaufruf
h_ReturnValue = api_function ("workpath_start_document", h_WfId, \
                               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! ")
}
```

5.9.4 workpath_go_to_next_step

Funktion verschiebt das Dokument im Workflow, entweder durch Vorgabe eines Entscheidungswertes oder durch Angabe des nächsten Schrittes.



Diese Funktion kann ab d.3 Version 6.2 nicht mehr aus einem JPL in einem Workflowstep aufgerufen werden für das Dokument, für das das JPL ausgeführt wird.

Die Funktion verschiebt dann innerhalb eines WFL-Step die Position des Dokumentes auf einen anderen Step.

Diese Funktion ist künftig nur noch im Designer abgebildet, da sie beim Aufruf über JPL zu Problemen führen kann.

Parameter	Beschreibung
exit_value	Ausgang, mit dem der aktuelle Schritt abgeschlossen werden soll: 1: noch im Startelement

Parameter	Beschreibung
	2: TRUE 3: FALSE 100: Freihandverschieben
next_step_id	ID des Schrittes, mit dem der Leitweg fortgesetzt werden soll (erforderlich für exit_value = 100)
doc_id	ID des betroffenen Dokumentes; wird nichts angegeben, wird die ID des Kontexts benutzt
user_name	Benutzer, in dessen Namen gearbeitet werden soll; wird nichts angegeben, wird der ausführende Benutzer verwendet

Rückgabewert	Bedeutung
0	alles OK
> 0	Datenbankfehler
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler, d.velop support kontaktieren
-101	weder Dokument ID übergeben, noch im Kontext gesetzt
-201	ungültigen Benutzernamen vorgegeben
-212	ungültige Dokument ID vorgegeben
-220	ungültige Leitweg ID vorgegeben
-221	ungültige Leitweg Schritt ID vorgegeben
-222	ungültigen Ausgangswert vorgegeben
-301	ungültige Dokument ID vorgegeben oder Benutzer hat keine Rechte darauf
-302	Fehler beim Ermitteln der Benutzerberechtigungen
-303	Benutzer hat kein Zugriffsrecht auf das Dokument
-523	das Dokument befindet sich derzeit nicht in einem Leitweg
-524	der Workflow über Leitwege wird nicht unterstützt für dieses Archiv
-527	Fehler beim Bestimmen der aktuellen Position im Leitweg



verfügbar ab d.3 Version 5.5.1 Hotfix 11, 6.0

Das erste Beispiel gibt für das Dokument mit der ID „D000000284“ an dessen aktueller Workflow-Position die Entscheidung TRUE an.

Das zweite Beispiel gibt für das Dokument den nächsten Workflow-Schritt an.



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

// Variablen-deklaration
vars h_ReturnValue           // Fehlercode
vars h_DocRefID = "D000000284" // Dokumenten-ID

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



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

// Variablen-deklaration
vars h_ReturnValue           // Fehlercode
vars h_DocRefID = "D000000284" // Dokumenten-ID
vars next_step_ref_id = "000000000000MT089ITI1OOCP200" // ID des nächsten Workflowschritts

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

5.10 Dokumentart

5.10.1 document_type_get_long

Diese Funktion bestimmt den Dokumentart-Langtext wahlweise zu einer DokumentID oder einem Dokumentartkürzel. Ist beides angegeben, wird die doc_id verwendet. Ist nichts angegeben, wird die doc_id des Kontexts verwendet. Der Langtext wird in api_single_info geliefert.

Parameter	Beschreibung
doc_type_short	Kürzel, dessen Langtext abgefragt wird optional
doc_id	Dokument, dessen Dokumentartlangtext abgefragt wird optional – unter Umständen wird api_doc_id verwendet

Rückgabewert	Bedeutung
0	alles OK
> 0	Datenbankfehler
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-101	weder Dokument ID übergeben noch im Kontext gesetzt
-501	Fehler beim Bestimmen der Dokumentart des Dokuments <\:doc_id>



verfügbar ab d.3 Version 5.5.1

Das erste Beispiel gibt den Dokumentartnamen der per DokumentenID identifizierten Dokumentart aus. Tritt ein Fehler auf, wird dieser ausgegeben.

Im zweiten Beispiel wird der Dokumentartname anhand des Dokumentartkürzels bestimmt und ausgegeben. Tritt ein Fehler auf, wird dieser ausgegeben.



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

// Variablendeclaration
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!")
}
```



```

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

// Variablen Deklaration
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!")
}

```

5.10.2 document_type_get_short

Diese Funktion bestimmt das Dokumentart-Kürzel wahlweise zu einer DokumentID oder einem Dokumentartlangtext.

Ist beides angegeben, wird die `doc_id` verwendet.

Ist nichts angegeben, wird die `doc_id` des Kontexts verwendet.

Der Kurztext wird in `api_single_info` geliefert.

Parameter	Beschreibung
<code>doc_type_long</code>	Langtext, dessen Kürzel abgefragt wird optional
<code>doc_id</code>	Dokument, dessen Dokumentartkürzel abgefragt wird optional – unter Umständen wird <code>api_doc_id</code> verwendet

Rückgabewert	Bedeutung
0	alles OK
> 0	Datenbankfehler

Rückgabewert	Bedeutung
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-101	weder Dokument ID übergeben noch im Kontext gesetzt
-501	Fehler beim Bestimmen der Dokumentart des Dokuments <\:doc_id>
-502	Fehler beim Bestimmen des Dokumentartlangtextes zum Kürzel



verfügbar ab d.3 Version 5.5.1

Das erste Beispiel gibt das Dokumentartkürzel für eine Dokumentart zurück, welche anhand einer DokumentID identifiziert wird. Tritt ein Fehler auf, wird dieser zurückgegeben.

Das zweite Beispiel ruft die Funktion nicht mit einer DokumentID, sondern mit dem Dokumentartnamen auf.



```
// -----
// Beispiel A:
// Abfrage des Dokumentartkürzels anhand ID
// -----

// Variablendeclaration
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!")
}
```



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

// Variablen-deklaration
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!")
}
```

5.11 Wertemengen

5.11.1 predefined_values_get

Die Funktion fragt eine Wertemenge ab.

Die Rückgabe erfolgt in die globale Variable `api_data_type` (=Datentyp der Wertemenge) und die globalen Arrays

`d3server_value_char_allowed[]`
`d3server_value_date_allowed[]`
`d3server_value_num_allowed[]`.

Die Anzahl der gelieferten Werte wird in der Variablen `api_single_info` angegeben.



Die Funktion fragt nur statische Wertemengen ab.

Parameter	Beschreibung
pre_val_id	ID der Wertemenge, deren Werte gelesen werden sollen (hat Vorrang vor pre_val_name)
pre_val_name	Name der Wertemenge, deren Werte gelesen werden sollen (alternativ zu pre_val_id)

Rückgabewert	Bedeutung
0	alles OK
> 0	Datenbankfehler
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-119	weder Wertemengen-ID noch Wertemengen-Name vorgegeben
-218	ungültige Wertemengen-ID oder Wertemengen-Name vorgegeben



verfügbar ab d.3 Version 5.5.1 Hotfix 1

Das erste Beispiel gibt die Werte einer per ID identifizierten Wertemenge an und gibt diese aus. Tritt ein Fehler auf, wird diese ausgegeben.

Das zweite Beispiel erledigt die gleiche Aufgabe. Jedoch wird die Funktion hier mit dem Namen der zu prüfenden Wertemenge aufgerufen.



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

// Variablendeclaration
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 enthielt keine Einträge. ")
}
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
                        zurückgeliefert! ")
}
}

```



```

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

// Variablen-deklaration
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 enthielt keine Einträge. ")
        }
    else
    {
        call api_log_error("Fehlernummer :h_ReturnValue wurde
                           zurückgeliefert! ")
    }
}

```

5.11.2 predefined_values_test

Die Funktion prüft Werte gegen Wertemenge.

Parameter	Beschreibung
test_value	Wert, der getestet werden soll
pre_val_id	ID der Wertemenge, deren Werte gelesen werden sollen (hat Vorrang vor pre_val_name)
pre_val_name	Name der Wertemenge, deren Werte gelesen werden sollen (alternativ zu pre_val_id)

Rückgabewert	Bedeutung
1	Wert ist in der Wertemenge
0	Wert ist nicht in der Wertemenge
> 0	Datenbankfehler
-1	Panther Syntax Fehler, Details siehe d.3 logviewer



verfügbar ab d.3 Version 5.5.1 Hotfix 15, 6.0

Dieser Beispielcode prüft, ob der Wert „Gronau“ in der Wertemenge mit der ID 39 vorhanden ist und gibt das Ergebnis aus. Tritt ein Fehler auf, wird dieser ausgegeben.



```

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

```

```

// Variablendeclaration
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 Eingrag ist in der Wertemenge vorhanden. ")
}
elseif( 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! ")
}

```

5.12 Benutzer

5.12.1 user_get

Funktion für die Abfrage aller Benutzernamen. Das Ergebnis wird im globalen Array `api_users[]` abgelegt. Die Anzahl der gelieferten Werte wird in der Variablen `api_single_info` angegeben.

Parameter	Beschreibung
<code>user_type</code>	1 Nicht-System-Benutzer (default) 2 System-Benutzer (<code>d3_server</code> , ...) 3 System- und Nicht-System-Benutzer



Parameter verfügbar ab d.3 Version 6.0 HF 3

Rückgabewert	Bedeutung
0	alles OK
> 0	Datenbankfehler

Rückgabewert	Bedeutung
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen



verfügbar ab d.3 Version 5.5.1 Hotfix 1

In diesem Beispiel werden alle Systembenutzer abgefragt und ausgegeben. Tritt ein Fehler auf, wird dieser ausgegeben.



```

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

// Variablendeclaration
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]")
    }
}

```

5.12.2 user_is_system

Funktion testet, ob vorgegebener Benutzername ein System-Benutzer ist. Das Ergebnis (0=kein System-Benutzer, 1=System-Benutzer) wird in `api_single_info` geliefert.

Parameter	Beschreibung
user_name	Name des Benutzers der getestet werden soll

Rückgabewert	Bedeutung
0	fehlerfrei
> 0	Datenbankfehler
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen



verfügbar ab d.3 Version 6.0 HF 3

In diesem Beispiel wird geprüft, ob der Benutzer „dvelop“ ein Systembenutzer ist. Das Ergebnis wird ausgegeben. Tritt ein Fehler auf, wird dieser ausgegeben.



```
// -----
// 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! " )
}

```

5.12.3 user_get_groups

Die Funktion gibt an, in welcher Gruppe der Benutzer Mitglied ist.

Das Ergebnis wird im globalen Array `api_groups[]` abgelegt.

Die Anzahl der gelieferten Werte wird in der Variablen `api_single_info` angegeben.



neu in Version 7.1:

In `api_groups_id[]` werden die Gruppen-IDs der ermittelten Benutzergruppen zurückgegeben.

Parameter	Beschreibung
<code>user_name</code>	Benutzer, für den die Gruppen abgefragt werden sollen; wird nichts angegeben, wird der ausführende Benutzer verwendet
<code>group_name</code>	optional; Gruppenname, der geprüft werden soll <code>api_groups[1]</code> enthält dann diesen Gruppennamen, wenn der Benutzer Mitglied ist oder er ist leer, wenn der Benutzer kein Mitglied dieser Gruppe ist

geändert für 7.0:

Parameter:

`group_name` kann jetzt auch die Gruppen-ID sein

Rückgabewert	Bedeutung
0	alles OK
> 0	Datenbankfehler
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-201	ungültigen Benutzernamen vorgegeben



verfügbar ab d.3 Version 5.5.1 Hotfix 1

Das erste Beispiel fragt sämtliche Gruppenmitgliedschaften des Benutzers „dvelop“ ab und gibt diese aus. Hierzu wird die Funktion nur mit dem Benutzernamen als Parameter aufgerufen. Sollte ein Fehler auftreten, wird dieser ausgegeben.

In dem zweiten Beispiel wird die Funktion mit beiden Parametern aufgerufen. So wird geprüft, ob der Benutzer „dvelop“ Mitglied der Gruppe „Vertrieb“ ist. Sollte ein Fehler auftreten, wird dieser angezeigt.



```
// -----
// Beispiel A:
// Abfrage aller Gruppenmitgliedschaften
// -----

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

// Funtionsaufruf: 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! ")
}
```



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

// Variablendeclaration
vars h_ReturnValue
vars h_UserName = "dvelop"    // Benutzername
vars h_GroupName = "Vertrieb" // Gruppenname
```

```

// Funtionsaufruf:
// 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! ")
}

```

5.12.4 user_get_verifier_groups

Die Funktion gibt an, in welcher Gruppe der Benutzer Prüfer ist.

Das Ergebnis wird im globalen Array `api_groups[]` abgelegt.

Die Anzahl der gelieferten Werte wird in der Variablen `api_single_info` angegeben.

Parameter	Beschreibung
<code>user_name</code>	Benutzer, für den die Gruppen abgefragt werden sollen; wird nichts angegeben, wird der ausführende Benutzer verwendet

Rückgabewert	Bedeutung
0	alles OK
> 0	Datenbankfehler
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-201	ungültigen Benutzernamen vorgegeben



verfügbar ab d.3 Version 5.5.1 Hotfix 1

Dieses Beispiel bestimmt, in welchen Gruppen der Benutzer `dvelop` Prüfrechte hat. Die Gruppennamen und die Gesamtzahl der Gruppen werden ausgegeben. Tritt ein Fehler auf, wird die Fehlernummer ausgegeben.



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

// Variablen-deklaration
vars h_ReturnValue          // Fehlercode
vars h_UserName = "dvelop"   // Benutzername
vars h_Count                 // Zähler für Schleife

// Funtionsaufruf:
// 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! ")
}
```

5.12.5 user_get_rolls

Die Funktion gibt an, welche Rollen dem Benutzer zugeordnet sind.

Das Ergebnis wird im globalen Array `api_rolls[]` abgelegt.

Die Anzahl der gelieferten Werte wird in der Variablen `api_single_info` angegeben.

Parameter	Beschreibung
<code>user_name</code>	Benutzer, für den die Rollen abgefragt werden sollen; wird nichts angegeben, wird der ausführende Benutzer verwendet

Rückgabewert	Bedeutung
0	alles OK
> 0	Datenbankfehler
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-201	ungültigen Benutzernamen vorgegeben



verfügbar ab d.3 Version 5.5.1 Hotfix 1

Dieses Beispiel prüft welchen Benutzerprofilen der Benutzer „dvelop“ zugeordnet ist. Die Gesamtzahl der gefundenen Profile und deren Namen werden ausgegeben. Tritt ein Fehler auf, wird die Fehlernummer ausgegeben.



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

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

// Funtionsaufruf:
// 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! ")
}

```

5.12.6 user_get_doc_types

Funktion fragt die Dokumentarten ab, auf die der Benutzer Zugriff hat. Die Kürzel der Dokumentarten werden im globalen Array `api_doc_types[]` abgelegt, die Anzahl gefundener Dokumentarten in der Variablen `api_single_info`.

Parameter	Beschreibung
<code>user_name</code>	Name des Benutzers, der getestet werden soll

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler; dvelop Support kontaktieren



verfügbar ab d.3 Version 6.0 HF 1

Es werden die Dokumentarten – genauer deren Kürzel – abgefragt und ausgegeben, auf die der Benutzer „dvelop“ Zugriff hat. Tritt ein Fehler auf, wird die Fehlernummer ausgegeben.



```

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

```

```

// -----
// Variablen-deklaration
vars h_ReturnValue           // Fehlercode
vars h_UserName = "dvelop"    // Benutzername
vars h_Count                  // Zähler für Schleife

// Funtionsaufruf:
// 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! ")
}

```

5.12.7 user_get_data

Benutzerdaten werden in die globalen Variablen des Benutzerkontextes geschrieben:

Globale Variable	Beschreibung
api_user_name	Benutzername
api_user_right_level	Rechtestufe (1..5)
api_user_right_archive	Recht zur Ansicht von Archivversionen (0/1: nein/ja)
api_user_right_link	Recht zur Aktenverknüpfung (0/1: nein/ja)
api_user_right_flags	Rechteflags (0/1: nein/ja) Position 1: Verknüpfungen erstellen erlaubt Position 2: Webzugriff erlaubt Position 3: Benutzer deaktiviert Position 4: Benutzer gesperrt

Globale Variable	Beschreibung
	Position 5: Admin-Rechte Position 6-10: reserviert
api_user_field[1..10]	benutzerbezogene Felder
api_user_checkout_flag	Benutzer ist abwesend (0/1: nein/ja)
api_user_checkout_text	Abwesenheitstext des Benutzers
api_user_realname	Name
api_user_email	eMail Adresse
api_user_phone	Telefonnummer
api_user_plant	Werk
api_user_department	Abteilung
api_user_tstamp_current_login	Zeitstempel des aktuellen Logins
api_user_tstamp_last_login	Zeitstempel des letzten Logins
api_user_tstamp_hf_info	Zeitstempel der Information über Wiedervorlagen
api_user_ldap_dn	LDAP Name
api_user_change_count	Änderungszahl
api_user_pw_valid_until	Datum bis zu dem das aktuelle Paßwort gültig ist
api_user_tstamp_login_failed	Zeitstempel des Loginversuchs mit falschem Passwort
api_user_count_login_failed	Anzahl der Anmeldeversuche mit falschem Passwort

Parameter	Beschreibung
user_name	Benutzer, für den die Daten abgefragt werden sollen; wird nichts angegeben, wird der ausführende Benutzer verwendet

Rückgabewert	Bedeutung
0	alles OK
> 0	Datenbankfehler
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-201	ungültigen Benutzernamen vorgegeben



verfügbar ab d.3 Version 5.5.1 Hotfix 1

Dieses Beispiel liest die Benutzerdaten des Benutzers „dvelop“ aus und schreibt sie in die globalen Variablen. Diese globalen Variablen werden im Anschluss hieran teilweise ausgegeben. Tritt ein Fehler auf, wird die Fehlernummer ausgegeben.



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

// Variablen-deklaration
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! ")
}
```

5.12.8 user_validate_password

Testet, ob ein Benutzer mit diesem Passwort in d.3 bekannt ist.



Diese Funktion ist nicht für LDAP-Benutzer anwendbar.

Parameter	Beschreibung
password	Passwort, das getestet werden soll
user	Benutzer, der getestet werden soll

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-620	Benutzername/ Passwort ungültig

Testet, ob ein Benutzer mit diesem Passwort in d.3 bekannt ist (kein LDAP).



```
// -----
// Beispiel:
// Prüfen, ob Benutzername und Passwort zusammenpassen
// -----
// Variablenklärung
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!")
}

```

5.12.9 user_get_favourites

Liefert die Dokument-IDs der Favoriten des Benutzers im Array `api_doc_ids` zurück.

Die Anzahl der Favoriten wird im Rückgabewert der Funktion zurückgeliefert und wird zusätzlich in `api_single_info` geschrieben.



Es werden maximal 500 Favoriten zurückgegeben.

Parameter	Beschreibung
<code>user_name</code>	Benutzer, zu dem die Favoriten gelesen werden sollen

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler, d.velop Support kontaktieren



verfügbar ab d.3 Version 6.2

Dokument-IDs der Favoriten eines Benutzers ausgeben.



```

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]")
}
```

5.12.10 user_get_sid_rights



neu in d.3 Version 7.0

Beschreibung: Ermittelt zu einem Benutzer die benutzerspezifischen SID-Rechte/Eigenschaften S-1-*-*-*.

Parameter	Beschreibung
user_name	Benutzer, für den die Daten abgefragt werden sollen; wird nichts angegeben, wird der ausführende Benutzer verwendet.
sid	Optional; SID der Form S-*-*-*-*.* Nur Benutzer-Rechte (S-1-*-*-*.) können abgefragt werden.

Globale Variable	Bedeutung
api_single_info	Anzahl Elemente der folgenden Felder
api_user_sid_str[]	SID-String der Form S-*-*-*-*.*
api_user_sid_right[]	0: Recht/Eigenschaft verweigert/nicht gesetzt 1: Recht/Eigenschaft erteilt/gesetzt

Rückgabewert	Bedeutung
0	alles OK
> 0	Datenbankfehler

5.12.11 user_checkinout



neu in d.3 Version 7.0

Funktion zur An- bzw. Abwesenheitsmeldung von d.3 Benutzern.

Parameter	Beschreibung
in_out	in = Benutzer wird anwesend gemeldet out = Benutzer wird abwesend gemeldet
checkout_text	Abwesenheitstext
check_user	Benutzer, der anwesend bzw. abwesend gemeldet werden soll
call_user	Aufrufender Benutzer, der im Falle, dass check_user gesetzt ist, der Vorgesetzte sein muss. Ist check_user nicht gesetzt, wird call_user verwendet.
force_dist	1: Bewirkt ein sofortiges Umverteilen aller Wiedervorlagen-Einträge des ausgecheckten Benutzers, die über Tätigkeitsprofile gesendet wurden

Rückgabewert	Bedeutung
0	alles OK
3	fehlerhafte oder fehlende Parameter
120	Benutzer (granted bzw. grantee) existieren nicht
420	call_user ist nicht der Vorgesetzte von check_user
423	der ausgewählte Vertreter ist zur Zeit abwesend gemeldet
9000	Datenbank-Fehler (siehe LogViewer)

5.12.12 user_get_superior



neu in d.3 Version 7.0

Funktion zur Ermittlung der Vorgesetzten eines Benutzers.

Parameter	Beschreibung
user_name	Benutzer, für den die Vorgesetzten ermittelt werden sollen. Wird kein Benutzer übergeben, wird der Aufrufer der Funktion verwendet.

Globale Variable	Bedeutung
api_single_info	Anzahl Elemente der folgenden Feldes
api_users[]	Liste der Vorgesetzten

Rückgabewert	Bedeutung
0	alles OK

Rückgabewert	Bedeutung
120	Daten des Benutzers nicht im Cache gefunden

5.13 Gruppen

5.13.1 group_get

Die Funktion fragt Gruppennamen ab. Das Ergebnis wird im globalen Array `api_groups[]` abgelegt. Die Anzahl der gelieferten Werte wird in der Variablen `api_single_info` angegeben.

Parameter	Beschreibung
language	Sprache zur Ermittlung der Gruppen-Namen (Standard = 049)



neuer Parameter in d.3 Version 7.0

Rückgabewert	Bedeutung
0	alles OK
> 0	Datenbankfehler
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen



verfügbar ab d.3 Version 5.5.1 Hotfix 1



neu ab d.3 Version 7.0:

Die IDs zu den Gruppen aus `api_groups[]` wird im globalen Array `api_groups_id[]` abgelegt.



```
/ -----
// Beispiel:
// Abfrage aller Gruppennamen
// -----
// Variablendeclaration
```

```

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! ")
}

```

5.13.2 group_get_members

Die Funktion fragt die Mitglieder einer Gruppe ab. Das Ergebnis wird im globalen Array `api_users[]` abgelegt. Die Anzahl der gelieferten Werte wird in der Variablen `api_single_info` angegeben.

Parameter	Beschreibung
<code>group_name</code>	Name der Gruppe, deren Mitglieder abgefragt werden sollen <div style="border: 1px solid #0070C0; padding: 10px; margin-top: 10px;">  neu in d.3 Version 7.0: Parameter: <code>group_name</code> kann jetzt auch die Gruppen-ID sein </div>
<code>user_name</code>	optional; Benutzername, der geprüft werden soll. <code>api_users[1]</code> enthält dann diesen Benutzernamen, wenn der Benutzer Mitglied ist, oder ist leer, wenn der Benutzer kein Mitglied dieser Gruppe ist
<code>check_subgroups</code>	neu ab d.3 Version 7.2 optional; Standard=0; wenn =1, werden alle Untergruppen von <code>group_name</code> durchsucht

Rückgabewert	Bedeutung
0	alles OK
> 0	Datenbankfehler
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-113	keinen Gruppennamen vorgegeben
-213	ungültigen Gruppennamen vorgegeben



verfügbar ab d.3 Version 5.5.1 Hotfix 1

Das erste der beiden folgenden Beispiele fragt die Mitglieder der Gruppe „Vertrieb“ ab.

Die Gruppe wird hier mit ihrem Namen (`group_name = „Vertrieb“`) identifiziert.

Im Anschluss an den Funktionsaufruf wird entweder das Ergebnis (bei `h_ReturnValue=0`, d.h. kein Fehler) oder der Fehlercode ausgegeben.

Das zweite Beispiel prüft, ob der angegebene Benutzername (`h_UserName = „dvelop“`) Mitglied der angegebenen Gruppe (`group_name = „Vertrieb“`) ist. Im Anschluss an den Funktionsaufruf wird entweder das Ergebnis (bei `h_ReturnValue=0`, d.h. kein Fehler) oder der Fehlercode ausgegeben. Ist im Anschluss an den erfolgreichen Funktionsaufruf kein Name in der Variablen `api_users` gespeichert, ist der Benutzer kein Mitglied der Gruppe.

Ist der Benutzer Mitglied der Gruppe, wird der Benutzername in `api_users` gespeichert.



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

// Variablendeclaration
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! ")
}
```



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

// Variablendeclaration
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! ")
}
```

5.13.3 group_get_verifiers



neu in d.3 Version 7.0:

Da es keine Prüfergruppen mehr gibt, gibt diese Funktion das gleiche zurück wie
`group_get_members`.

Die Funktion fragt den Prüfer einer Gruppe ab. Das Ergebnis wird im globalen Array `api_users[]` abgelegt. Die Anzahl der gelieferten Werte wird in der Variablen `api_single_info` angegeben.

Parameter	Beschreibung
<code>group_name</code>	Name der Gruppe, deren Prüfer abgefragt werden sollen. neu in d.3 Version 7.0: Parameter: <code>group_name</code> kann jetzt auch die Gruppen-ID sein

Rückgabewert	Bedeutung
0	alles OK
> 0	Datenbankfehler
-1	Panther Syntax Fehler, Details siehe LogViewer
-2	Funktion wurde falsch aufgerufen
-113	keinen Gruppennamen vorgegeben
-213	ungültigen Gruppennamen vorgegeben



verfügbar ab d.3 Version 5.5.1 Hotfix 1

Das Beispiel fragt die Prüfer der Gruppe „Vertrieb“ ab.

Alle gefundenen Prüfer werden in dem Array `api_users[]` gespeichert, welches nach erfolgreichem Funktionsaufruf durchlaufen wird, um die Namen auszugeben. Tritt ein Fehler auf, wird die Fehlernummer ausgegeben.



```

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

// Variablen-deklaration
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! ")
}

```

5.13.4 group_get_doc_types



neu in d.3 Version 7.0:
Da es keine direkte Zuordnung von Dokumentarten zu Benutzergruppen mehr gibt, gibt es hier keine Ergebnisse mehr zurück.

Die Funktion fragt Dokumentenarten ab, die einer Gruppe zugeordnet sind. Das Ergebnis wird im globalen Array `api_doc_types[]` abgelegt.

Die Anzahl der gelieferten Werte wird in die Variable `api_single_info` angegeben.

Parameter	Beschreibung
<code>group_name</code>	Name der Gruppe, deren Dokumentarten abgefragt werden sollen

Rückgabewert	Bedeutung
0	alles OK
> 0	Datenbankfehler
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-113	keinen Gruppennamen vorgegeben
-213	ungültigen Gruppennamen vorgegeben



verfügbar ab d.3 Version 5.5.1 Hotfix 1

Das Beispiel fragt alle Dokumentarten ab, die der Gruppe „Vertrieb“ zugeordnet sind.

Die Namen der Dokumentarten werden in dem Array `api_doc_type[]`, die Anzahl wird in die Variable `api_single_info` gespeichert.

Tritt ein Fehler auf, wird die Fehlernummer ausgegeben.



```

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


// Variablendeclaration
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! ")
    }
}

```

5.14 Rollen

5.14.1 roll_get

Funktion fragt alle Rollen_IDs ab. Das Ergebnis wird in das Array `api_rolls[]` geschrieben und die Anzahl der gelieferten Werte in die Variable `api_single_info`.

Rückgabewert	Bedeutung
0	alles OK
> 0	Datenbankfehler
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen



verfügbar ab d.3 Version 5.5.1 Hotfix 1

Das Beispiel fragt mit dem Funktionsaufruf zunächst alle Benutzerprofile ab. Anschließend werden die RollenIDs, welche in dem Array `api_rolls[]` durch die Funktion gespeichert worden sind, ausgegeben. Tritt ein Fehler auf, wird die Fehlernummer ausgegeben.



```

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

// Variablendeclaration
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! ")
}

```

5.14.2 roll_get_names

Die Funktion fragt alle Rollen-Namen ab. Das Ergebnis wird in das Array `api_rolls[]` geschrieben. Die Anzahl der gelieferten Werte wird in die Variable `api_single_info` angegeben.

Rückgabewert	Bedeutung
0	alles OK
> 0	Datenbankfehler
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen



verfügbar ab d.3 Version 5.5.1 Hotfix 1

Dieses Beispiel fragt alle Rollen bzw. deren Name ab und speichert diese in dem Array `api_rolls[]`. Die Anzahl der gefundenen Rollen wird in der Variablen `api_single_info` gespeichert. Die gespeicherten Namen werden im Anschluss an den Funktionsaufruf ausgegeben. Tritt ein Fehler auf, wird die Fehlernummer ausgegeben.



```

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

// Variablendeclaration
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! ")
}

```

5.14.3 roll_get_users

Die Funktion zeigt die Benutzer, denen die Rolle zugeordnet ist.

Das Ergebnis wird in das Array `api_users[]` geschrieben.

Die Anzahl der gelieferten Werte wird in der Variablen `api_single_info` abgelegt.



neu in d.3 Version 7.0:
es können auch Gruppen-IDs in `api_users` zurückgeliefert werden, da man jetzt auch Benutzergruppen Rollen zuweisen kann

Parameter	Beschreibung
<code>roll_id</code>	ID der Rolle, deren zugeordnete Benutzer abgefragt werden sollen (hat Vorrang vor <code>roll_name</code>)
<code>roll_name</code>	Name der Rolle, deren zugeordnete Benutzer abgefragt werden sollen (statt <code>roll_id</code> möglich)

Rückgabewert	Bedeutung
0	alles OK
> 0	Datenbankfehler
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
	Funktion wurde falsch aufgerufen

Rückgabewert	Bedeutung
-2	
-118	weder Rollen-ID noch Rollen-Name vorgegeben
-217	ungültige Rollen-ID oder Rollen-Name vorgegeben



verfügbar ab d.3 Version 5.5.1 Hotfix 1

Das erste Beispiel fragt die Benutzernamen der Benutzer ab, die der Berechtigungsprofil-ID 1 zugeordnet sind. Die Anzahl der gefundenen Benutzer wird in der Variablen `api_single_info`, die Namen der gefundenen Benutzer werden in dem Array `api_users[]` gespeichert. Anschließend werden Anzahl und Benutzernamen ausgegeben. Tritt ein Fehler auf, so wird die Fehlernummer ausgegeben.

In dem zweiten Beispiel werden ebenfalls alle einem Berechtigungsprofil zugeordneten Benutzer, sowie die Gesamtzahl der zugeordneten Benutzer ausgegeben. Allerdings unterscheidet sich der Funktionsaufruf zum ersten Beispiel dadurch, dass anstelle der Berechtigungsprofil-ID der Name des Berechtigungsprofils übergeben wird.



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

// Variablendeclaration
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! ")
}

```



```

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

// Variablen Deklaration
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! ")
}

```

5.14.4 roll_get_data

Die Funktion fragt die Rollendaten (Klasse, Rechte) ab. Gefüllt werden die globalen Variablen:

Globale Variable	Beschreibung
api_roll_id	ID der Rolle
api_roll_name	Name der Rolle

Sowie die globalen Arrays:

Globale Arrays	Beschreibung
api_classes[]	Der Rolle zugeordnete Klassen
api_class_right[]	Recht der Rolle für die Klasse aus api_classes mit dem gleichen Index 0: Recht verweigert 1: Recht erteilt 2: nicht angegeben Position 1: Lesen Position 2: Schreiben Position 3: Prüfen Position 4: Freigeben

Parameter	Beschreibung
roll_id	ID der Rolle, deren Daten abgefragt werden sollen (hat Vorrang vor roll_name)
roll_name	Name der Rolle, deren Daten abgefragt werden sollen (statt roll_id möglich)

Rückgabewert	Bedeutung
0	alles OK
> 0	Datenbankfehler
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-118	weder Rollen-ID noch Rollen-Name vorgegeben
-217	ungültige Rollen-ID oder Rollen-Name vorgegeben



verfügbar ab d.3 Version 5.5.1 Hotfix 1

Dieses Beispiel fragt die Klasse und deren Rechte für das Berechtigungsprofil mit der ID 9 ab. In zwei for-Schleifen werden alle Klassen und je Klasse alle Rechte durchlaufen und ausgegeben. Tritt ein Fehler auf, wird die Fehlernummer ausgegeben.

Der alternative Funktionsaufruf von roll_get_data mit Angabe des Names des Berechtigungsprofils unter Auslassung der entsprechenden ID lautet:

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

Hierzu muss in der Variablen h_RollName der entsprechende Name gespeichert sein. Im Beispielcode sind eine entsprechende Variablendeclaration und der Funktionsaufruf auskommentiert.



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

```
// Variablen-deklaration
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 ")
            }
            elseif (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! ")
}

```

5.15 Restriktionsmengen

5.15.1 d3set_add_filter



neu in d.3 Version 7.0

Fügt einen Filter zu einer bestehenden Restriktionsmenge hinzu.

Sollte die Restriktionsmengen-Zuordnung (`set_name`, `object_id`) noch nicht existieren, wird diese angelegt.

Parameter	Beschreibung
<code>user_name</code>	d.3 Benutzername, in dessen Name die Änderung durchgeführt werden soll.
<code>doc_id</code>	Dokument-ID des Restriktionsmengen-Dokumentes; wenn gesetzt, können <code>set_name</code> und <code>object_id</code> leer bleiben
<code>set_name</code>	Name der Restriktionsmenge
<code>object_id</code>	Benutzername bzw. Gruppen-ID; wenn leer, dann wird nach einer Menge ohne Zuordnung gesucht
<code>filter</code>	Filter-Wert(e), die der Menge hinzugefügt werden sollen; mehrere Filter-Werte können durch ein „;“-Zeichen voneinander getrennt werden; maximale Länge des Filters ist 150 Zeichen
<code>overwrite</code>	0: Filter wird an bisherigen Filter angehängt 1: Bisheriger Filter wird überschrieben

Rückgabewert	Bedeutung
0	alles OK
1505	Restriktionsmengen-Dokument nicht gefunden
1510	keine Zugriffsrecht auf das Mengen-Dokument
1512	kein Schreibrecht auf das Mengen-Dokument
1504	Länge des Mengen-Filters der Restriktionsmenge überschreitet das Maximum von 150.000 Zeichen

Rückgabewert	Bedeutung
1513	die Restriktionsmenge existiert nicht
9000	Datenbank-Fehler

5.15.2 d3set_remove_filter



neu in d.3 Version 7.0

Entfernt einen Filter von einer bestehenden Restriktionsmenge.

Parameter	Beschreibung
user_name	d.3 Benutzername, in dessen Name die Änderung durchgeführt werden soll
doc_id	Dokument-ID des Restriktionsmengen-Dokumentes; wenn gesetzt, können set_name und object_id leer bleiben
set_name	Name der Restriktionsmenge
object_id	Benutzername bzw. Gruppen-ID; wenn leer, dann wird nach einer Menge ohne Zuordnung gesucht
filter	Filter-Wert(e), die der Menge entfernt werden sollen; mehrere Filter-Werte können durch ein „;“-Zeichen voneinander getrennt werden; maximale Länge des Filters ist 150 Zeichen

Rückgabewert	Bedeutung
0	alles OK
1505	Restriktionsmengen-Dokument nicht gefunden.
1510	keine Zugriffsrecht auf das Mengen-Dokument.
1512	kein Schreibrecht auf das Mengen-Dokument.
1504	Länge des Mengen-Filters der Restriktionsmenge überschreitet das Maximum von 150.000 Zeichen
9000	Datenbank-Fehler

5.15.3 d3set_get_filter



neu in d.3 Version 7.0

Ermittelt den Filter einer bestehenden Restriktionsmenge.

Der Filter wird in den Kontextvariablen `api_doc_field_6x[...]` zurückgeliefert.

Parameter	Beschreibung
<code>user_name</code>	d.3 Benutzername, in dessen Name die Änderung durchgeführt werden soll
<code>doc_id</code>	Dokument-ID des Restriktionsmengen-Dokumentes; wenn gesetzt, können <code>set_name</code> und <code>object_id</code> leer bleiben
<code>set_name</code>	Name der Restriktionsmenge
<code>object_id</code>	Benutzername bzw. Gruppen-ID; wenn leer, wird nach einer Menge ohne Zuordnung gesucht

Rückgabewert	Bedeutung
0	alles OK
1505	Restriktionsmengen-Dokument nicht gefunden
1510	keine Zugriffsrecht auf das Mengen-Dokument
1512	kein Schreibrecht auf das Mengen-Dokument
1504	Länge des Mengen-Filters der Restriktionsmenge überschreitet das Maximum von 150.000 Zeichen

Bei der Restriktionsmengen-Zuordnung handelt es sich um ein Dokument im d.3 in der Dokumentart \$SET\$.

Die 60er-Felder dieses Dokuments werden zur Speicherung der Mengensuchfilter verwendet. Als Trennzeichen der einzelnen Attribute wird ein Semikolon verwendet.
`d3set_get_filter` füllt die 60er Felder in den `api_doc_fields`.

Der Filter der Restriktionsmengen-Zuordnung steht in den 60er Feldern und kann sich im Maximalfall über alle möglichen 60er Felder (Feld 60, Zeile 1 - Feld 69, Zeile 100) erstrecken.

Der Filter wird zurückgegeben 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]`

Wenn das erste dieser Felder leer ist, gibt es keine weiteren Einträge mehr.

Dies bedeutet, wenn `api_doc_field_60[1]` einen Wert hat, aber `api_doc_field_60[2]` leer ist, folgen auch keine weiteren Werte.

5.16 Tiff-Funktionen

5.16.1 document_render

`document_render` erzeugt wahlweise TIFF oder PDF (Dokument oder Datei rendern).



Diese Funktion ersetzt die Funktion `tiff_create`. Unterschied: `tiff_create` erzeugt nur TIFF; `document_render` erzeugt wahlweise TIFF oder PDF.



verfügbar ab d.3 Version 6.1

Parameter	Beschreibung
<code>source</code>	Quelldatei; erforderlich, falls nicht das Dokument <code>doc_id</code> (neueste Version) gerendert werden soll.
<code>destination</code>	Zieldatei; erforderlich, falls das erzeugte Render-Dokument nicht als abhängige t1 oder p1 Datei zur <code>doc_id</code> abgelegt werden soll.
<code>render_option</code>	0: Es soll keine Render-Datei geliefert werden, z.B. lediglich die OCR-Datei 1 oder " ": TIFF-Datei als Render-Datei erzeugen 2: PDF-Datei als Render-Datei erzeugen 3: TIFF- und PDF-Datei als Render-Dateien default: 1
<code>ocr</code>	Datei verschlagworten default: 0
<code>asynchronous</code>	asynchron rendern, d.h. es wird nur ein Job angelegt (<code>job_id</code> wird für <code>error_var[...]="job_id"</code> geliefert; diese Option steht nur für das dokumentbasierte Rendern zur Verfügung, d.h. <code>doc_id</code> muss vorhanden sein ab 6.2 default: 1, vorher 0 ab 6.2 es wird nur noch der Wert 1 unterstützt
<code>replace_doc</code>	erzeugtes Render-Dokument ersetzt das Stammdokument im d.3 (siehe auch Parameter <code>ERLAUBE_STAMM_2_TIFF_KONVERT</code>) default: 0
<code>overwrite</code>	Zieldatei ersetzen, falls schon vorhanden default: 0
<code>doc_id</code>	Dokument, das gerendert / zu dem ein abhängiges t1 bzw. p1 Dokument erzeugt werden soll; falls nichts angegeben ist, wird die Dokumenten-ID aus dem Kontext verwendet



Parameter	Beschreibung
user_name	Benutzer in dessen Namen der Aufruf erfolgt; wird nichts angegeben, wird der ausführende Benutzer verwendet
status	Status der Version, die gerendert werden soll. Ist nichts angegeben, wird die aktuelle Version verwendet (=B, Bearbeitung, P, Prüfung, F, Freigabe, A, Archiv)
archiv_index	Index der Archivversion, die gerendert werden soll (nur bei status="A"/"Archiv" nötig)
prio	Gültige Werte sind low, normal, high Wird ein anderer Wert übergeben, wird immer "normal" verwendet; steht auch ohne Ausführen von <code>upd721.jpl</code> in 7.2.1 zur Verfügung.

Rückgabewert	Bedeutung
0	alles OK
>0	Datenbank Fehler
-1	Panther Syntax Fehler, Details siehe LogViewer
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler, dvelop Support kontaktieren
-101	weder Dokument ID übergeben, noch im Kontext gesetzt
-151	Quelldateiname nicht vorgegeben
-152	Zieldateiname nicht vorgegeben
-201	ungültiger Benutzername vorgegeben
-301	ungültige Dokumenten-ID vorgegeben oder Benutzer hat keine Rechte darauf
-302	Fehler beim Ermitteln der Benutzerberechtigungen
-303	Benutzer hat kein Zugriffsrecht auf das Dokument
-543	Fehler in der Rendition Service Verbindung
-544	Rendition Service ist nicht verfügbar
-545	Dokumente im Status Archiv dürfen nicht gerendert werden
-546	Platzhalter des bidirektionalen Proxy dürfen nicht gerendert werden
-547	DTIFF_SUPPORT ist nicht aktiviert <code>api_error_var[...] = "dtiff_support"</code> <code>api_error_value[...] = DTIFF_SUPPORT</code>
-557	Zieldatei existiert bereits und soll nicht überschrieben werden
-558	bei <code>asynchronous=1</code> darf die Länge von Pfad und Name der Zieldatei 100 Zeichen nicht übersteigen
	Dokument konnte nicht vom externen Storage bereitgestellt werden

Rückgabewert	Bedeutung
-587	
-588	Dokumente, die in der Bearbeitung des d.3 Servers sind, können nicht gerendert werden
-589	Status des Dokuments hat sich während des Vorgangs verändert

5.16.2 tiff_create



tiff_create erzeugt nur TIFF.

Sie sollten die neue Funktion document_render verwenden, die wahlweise TIFF oder PDF erzeugt.

Funktion vertifft Dokument oder Datei.

Parameter	Beschreibung
source	Quelldatei; erforderlich, falls nicht das Dokument doc_id (neueste Version) vertifft werden soll
destination	Zieldatei; erforderlich, falls das erzeugte Tiff nicht als abhängige t1 Datei zu doc_id abgelegt werden soll
tiff	Tiff-Datei erzeugen (default: 1)
ocr	Datei verschlagworten (default: 0)
asynchronous	Asynchron vertiffen, d.h. es wird nur ein Job angelegt (job_id wird für error_var[...]="job_id" geliefert); diese Option steht nur für das dokumentbasierte Vertiffen zur Verfügung, d.h. doc_id muss vorhanden sein default: 0
replace_doc	Erzeugtes TIF ersetzt das Stammdokument im d.3 (siehe auch Parameter ERLAUBE_STAMM_2_TIFF_KONVERT) default: 0
overwrite	Zieldatei ersetzen, falls schon vorhanden default: 0
doc_id	Dokument, das vertifft / zu dem ein abhängiges t1 Dokument erzeugt werden soll; falls nichts angegeben ist, wird die Dokumenten-ID aus dem Kontext verwendet
user_name	Benutzer in dessen Namen der Aufruf erfolgt; wird nichts angegeben, wird der ausführende Benutzer verwendet

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler, d.velop Support kontaktieren
-101	weder Dokument ID übergegeben noch im Kontext gesetzt
-151	Quelldateiname nicht vorgegeben
-152	Zieldateiname nicht vorgegeben
-201	ungültiger Benutzername vorgegeben
-301	ungültige Dokumenten-ID vorgegeben oder Benutzer hat keine Rechte darauf
-302	Fehler beim Ermitteln der Benutzerberechtigungen
-303	Benutzer hat kein Zugriffsrecht auf das Dokument
-543	Fehler in der dtiffVerbindung
-544	dtiff ist nicht verfügbar
-545	Dokumente im Status Archiv dürfen nicht vertifft werden
-546	Platzhalter des bidirektionalen Proxy dürfen nicht vertifft werden
-547	DTIFF_SUPPORT ist nicht aktiviert api_error_var[...] = "dtiff_support" api_error_value[...] = DTIFF_SUPPORT
-557	Zieldatei existiert bereits und soll nicht überschrieben werden
-558	bei asynchronous=1 darf die Länge von Pfad und Name der Zieldatei 100 Zeichen nicht übersteigen
<-9500	Fehler in kundenspezifischer Hookfunktion hook_rendition_..



verfügbar ab d.3 Version 6.0

5.16.3 tiff_concat

Die Funktion hängt ein Tiff-Dokument an ein anderes.

Parameter	Beschreibung
source	Quell-Tiff-Datei

Parameter	Beschreibung
destination	Ziel-Tiff-Datei
source_rdl	Quell-Redlining-Datei (optional)
destination_rdl	Ziel-Redlining-Datei (optional)

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler, d.velop Support kontaktieren
-151	Quelldateiname nicht vorgegeben
-152	Zieldateiname nicht vorgegeben
-559	Fehler beim Aneinanderfügen der Tiff-Dateien
-560	Anzahl der Zielseiten ungleich Summe der Quellseiten
-561	Quell-Tif-Datei nicht vorhanden
-562	Quell-Tif-Datei nicht unterstützt (evtl. korrupt oder falsches Format)
-563	Ziel-Tif-Datei kann nicht geöffnet werden
-564	Quell-Redlining-Datei nicht unterstützt (evtl. korrupt oder falsches Format)
-565	Quell-Redlining-Datei enthält nicht zugeordnete Seiten



verfügbar ab d.3 Version 6.0



neu in d.3 Version 7.0

Vertifft die Workflow-Protokolle zu einer Dokument-ID und erzeugt abhängige w1-Datei. Existiert bereits eine w1-Datei, führt die Funktion keine Aktion aus (Rückgabe 1). Wenn neu vertifft werden soll, weil z. B. ein weiteres Protokoll zur Dokument-ID hinzugefügt wurde, muss das abhängige w1 zunächst entfernt werden.

Parameter	Beschreibung
doc_id (P)	Dokument, dessen Workflow-Protokolle vertifft werden sollen
user_name	Name des ausführenden Benutzers

Rückgabewert	Bedeutung
1	vertifftes Workflow-Protokoll existiert bereits zu dieser Dokument-ID
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-630	temporäre Sammeldatei aller Workflow-Protokolle der Dokument-ID konnte nicht angelegt werden, Details siehe d.3 logviewer
-631	temporäre Sammeldatei aller Workflow-Protokolle konnte nicht erfolgreich abgeschlossen werden, Details siehe d.3 logviewer
-632	ein Workflow-Protokoll zur Dokument-ID konnte nicht gefunden werden, Details siehe d.3 logviewer

5.17 Dateizugriffe

5.17.1 file_add_string

Diese Funktion schreibt einen String in eine Datei. Ist die Datei im angegebenen Verzeichnis nicht vorhanden, wird sie erzeugt. Ist die Datei vorhanden, wird der String wahlweise an die Datei angehängt oder die Datei wird überschrieben.



geändert ab 7.1.0 HF5:

Die globale Variable `UTF8_BOM` kann genutzt werden, um ein UTF8 Byte-Order-Mark vor eine Datei zu schreiben.

Es war nicht ohne weiteres möglich, ein UTF8 Byte-Order-Mark vor eine Textdatei zu schreiben, die man mit der Server-API-Funktionen `file_add_line()` oder

`file_add_string()` geschrieben hat.

Hierfür kann nun die globale Variable `UTF8_BOM` genutzt werden.

Das BOM kann nun mit `call api_function('file_add_string', file_name, UTF8_BOM, 'OVERWRITE')` an den Anfang der Datei gesetzt werden, bevor man den eigentlich Inhalt der Datei im 'APPEND'-Modus schreibt. Hierbei hat der Hook-Programmierer dafür Sorge zu tragen, dass alle folgenden Zeichenfolgen in UTF8-Kodierung an die API-Funktionen übergeben werden.

Parameter	Beschreibung
file_name	Name und Pfad der Datei, in die geschrieben werden soll
text	String, der in die Datei geschrieben werden soll
mode	Modus der Dateioperation: APPEND: Ist die Datei vorhanden, wird der Text angehängt (Standard) OVERWRITE: Ist die Datei vorhanden, wird die Datei überschrieben. Wird kein Wert übergeben, so wird APPEND benutzt.
count	Anzahl der Zeichen, die mindestens geschrieben werden sollen. Ist <text> kürzer, werden Leerzeichen angehängt.

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler, d.velop Support kontaktieren
-150	Dateiname nicht vorgegeben
-210	ungültigen Modus vorgegeben
-552	Datei kann nicht zum Überschreiben geöffnet werden
-553	Datei kann nicht zum Anhängen geöffnet werden



verfügbar ab d.3 Version 5.5.1

Dieses Beispiel ergänzt die Datei c:\text.txt um den Text „Ich, dieser Text, werde in die Datei geschrieben.“ Tritt ein Fehler auf, wird die Fehlernummer ausgegeben.



```

/ -----
// 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! ")
}

```

5.17.2 file_add_line

Diese Funktion schreibt in eine neue Zeile der spezifizierten Datei. Ist die Datei im angegebenen Verzeichnis nicht vorhanden, wird sie erzeugt. Ist die Datei vorhanden, wird eine neue Zeile wahlweise an die Datei angehängt oder die Datei wird überschrieben.



geändert ab 7.1.0 HF5:

Die globale Variable `UTF8_BOM` kann genutzt werden, um ein UTF8 Byte-Order-Mark vor eine Datei zu schreiben.

Es war nicht ohne weiteres möglich, ein UTF8 Byte-Order-Mark vor eine Textdatei zu schreiben, die man mit der Server-API-Funktionen `file_add_line()` oder

`file_add_string()` geschrieben hat.

Hierfür kann nun die globale Variable `UTF8_BOM` genutzt werden.

Das BOM kann nun mit `call api_function('file_add_string', file_name, UTF8_BOM, 'OVERWRITE')` an den Anfang der Datei gesetzt werden, bevor man den eigentlich Inhalt der Datei im 'APPEND'-Modus schreibt.

Hierbei hat der Hook-Programmierer dafür Sorge zu tragen, dass alle folgenden Zeichenfolgen in UTF8-Kodierung an die API-Funktionen übergeben werden.

Parameter	Beschreibung
<code>file_name</code>	Name und Pfad der Datei, in die geschrieben werden soll
<code>line</code>	Zeile, die in die Datei geschrieben werden soll
<code>mode</code>	Modus der Dateioperation: <code>APPEND</code> : Ist die Datei vorhanden, wird der Text angehängt (Standard) <code>OVERWRITE</code> : Ist die Datei vorhanden, wird die Datei überschrieben Wird kein Wert übergeben, wird <code>APPEND</code> benutzt.
<code>count</code>	Anzahl der Zeichen, die mindestens geschrieben werden sollen. Ist <code><line></code> kürzer, werden Leerzeichen angehängt.

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler, d.velop Support kontaktieren
-150	Dateiname nicht vorgegeben
-210	ungültigen Modus vorgegeben
-552	Datei kann nicht zum Überschreiben geöffnet werden
-553	Datei kann nicht zum Anhängen geöffnet werden



verfügbar ab d.3 Version 5.5.1

In diesem Beispiel wird in die Datei C:\text.txt eine neue Zeile mit dem Inhalt der in der Variablen h_Line angegebenen Zeichenkette geschrieben. Tritt ein Fehler auf, wird die Fehlernummer ausgegeben.



```

// -----
// Beispiel:
// Zeichenkette in eine bestimmte Zeile einer Datei schreiben
// -----

// Variablendeclaration
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! ")}
```

5.17.3 file_read_line

Liest eine Zeile aus einer Datei; Ergebnis wird in die Variable `api_single_info` geschrieben; befindet sich das Dateiende vor der spezifizierten Zeile, die gelesen werden soll, enthält `api_single_info` "EOF"

Parameter	Beschreibung
<code>file_name</code>	Dateiname
<code>line</code>	Startzeile
<code>mode</code>	Leserichtung: BOF=vorwärts, EOF=rückwärts
<code>count</code>	Anzahl Zeichen maximal lesen

Rückgabewert	Bedeutung
<code>>=0</code>	alles OK; entspricht Anzahl der gefundenen Zeichen
<code>-1</code>	Panther Syntax Fehler, Details siehe d.3 logviewer
<code>-2</code>	Funktion wurde falsch aufgerufen
<code>-3</code>	unbekannter Fehler, d.velop Support kontaktieren
<code>-150</code>	Dateiname nicht vorgegeben
<code>-210</code>	ungültige Leserichtung vorgegeben
<code>-550</code>	Datei kann nicht zum Lesen geöffnet werden



Die Funktion eignet sich (derzeit) nicht, um große Dateien zu lesen und zu verarbeiten. Wenn z.B. Zeile 10000 einer Datei eingelesen werden soll, so werden aktuell (intern) leider die 9999 Zeilen davor ebenfalls nacheinander eingelesen, weil man nicht direkt zu Zeile 10000 springen kann.

Bei großen Dateien empfiehlt es sich also vorerst noch auf die Panther-Funktionen direkt zuzugreifen (es sei denn es muss immer nur eine der ersten Zeilen gelesen werden). Mit einer der nächsten Server-Versionen wird voraussichtlich ein verbessertes Lesen von Dateien ermöglicht.

5.17.4 file_concat

Diese Funktion hängt den Inhalt einer Datei an eine andere an.

Parameter	Beschreibung
<code>source</code>	Quelldatei, deren Inhalt angehängt werden soll
<code>destination</code>	Zieldatei, an die angehängt werden soll

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler, d.velop Support kontaktieren
-151	Quelldateiname nicht vorgegeben
-152	Zieldateiname nicht vorgegeben
-550	Datei kann nicht zum Lesen geöffnet werden
-551	Datei kann nicht zum Schreiben geöffnet werden
-554	Fehler beim Schreiben der Datei



verfügbar ab d.3 Version 5.5.1

5.17.5 file_copy

Diese Funktion kopiert eine Datei.

Parameter	Beschreibung
source	Quelldatei
destination	Zieldatei

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler, d.velop Support kontaktieren
-151	Quelldateiname nicht vorgegeben
-152	Zieldateiname nicht vorgegeben
-504	Fehler beim Allokieren von Speicher
-550	Datei kann nicht zum Lesen geöffnet werden
-551	Datei kann nicht zum Schreiben geöffnet werden

Rückgabewert	Bedeutung
-554	Fehler beim Schreiben der Datei



verfügbar ab d.3 Version 5.5.1

In dem Beispiel wird die Datei C:\text.txt kopiert. Besteht die Datei C:\text2.txt noch nicht, wird sie mit dem Inhalt von C:\text.txt erstellt. Besteht sie jedoch bereits, wird der Inhalt mit dem Inhalt von C:\text.txt überschrieben. Tritt ein Fehler auf, wird die Fehlernummer ausgegeben.



```
// -----
// Beispiel:
// Datei kopieren
// -----

// Variablen-deklaration
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! ")
}
```

5.17.6 file_delete

Diese Funktion löscht eine Datei.

Parameter	Beschreibung
file_name	Datei, die gelöscht werden soll

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen

Rückgabewert	Bedeutung
-3	unbekannter Fehler, d.velop Support kontaktieren
-150	Dateiname nicht vorgegeben
-250	Datei nicht gefunden
-350	Dateizugriff verweigert



verfügbar ab d.3 Version 5.5.1

Dieses Beispiel löscht die Datei c:\text2.txt. Tritt ein Fehler auf, wird die Fehlernummer ausgegeben.



```
// -----
// Beispiel:
// Löschen einer Datei
// -----

// Variablendeclaration
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! ")
}
```

5.17.7 file_compare

Diese Funktion vergleicht zwei Dateien.

Parameter	Beschreibung
file_name1	Datei 1
file_name2	Datei 2

Rückgabewert	Bedeutung
4	Dateiinhalt unterschiedlich

Rückgabewert	Bedeutung
3	Dateigröße unterschiedlich
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler, d.velop Support kontaktieren
-153	Dateiname 1 nicht vorgegeben
-154	Dateiname 2 nicht vorgegeben
-504	Fehler beim Allokieren von Speicher
-550	Datei kann nicht zum Lesen geöffnet werden



verfügbar ab d.3 Version 5.5.1

Das folgende Beispiel vergleicht die beiden Dateien C:\text.txt und C:\text2.txt miteinander und gibt das Ergebnis aus. Tritt ein Fehler auf, wird die Fehlernummer ausgegeben.



```

/ -----
// Beispiel:
// Vergleich von Dateien
// -----

// Variablendeclaration
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! ")
    }
}

```

5.17.8 file_rename

Diese Funktion benennt eine Datei um.

Parameter	Beschreibung
source	alter Dateiname
destination	neuer Dateiname

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler, dvelop Support kontaktieren
-151	Quelldateiname nicht vorgegeben
-152	Zieldateiname nicht vorgegeben
-555	Fehler beim Umbenennen der Datei



verfügbar ab d.3 Version 5.5.1

Dieses Beispiel benennt die Datei C:\text.txt in text2.txt um. Tritt ein Fehler auf, wird die Fehlernummer ausgegeben.



```

// -----
// Beispiel:
// Umbenennen einer Datei
// -----

```

```

// Variablen Deklaration
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! ")
}

```

5.17.9 file_exist

Diese Funktion prüft, ob eine Datei existiert.

Parameter	Beschreibung
file_name	Dateiname

Rückgabewert	Bedeutung
1	Datei existiert
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler, d.velop Support kontaktieren
-150	Dateiname nicht vorgegeben



verfügbar ab d.3 Version 5.5.1

Es wird geprüft, ob die Datei C:\\text.txt bereits existiert. Das Ergebnis wird ausgegeben. Tritt ein Fehler auf, wird die Fehlernummer ausgegeben.



```

/ -----
// Beispiel:
// Prüfen, ob Datei existiert
// -----

// Variablen Deklaration
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! ")
}

```

5.18 Stringkonvertierung

5.18.1 string_to_lower

Diese Funktion wandelt einen String in Kleinbuchstaben um. Der konvertierte String wird in die globale Variable `apiConvertedString` geschrieben.



ab Version 6.2.: Strings beliebiger code pages können konvertiert werden

Parameter	Beschreibung
text	String, der konvertiert werden soll
code_page	code page des übergebenen strings in Kleinbuchstaben konvertieren optional; default "1252"; ab Version 6.2

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen



verfügbar ab d.3 Version 5.5.1

In diesem Beispiel wird der String „Anton Berta Caesar Dora Emil“ in den String „anton berta caesar dora emil“ umgewandelt. Tritt ein Fehler auf, wird die Fehlernummer ausgegeben.



```
// -----
// Beispiel:
// Zeichenkette in Kleinbuchstaben konvertieren
// -----

// Variablendeclaration
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: :apiConvertedString")
}
// Fehlerbehandlung
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
        zurückgeliefert! ")
}
```

5.18.2 string_to_upper

Diese Funktion wandelt einen String in Großbuchstaben um. Der konvertierte String wird in die globale Variable `apiConvertedString` geschrieben.



ab Version 6.2.: Strings beliebiger code pages können konvertiert werden

Parameter	Beschreibung
text	String, der konvertiert werden soll
code_page	code page des übergebenen strings in Großbuchstaben konvertieren optional; default "1252"; ab Version 6.2

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen



verfügbar ab d.3 Version 5.5.1

Dieses Beispiel wandelt den String „Anton Berta Caesar Dora Emil“ in den String „ANTON BERTA CAESAR DORA EMIL“ um. Tritt ein Fehler auf, wird die Fehlernummer ausgegeben.



```
// -----
// Beispiel:
// Zeichenkette in Grossbuchstaben konvertieren
// -----

// Variablendeclaration
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! ")
}

```

5.18.3 string_to_number

Diese Funktion wandelt einen String in das JPL Zahlenformat, d. h. ohne Tausendertrenner und mit dem Punkt „.“ als Dezimaltrenner um. Der konvertierte String wird in die globale Variable `apiConvertedString` geschrieben.

Parameter	Beschreibung
<code>text</code>	String, der konvertiert werden soll
<code>data_type</code>	Datentyp; zulässig sind <code>M=MONEY</code> und <code>N=NUMERIC</code> ; entspricht dem jeweiligen d.3 Datentyp und ist erforderlich für die Bestimmung des Tausendertrenners. Wird nichts angegeben, wird <code>NUMERIC</code> als Standard verwendet

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-206	ungültigen Datentyp vorgegeben



verfügbar ab d.3 Version 5.5.1

5.18.4 string_to_date

Die Funktion formatiert einen String in das d.3 Standard-Datums-Format. Der konvertierte String wird in die globale Variable `apiConvertedString` geschrieben. Enthält `text` kein "ordentliches" Datum, wird ein Leerstring in die globale Variable geschrieben.

Parameter	Beschreibung
<code>text</code>	String, der konvertiert werden soll

Rückgabewert	Bedeutung
0	alles OK

Rückgabewert	Bedeutung
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-208	ungültiges Datum vorgegeben



verfügbar ab d.3 Version 5.5.1 Hotfix 11, 6.0

Der folgende Beispielcode wandelt das Datum des Strings in das d.3-Datumformat um. Tritt ein Fehler auf, wird die Fehlernummer ausgegeben.



```
// -----
// Beispiel:
// Umwandeln eines String-Datums in ein d.3-Datum
// -----

// Variablendeclaration
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: :apiConvertedString")
}
// Fehlerbehandlung
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
                        zurückgeliefert! ")
}
```

5.18.5 string_split

Diese Funktion splittet einen String an einem Separator auf.

Die Teilstrings werden im globalen Array `api_splitted_string[]` gespeichert.



Falls kein Fehler aufgetreten ist, wird die Anzahl der ermittelten Werte durch die Funktion zurückgegeben und nicht in der Variable `api_single_info` gespeichert. Tritt ein Fehler auf, wird die Fehlernummer ausgegeben.

Parameter	Beschreibung
<code>text</code>	String, der aufgesplittet werden soll
<code>separator</code>	Zeichen(kette), an dem / der gesplittet werden soll

Rückgabewert	Bedeutung
>0	Anzahl der Werte im Array
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen



verfügbar ab d.3 Version 5.5.1

Der String „Anton|Berta|Caesar|Dora|Emil“ wird in diesem Beispiel in die fünf Strings „Anton“, „Berta“, „Caesar“, „Dora“ und „Emil“ aufgeteilt. Diese wird ausgegeben. Tritt ein Fehler auf, wird die Fehlernummer ausgegeben.



```
// -----
// Beispiel:
// Zeichenkette aufspalten
// -----

// Variablendeclaration
vars h_ReturnValue
    // Anzahl zurückgegebener Werte
vars h_String = "Anton|Berta|Caesar|Dora|Emil"
    // zu teiler 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: :(api_splitted_string
            [:h_Count])")
    }
}
// Fehlerausgabe
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
        zurückgeliefert!")
}

```

5.18.6 string_cut_leading_char

Diese Funktion entfernt die führenden Zeichen aus einem String. Der konvertierte String wird in der globalen Variable `api_converted_string` gespeichert.



Falls kein Fehler aufgetreten ist, wird die Anzahl der ermittelten Werte durch die Funktion zurückgegeben und NICHT in der Variable `api_single_info` gespeichert. Tritt ein Fehler auf, wird die Fehlernummer ausgegeben.

Parameter	Beschreibung
<code>text</code>	String, dessen führende Zeichen entfernt werden sollen
<code>character</code>	Zeichen, das entfernt werden soll (Standard: Leerzeichen)

Rückgabewert	Bedeutung
<code>>=0</code>	Anzahl entfernter Zeichen
<code>-1</code>	Panther Syntax Fehler, Details siehe d.3 logviewer
<code>-2</code>	Funktion wurde falsch aufgerufen



verfügbar ab d.3 Version 5.5.1

In diesem Beispiel werden von dem String „AAAAAAABCDEFG“ die führenden „A“s entfernt. Der neue String – „BCDEFG“ – wird durch die Funktion in der globalen Variable `apiConvertedString` gespeichert und ausgegeben.



```
// -----
// Beispiel:
// Entfernen führender Zeichen
// -----

// Variablendeclaration
vars h_ReturnValue          // Fehlercode
vars h_Text = "AAAAAAABCDEFG" // 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: :apiConvertedString")
    call api_log_info("Es wurden :h_ReturnValue Zeichen entfernt. ")
}
// Fehlerausgabe
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
                        zurückgeliefert! ")
}
```

5.18.7 string_fill_leading_char

Diese Funktion fügt führende Zeichen in einen String ein, um eine Gesamtstringlänge zu erreichen. Der konvertierte String wird in der globalen Variable `apiConvertedString` gespeichert. Tritt ein Fehler auf, wird die Fehlernummer ausgegeben.

Parameter	Beschreibung
text	String, der aufgefüllt werden soll
character	Zeichen, das eingefügt werden soll (Standard: Leerzeichen)

Parameter	Beschreibung
length	Länge, die der String am Ende haben soll (Standard: 0)

Rückgabewert	Bedeutung
>=0	Anzahl eingefügter Zeichen
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen



verfügbar ab d.3 Version 5.5.1

Dieses Beispiel fügt vor dem Ausgangs-String „12345“ Nullen an, um den neuen String auf eine Gesamtlänge von zehn Zeichen zu bringen. Der neue String und die Anzahl eingefügter Zeichen werden anschließend ausgegeben. Tritt ein Fehler auf, wird die Fehlernummer ausgegeben.



```
// -----
// Beispiel:
// Anfügen von Zeichen
// -----

// Variablen-deklaration
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")
}
// Fehler-ausgabe
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
                       zurückgeliefert! ")
}
```

5.18.8 string_fill_trailing_char

Diese Funktion fügt anhängende Zeichen (Standard: Leerzeichen) in einen String ein, um eine Gesamtstringlänge zu erreichen.

Der konvertierte String wird in der globalen Variable `api_converted_string` gespeichert. Dieser String ist um ein Zeichen „|“ länger als `length`, da Leerzeichen sonst von Panther entfernt werden.

Parameter	Beschreibung
<code>text</code>	String, an den angehängt werden soll
<code>character</code>	Zeichen, das eingefügt werden soll (Standard: Leerzeichen)
<code>length</code>	Länge, die der String am Ende haben soll (Standard: 0)

Rückgabewert	Bedeutung
<code>>=0</code>	Anzahl eingefügter Zeichen
<code>-1</code>	Panther Syntax Fehler, Details siehe d.3 logviewer
<code>-2</code>	Funktion wurde falsch aufgerufen



verfügbar ab d.3 Version 5.5.1

In diesem Beispiel werden an die Zeichenkette „AntonBertaCaesar“ Leerzeichen angehängt, um den String auf eine Länge von 25 Zeichen zu bringen. Zusätzlich wird das 26. Zeichen „|“ angehängt, damit die Leerzeichen nicht automatisch von JPL entfernt werden.

Da beim Funktionsaufruf kein Parameter für das Zeichen, welches eingefügt werden soll, angegeben ist, wird die übergebene Zeichenkette mit Leerzeichen aufgefüllt.

Anschließend werden die neue Zeichenkette und die Anzahl der eingefügten Zeichen ausgegeben. Tritt ein Fehler auf, wird die Fehlernummer ausgegeben.



```
// -----
// Beispiel:
// Anhängen von Zeichen
// -----

// Variablenklärung
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: :apiConvertedString")
        call api_log_info("Es wurden :h_ReturnValue Zeichen angehängt. ")
    }
    // Fehlerausgabe
    else
    {
        call api_log_error("Fehlernummer :h_ReturnValue wurde
            zurückgeliefert! ")
    }
}

```

5.18.9 string_replace

Diese Funktion ersetzt alle Vorkommen einer Zeichenkette in einem String durch eine andere. Den konvertierten String enthält die globale Variable `apiConvertedString`.

Parameter	Beschreibung
<code>text</code>	String, in dem ersetzt werden soll
<code>old_str</code>	String, der ersetzt werden soll (Standard: Leerzeichen)
<code>new_str</code>	neuer String

Rückgabewert	Bedeutung
<code>>=0</code>	Anzahl eingefügter Zeichen
<code>-1</code>	Panther Syntax Fehler, Details siehe d.3 logviewer
<code>-2</code>	Funktion wurde falsch aufgerufen
<code>-107</code>	Text nicht vorgegeben



verfügbar ab d.3 Version 5.5.1

Innerhalb der ursprünglichen Zeichenkette werden in diesem Beispiel alle Vorkommen von „Anton“ durch „Friedrich“ ersetzt. Anschließend werden die neue Zeichenkette und die Anzahl der Austausche ausgegeben. Tritt ein Fehler auf, wird die Fehlernummer ausgegeben.



```

// -----
// Beispiel:
// Ersetzen von Zeichen(ketten) innerhalb von Zeichenketten
// -----

// Variabledeklaration
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! ")
}

```

5.18.10 string_get_position

Diese Funktion liefert die Position einer Zeichenkette in einem String.

Parameter	Beschreibung
text	String, in dem gesucht werden soll
sub_str	String, der gesucht werden soll (Standard: Leerzeichen)
direction	Suchrichtung (1: vorwärts, -1: rückwärts, Standard: 1)
start	Zeichen, ab dem gesucht werden soll (1 = 1. Zeichen) (Standard: 1 für direction=1; @length(text) für direction=-1)

Rückgabewert	Bedeutung
0	nicht gefunden

Rückgabewert	Bedeutung
>0	erste gefundene Position (1. Zeichen=1)
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler; d.velop Support kontaktieren
-107	Text nicht vorgegeben



verfügbar ab d.3 Version 5.5.1

Die ursprüngliche Zeichenkette wird nach „Caesar“ durchsucht. Anschließend wird ausgegeben ob und wo „Caesar“ als erstes gefunden worden ist. Tritt ein Fehler auf, wird die Fehlernummer ausgegeben.



```
// -----
// Beispiel:
// Suchen innerhalb von Zeichenketten
// -----


// Variablendeclaration
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. ")
}
elseif (h_ReturnValue == 0)
{
    call api_log_info("Die Zeichenkette wurde NICHT gefunden. ")
}
```

```
// Fehlerausgabe
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
                       zurückgeliefert! ")
}
```

5.18.11 string_mask_special_char

Diese Funktion maskiert die JAM-Sonderzeichen eines Strings.

Der maskierte String wird in die globale Variable `api_converted_string` geschrieben.

Parameter	Beschreibung
<code>text</code>	String, der maskiert werden soll

Rückgabewert	Bedeutung
0	nicht gefunden
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen



verfügbar ab d.3 Version 6.1.1

5.18.12 api_str_sub



verfügbar ab d.3 Version 7.0.1



Bei den hier vorgestellten Funktionen handelt es sich um Inline-Funktionen, die NICHT über

`call api_function()` gekapselt werden müssen, sondern direkt das Ergebnis zurückliefern.

Äquivalent zu nativen JPL-Funktionen wird im Fehlerfall '-1' zurückgegeben.

Liefert einen Teilstring der übergebenen Zeichenfolge. Im Gegensatz zu dem Substring-Operator von JPL behandelt diese Funktion auf Unicode-Systemen Multi-Byte-Zeichen richtig. Es werden also keine Zeichen

durchtrennt oder mehrmals gezählt.

Diese Funktion erwartet auf Unicode-Systemen UTF8-Zeichenfolgen und auf ANSI-Systemen ANSI-Zeichenfolgen, somit ist diese Funktion in portalem Code nutzbar.

Parameter	Beschreibung
inString	die Ursprungs-Zeichenfolge
startIdx	der Index des ersten Zeichens, das in der Ausgabe auftauchen soll (beginnend bei Index 1) negative Zahlen referenzieren Positionen relativ zum Ende des Strings (-1 ist das letzte Zeichen)
charCount	Anzahl der Zeichen, die in die Ausgabe übernommen werden sollen wird dieser Parameter weggelassen, werden alle Zeichen vom startIdx bis zum Ende zurückgegeben; auch hier referenzieren negative Zahlen Positionen relativ zum Ende des Strings

Rückgabewert	Bedeutung
	der angeforderte Teilstring
-1	Panther Syntax Fehler, Details siehe d.3 logviewer

5.18.13 api_str_sub_utf8



verfügbar ab d.3 Version 7.0.1

Diese Funktion verhält sich so wie `api_str_sub()` mit der Ausnahme, dass auch auf ANSI-Systemen Zeichenfolgen in UTF8 als Eingabe erwartet werden.

5.18.14 api_str_sub_ansi



verfügbar ab d.3 Version 7.0.1

Diese Funktion verhält sich so wie `api_str_sub()` mit der Ausnahme, dass auch auf Unicode-Systemen Zeichenfolgen in ANSI als Eingabe erwartet werden.

5.18.15 api_str_len



verfügbar ab d.3 Version 7.0.1



Bei den hier vorgestellten Funktionen handelt es sich um Inline-Funktionen, die NICHT über `call api_function()` gekapselt werden müssen, sondern direkt das Ergebnis zurückliefern.
Äquivalent zu nativen JPL-Funktionen wird im Fehlerfall '-1' zurückgegeben.

Liefert die Anzahl an Zeichen in einem gegebenem String zurück.
Auf Unicode-Systemen kann die Zeichen-Anzahl geringer als die Byte-Anzahl sein, es gilt
`api_str_len(X) <= @length(X)`
Diese Funktion erwartet auf Unicode-Systemen UTF8-Zeichenfolgen und auf ANSI-Systemen ANSI-Zeichenfolgen, somit ist diese Funktion in portablem Code nutzbar.

Parameter	Beschreibung
<code>inString</code>	die Zeichenfolge

Rückgabewert	Bedeutung
>0	die Anzahl der Zeichen im übergebenen String.
-1	Panther Syntax Fehler, Details siehe d.3 logviewer

5.18.16 `api_str_len_utf8`



verfügbar ab d.3 Version 7.0.1

Diese Funktion verhält sich so wie `api_str_len()` mit der Ausnahme, dass auch auf ANSI-Systemen Zeichenfolgen in UTF8 als Eingabe erwartet werden.

5.18.17 `api_str_len_ansi`



verfügbar ab d.3 Version 7.0.1

Diese Funktion verhält sich wie `api_str_len()` mit der Ausnahme, dass auch auf Unicode-Systemen Zeichenfolgen in ANSI als Eingabe erwartet werden.

5.18.18 api_str_to_utf8



verfügbar ab d.3 Version 7.0.1



Bei den hier vorgestellten Funktionen handelt es sich um Inline-Funktionen, die NICHT über `call_api_function()` gekapselt werden müssen, sondern direkt das Ergebnis zurückliefern.

Äquivalent zu nativen JPL-Funktionen wird im Fehlerfall '-1' zurückgegeben.

Konvertiert eine Zeichenfolge in das UTF8-Format.

Parameter	Beschreibung
<code>ansiInString</code>	Zeichenfolge die nach UTF8 konvertiert werden soll.
<code>numCodepage</code>	Nummerische ID der Ursprungs-Codepage (1250-1258). Wird dieser Parameter nicht angegeben, wird die <code>DEFAULT_CODEPAGE</code> angenommen.

Rückgabewert	Bedeutung
	Die UTF8-Repräsentation der übergebenen Zeichenfolge.
-1	Panther Syntax Fehler, Details siehe d.3 logviewer

5.18.19 api_str_from_utf8



verfügbar ab d.3 Version 7.0.1



Bei den hier vorgestellten Funktionen handelt es sich um Inline-Funktionen, die NICHT über `call_api_function()` gekapselt werden müssen, sondern direkt das Ergebnis zurückliefern.

Äquivalent zu nativen JPL-Funktionen wird im Fehlerfall '-1' zurückgegeben.

Konvertiert eine UTF8-Zeichenfolge nach ANSI.

Parameter	Beschreibung
utf8InString	Zeichenfolge im UTF8-Format.
numCodepage	Nummerische Id der Ziel-Codepage (1250-1258); wird dieser Parameter nicht angegeben, wird die DEFAULT_CODEPAGE angenommen

Rückgabewert	Bedeutung
	Die ANSI-Präsentation der übergebenen Zeichenfolge.
-1	Panther Syntax Fehler, Details siehe d.3 logviewer

5.18.20 api_str_is_utf8



verfügbar ab d.3 Version 7.0.1



Bei den hier vorgestellten Funktionen handelt es sich um Inline-Funktionen, die NICHT über `call api_function()` gekapselt werden müssen, sondern direkt das Ergebnis zurückliefern.

Äquivalent zu nativen JPL-Funktionen wird im Fehlerfall '-1' zurückgegeben.

Gibt an, ob eine Zeichenfolge im UTF8-Format gültig ist.

Parameter	Beschreibung
inString	Zeichenfolge, die überprüft werden soll

Rückgabewert	Bedeutung
1	Zeichenfolge ist im UTF8-Format gültig
0	Zeichenfolge ist nicht im UTF8-Format gültig

5.18.21 api_str_is_ascii



verfügbar ab d.3 Version 7.0.1



Bei den hier vorgestellten Funktionen handelt es sich um Inline-Funktionen, die NICHT über `call api_function()` gekapselt werden müssen, sondern direkt das Ergebnis zurückliefern.
Äquivalent zu nativen JPL-Funktionen wird im Fehlerfall '-1' zurückgegeben.

Gibt an, ob eine Zeichenfolge ausschließlich ASCII-Zeichen enthält (Bytewerte zwischen 1 und 127). Zeichenfolgen die nur ASCII-Zeichen enthalten, können ohne Konvertierung als ANSI- und UTF8-Strings verwendet werden.

Parameter	Beschreibung
inString	die Zeichenfolge, die überprüft werden soll

Rückgabewert	Bedeutung
1	Zeichenfolge enthält nur ASCII-Zeichen
0	Zeichenfolge enthält nicht nur ASCII-Zeichen

5.18.22 api_str_cut_fract_utf8



verfügbar ab d.3 Version 7.0.1



Bei den hier vorgestellten Funktionen handelt es sich um Inline-Funktionen, die NICHT über `call api_function()` gekapselt werden müssen, sondern direkt das Ergebnis zurückliefern.
Äquivalent zu nativen JPL-Funktionen wird im Fehlerfall '-1' zurückgegeben.

Diese Funktion entfernt ein durchtrenntes Multi-Byte-Zeichen vom Ende einer UTF8-kodierten Zeichenfolge. Diese Funktion kann genutzt werden um Werte, die von einem Fremdsystem geliefert wurden und möglicherweise undefiniert abgeschnitten wurden in gültige UTF8-Werte zurückzuführen.

Parameter	Beschreibung
inString	Ursprungs-Zeichenfolge, diese muss gültiges UTF-8 darstellen mit Ausnahme von möglicherweise einem durchtrennten Zeichen am Ende

Rückgabewert	Bedeutung
	Befindet sich kein durchtrenntes Zeichen am Ende der Eingabe, wird diese unverändert zurückgegeben. Die Zeichenfolge in der Rückgabe dieser Funktion kann bis zu drei Bytes kürzer sein als die Eingabe.
-1	Panther Syntax Fehler, Details siehe d.3 logviewer

5.18.23 api_str_hash



verfügbar ab d.3 Version 7.1.1



Bei den hier vorgestellten Funktionen handelt es sich um Inline-Funktionen, die NICHT über `call api_function()` gekapselt werden müssen, sondern direkt das Ergebnis zurückliefern.

Äquivalent zu nativen JPL-Funktionen wird im Fehlerfall '-1' zurückgegeben.

Diese Funktion erstellt Hash-Werte auf Strings.

Parameter	Beschreibung
hash_algorithm	<p>Gültige Werte sind:</p> <ul style="list-style-type: none"> • MD5 • SHA-1 • SHA-224 • SHA-256 • SHA-384 • SHA-512 • RIPEMD-128 • RIPEMD-160 • RIPEMD-256 • RIPEMD-320 <div style="border: 2px solid red; padding: 10px; margin-top: 10px;">  <p>MD5 und SHA-1 sollten nicht in sicherheitsrelevanten Szenarien eingesetzt werden.</p> </div>
value	Die Zeichenfolge für die ein Hash-Wert gebildet werden soll.
salt_level (0)	



Parameter	Beschreibung
	<p>Mit diesem Parameter kann angegeben werden, dass d3 automatisch ein Salting des Hashes vornimmt.</p> <p>0 Ist dies gesetzt, wird in den Hash-Wert nur die übergebene Zeichenfolge eingerechnet. In diesem Modus ist das Ergebnis mit der Hash-Berechnung in anderen Anwendungen kompatibel.</p> <p>1 Ein in d3 fest eingebauter Salt wird in den Hash-Wert eingerechnet. Alle d3 Installationen erstellen bei diesem Modus den gleichen Hash-Wert für die gleiche übergebene Zeichenfolge (<code>value</code>).</p> <p>2 Ein Salt, der für die jeweilige d3 Installation spezifisch ist, wird in den Hash-Wert eingerechnet. D.h. zwei d3 Installationen liefern in diesem Modus nur dann das gleiche Ergebnis, wenn die Datenbank der einen Installation für die andere Installation kopiert wurde und die gleiche Archiv-Kennung gegenüber dem d3 gateway verwendet wird.</p>



Um eine eigene Salting-Methode zu realisieren (beispielsweise zum Einrechnen des Benutzernamens in einen Passwort-Hash), kann der eigene Salt direkt an den `value`-Parameter angehängen werden. Diese Technik lässt sich selbstverständlich auch mit den d3 Salt-Modi kombinieren.



Zum Thema Salting:

http://de.wikipedia.org/wiki/Salt_%28Kryptologie%29

Rückgabewert	Bedeutung
	Im Erfolgsfall wird der jeweilige Hash-Wert in hexadezimaler Kodierung (0–9 und a–f) ohne Trennzeichen zurückgegeben.
-1	Panther Syntax Fehler, Details siehe d3 logviewer Beispielsweise der Übergabe einer ungültigen Algorithmus-Kennung

5.19 Diverses

5.19.1 restore_from_jukebox

Die Funktion lagert die Dateien von der Jukebox wieder ein.

Parameter	Beschreibung
status	Status der Version, die wieder eingelagert werden soll {"Freigabe", "Archiv"}
archiv_index	Index der Version, die wieder eingelagert werden soll
doc_id	ID des betroffenen Dokumentes; wird nichts angegeben, wird die ID des Kontexts benutzt
user_name	Benutzer, in dessen Namen gearbeitet werden soll; wird nichts angegeben, wird der ausführende Benutzer verwendet

Rückgabewert	Bedeutung
0	alles OK
>0	Datenbankfehler
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler, d.velop support kontaktieren
-101	weder Dokumenten-ID übergegeben, noch im Kontext gesetzt
-126	keinen Status vorgegeben
-201	ungültigen Benutzernamen vorgegeben
-223	ungültigen Status; "Freigabe" oder "Archiv" erwartet
-301	ungültige Dokument ID vorgegeben oder Benutzer hat keine Rechte darauf
-302	Fehler beim Ermitteln der Benutzerberechtigungen
-303	Benutzer hat kein Zugriffsrecht auf das Dokument api_error_var[...] = "doc_permission" api_error_value[...] = "<Berechtigung>"
-310	Benutzer hat keine Recht alte Archivversionen zu sehen api_error_var[...] = "doc_permission" api_error_value[...] = "<Berechtigung>"
-506	es liegt keine Freigabe-Version vor api_error_var[...] = "doc_permission" api_error_value[...] = "<Berechtigung>"
-531	es liegt keine Version im Status Archiv vor api_error_var[...] = "doc_permission" api_error_value[...] = "<Berechtigung>"
-532	Lokalisation des Dokumentes <\:doc_id> im Status <\:status> kann nicht bestimmt werden

Rückgabewert	Bedeutung
-533	Dokument ist nicht ausschließlich auf der Jukebox, sondern auch auf Festplatte; eine Wiedereinlagerung ist nicht erforderlich api_error_var[...] = "lokalisierung" api_error_value[...] = "<Lokalisierung>"
-534	die Jukebox hat die Datei nicht geliefert
-556	Fehler beim Kopieren der Datei



verfügbar ab d.3 Version 6.0

5.19.2 restore_from_history

Die Funktion stellt ein gelöschtes Dokument aus der History wieder her.

Parameter	Beschreibung
aktion_id	ID der Aktion, die das Dokument gelöscht hat; wird nichts angegeben, wird die höchste Aktionsnummer für dieses Dokument verwendet
doc_id	ID des betroffenen Dokumentes; wird nichts angegeben, wird die ID im Kontext benutzt
user_name	Benutzer, in dessen Namen gearbeitet werden soll; wird nichts angegeben, wird der ausführende Benutzer verwendet

Rückgabewert	Bedeutung
0	alles OK
>0	Datenbankfehler
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-101	weder Dokument ID übergeben, noch im Kontext gesetzt
-535	keinen Historieneintrag gefunden für dieses Dokument und diese Aktion ID
-536	Aktion war kein Löschen; Dokument kann nicht wiederhergestellt werden
-537	Fehler beim Kopieren eines Eintrages von einer Historientabelle in eine Permanente
-538	Fehler beim Löschen eines Eintrages aus der Historientabelle
-539	Fehler beim Schreiben der Dokument Flags
<-9500	Fehler in Hook-Funktion hook_insert_exit_d3_20



verfügbar ab d.3 Version 5.5.1 Hotfix 11, 6.0

5.19.3 send_email

Die Funktion ermöglicht das Versenden einer E-Mail.



Versand mit Adressarrays (s.u.) funktioniert nicht bei zukünftigem Datum (Parameter date)

Parameter	Beschreibung
recipient	Empfänger der E-Mail (d.3 Benutzername) oder (ab 6.2) d.3 Gruppenname
notice	Betrefftext
date	Datum des Sendens (zukünftig möglich) optional
doc_id	ID des zu versendenden Dokumentes (als Link) optional wird nichts angegeben, wird die ID des Kontextes benutzt
user_name	Absender (d.3 Benutzername); wird nichts angegeben, wird der ausführende Benutzer verwendet
body_file	Name und Pfad einer Datei, die den Body-Text enthält
mail_format	html: HTML-Format, sonst: Text-Format (default)
attach	Dokument als Anhang mitsenden (0 (default) / 1)
doc_status	mitzusendende Version im Status
archive_index	Index der Archivversion, die mitgesendet werden soll
attach_abh	Liste der abhängigen Dateien (z.B. "T1,P1")
attach_file	Liste der anzuhängenden Dateien (z.B. "C:\\\\datei1.jpg,c:\\\\datei2.txt")
use_recip_array	1: statt recipient api_email_recipients als Empfänger nutzen (-> bis zu 100 Empfänger möglich) default: 0
use_cc_array	1: die E-Mail in Kopie senden an die Empfänger in api_email_cc_recipients

Parameter	Beschreibung
	default: 0
use_bcc_array	1: die E-Mail in versteckter Kopie senden an die Empfänger in api_email_bcc_recipients default: 0

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler, d.velop support kontaktieren
-110	kein Empfänger angegeben
-201	ungültigen Benutzernamen vorgegeben
-301	ungültige Dokument ID vorgegeben oder Benutzer hat keine Rechte darauf
-302	Fehler beim Ermitteln der Benutzerberechtigungen
-303	Benutzer hat kein Zugriffsrecht auf das Dokument api_error_var[...] = "doc_permission" api_error_value[...] = "<Berechtigung>"
-540	SMTP_SUPPORT ist nicht aktiviert api_error_var[...] = "smtp_support" api_error_value[...] = "SMTP_SUPPRT"
-541	Die SMTP DLL ist nicht verfügbar api_error_var[...] = "smtp_dll_available" api_error_value[...] = "SMTP_DLL_VERFUEGBAR"
-542	Fehler beim Senden der E-Mail



verfügbar ab d.3 Version 5.5.1 Hotfix 11, 6.0

In dem Beispiel wird eine Email von dem Benutzer „dvelop“ an „user1“ geschickt. Die Email enthält html-formatierten Text und einen Link auf ein Dokument aus dem d.3-Archiv. Tritt ein Fehler auf, wird die Fehlernummer ausgegeben.



```
// -----
// Beispiel:
// Versenden einer Email
// -----



// Variablen Deklaration
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 versendenes
                                // 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!")
    }
}

```

5.19.4 lock_token_acquire

Legt eine Sperre für ein Objekt in der DB ab oder verlängert diese.

Parameter	Beschreibung
object_id	50 Zeichen langer Name des LockTokens. Dieser Name kann frei vergeben werden und dient dazu, die object_id's voneinander eindeutig zu trennen. z.B. REDLINE_SYNC
object_name	200 Zeichen langer Schlüssel, passend zum object_name. Der Inhalt von object_id wird definiert und gibt an, was synchronisiert werden soll. z.B. <doku_id>, <status>, <archivindex> Der Server interpretiert diesen Wert nicht.
token	Wenn ein Lock verlängert werden soll, gibt man hier das bereits vergebene Token an.
object_info	255 Zeichen Informationstext, den der Aufrufer zum Objekt ablegen kann. Dieser wird zurückgeliefert, wenn ein weiterer Aufruf dasselbe Objekt sperren möchte.
ttl	Hier wird ein gewünschter Time to Life Wert in Minuten angegeben. Ist dieser Wert nicht vorhanden, wird ein Default von fünf Minuten angenommen.
user_name	User, der als Halter der Sperre in der DB eingetragen werden soll

Nach dem erfolgreichen Aufruf der Funktion sind folgende Globale Variablen gefüllt:

Globale Variablen	
api_lock_token (P)	Wurde erfolgreich eine neue Sperre angelegt, wird das zugehörige Token zurückgeliefert; damit kann die Sperre verlängert oder aufgehoben werden
api_lock_ttl (P)	Tatsächliche Gültigkeitsdauer der Sperre in Minuten
api_lock_object_info (O)	Schlug das Anlegen der Sperre fehl, da das Objekt schon anderweitig gesperrt ist, wird hier der mit der Sperre angelegte Info-Text zurückgeliefert
api_lock_user (O)	Schlug das Anlegen der Sperre fehl, da das Objekt schon anderweitig gesperrt ist, wird hier der User zurückgeliefert, der die Sperre angelegt hat

Rückgabewert	Bedeutung
0	alles OK (<code>api_lock_token</code> enthält das vergebene token, <code>api_lock_ttl</code> die eingetragene Dauer)
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler, d.velop support kontaktieren
-130	keine <code>object_id</code> übergeben
-131	kein <code>object_name</code> übergeben
-255	ungültiges Token übergeben
-600	Fehler beim Löschen eines alten Tokens. Näheres im Logviewer
-601	Objekt bereits gesperrt (<code>api_lock_user</code> und <code>api_lock_info</code> enthalten Näheres dazu)
-602	Fehler beim Ermitteln der Lebensdauer des Tokens
-603	Fehler beim Einfügen der Lebensdauer in die DB
-604	Fehler beim Einfügen des Tokens in die DB

5.19.5 lock_token_release

Gibt eine Sperre für ein Objekt in der DB frei.

Parameter	Beschreibung
<code>object_id</code>	50 Zeichen langer Name des LockTokens. Dieser Name kann frei vergeben werden und dient dazu, die <code>object_id</code> 's voneinander eindeutig zu identifizieren. z.B. REDLINE_SYNC
<code>object_name</code>	200 Zeichen langer Schlüssel, passend zum <code>object_name</code> . Der Inhalt von <code>object_id</code> wird definiert und gibt an, was synchronisiert werden soll. z.B. <doku_id>, <status>, <archivindex> Der Server interpretiert diesen Wert nicht.
<code>token</code>	Token, das zuvor mit <code>lock_token_acquire</code> angelegt wurde und jetzt freigegeben werden soll

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler, d.velop support kontaktieren

Rückgabewert	Bedeutung
-130	keine object_id übergeben
-131	kein object_name übergeben
-132	kein token übergeben
-605	Fehler beim Löschen des Tokens. (DB-Fehler oder ungültige Kombination)

5.19.6 start_ext_program

Führt einen Kommandozeilenaufruf durch (Start eines externen Programms)
z. B. notepad.exe C:\\\\test.txt c:\\\\test1.txt



Bei erfolgreichem Aufruf enthält api_single_info den Rückgabewert des Kommandos.

Parameter	Beschreibung
program	auszuführendes Programm (z.B. notepad.exe) Übergabeparameter durch Leerzeichen getrennt z.B. C:\\\\test.txt c\\\\\\test1.txt
hidden	0: Programm wird mit sichtbarem Fenster gestartet (default) 1: Programm wird versteckt gestartet

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
138	kein Programm angegeben
212	ungültige DokumentID vorgegeben
226	ungültiger Wert für publish, muss 0 oder 1 sein
614	Dokumentart ist nicht für Web-Veröffentlichung vorgesehen
615	nur Dokumente mit einer Version im Status Freigabe können veröffentlicht werden
616	Fehler beim Aufruf des Programms



verfügbar ab d.3 Version 6.2

Führt einen Kommandozeilenaufruf durch (Start eines externen Programms)



```
// -----
// Beispiel:
// Externes Programm starten
// -----

// Variablendeclaration
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!")
}
```

5.19.7 get_current_datetime

Liefert den aktuellen Zeitstempel im gewünschten Format zurück.

Parameter	Beschreibung
format	Formatstring, bestehend aus festen Zeichen und Platzhaltern
hr12	0: 24-Stunden-Uhrzeit (default) 1: 12-Stunden-Uhrzeit

Datumsformat	String	Beschreibung	Beispiel
Jahreszahl	%4y	vierstellig	2007
	%2y	2-stellig	07
Monat	%m	1/2-stellig	[1..12]
	%0m	2-stellig	[01..12]
	%*m	ganzer Name	February
	%3m	Kurzname	Feb

Datumsformat	String	Beschreibung	Beispiel
Tag	%d	1/2-stellig	[1..12]
	%0d	2-stellig	[01..31]
Wochentag	%*d	ganzer Name	Wednesday
	%3d	Kurzname	Wed
Jahrestag	% .d	numerisch	[1..7]
	%+d	numerisch	[1..366]
Stunde	%h	1/2-stellig	[1..12] bzw. [0..23]
	%0h	2-stellig	[01..12] bzw. [00..23]
Minute	%M	1/2-stellig	[0..59]
	%0M	2-stellig	[00..59]
Sekunde	%s	1/2-stellig	[0..59]
	%0s	2-stellig	[00..59]
Millisekunde	%S	1-3-stellig	[0..999]
	%0S	3-stellig	[000..999]
AM/PM	%p	Zeitformat	am/pm



für % im Ausgabestring setzen Sie %% im Formatstring

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-243	ungültiges Format



verfügbar ab d.3 Version 6.2

Aktuellen Zeitstempel ausgeben.



```
// -----
// Beispiel:
// Aktuellen Zeitstempel ausgeben
// -----
// Variablendeclaration
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!")
}
```

Datum im Format “Heute ist der 1.9.2009 13:47:07 Uhr” ausgeben.



```
// -----
// Beispiel:
// Aktuellen Zeitstempel ausgeben
// -----
// Variablendeclaration
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! ")
}

```

Datum im Format "Heute ist der 1.9.2009 1:47:07 pm" ausgeben.



```

// -----
// Beispiel:
// Aktuellen Zeitstempel ausgeben
// -----
// Variablendeclaration
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!")
}

```

5.19.8 sleep

Diese Funktion "schläft" für eine Anzahl an Millisekunden ohne Prozessorauslastung.

Parameter	Beschreibung
milli_seconds	Wert zwischen 1 und 5000; andere Werte werden ignoriert

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen

5.19.9 process_get_id

Gibt den Wert der Prozess ID in der Globalen Variable `api_single_info` (hexadecimal) zurück.
Kann z.B. hilfreich sein, wenn von einer Hook Funktion kundenspezifische Logs geschrieben werden.

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler, d.velop support kontaktieren

5.19.10 regular_expression_test

Testet einen Wert gegen einen regulären Ausdruck.

regular_expression_test ruft die interne Funktion auf. Letztlich wird von Panther auf Gültigkeit geprüft, siehe http://www.prolifcs.com/docs/panther/html/eds_html/entry.htm#881493

Parameter	Beschreibung
regular_expression	der reguläre Ausdruck, gegen den getestet werden soll
test_value	der Wert, der getestet werden soll

Rückgabewert	Bedeutung
0	alles OK (Test-Ergebnis steht in api_single_info)
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-127	kein regulärer Ausdruck übergeben
-128	kein Testwert übergeben



verfügbar ab d.3 Version 6.2

Im Beispiel wird der reguläre Ausdruck `[0-9]{3,6}` getestet. Dieser ist gültig für alle Zahlen mit drei bis sechs Stellen.

Beispiel 1; regulärer Ausdruck gültig



```
// -----
// Beispiel 1:
// regulären Ausdruck testen
// „api_single_info“ wird auf 1 gesetzt (für gültig)
// -----
// Variablenklärung
vars h_ReturnValue                      // Fehlercode
vars h_Format = "[0-9]\{3,6\}" , "12345" // 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!")
}
```

Beispiel 2; regulärer Ausdruck ungültig



```
// -----
// Beispiel 2:
// regulären Ausdruck testen
// „api_single_info“ wird auf 0 gesetzt
// (für nicht gültig, weil ein Buchstabe enthalten ist).
// -----
// Variablenklärung
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!")
}

```

Beispiel 3; regulärer Ausdruck ungültig



```

// -----
// Beispiel 3:
// regulären Ausdruck testen
// „api_single_info“ wird auf 0 gesetzt
// (für nicht gültig, weil nur 2 Zeichen)
// -----
// Variablendeclaration
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!")
}

```

5.19.11 object_property_set

Setzt Eigenschaftswerte für Objekte. Durch das Setzen eines Leerstrings für value löscht man den Eigenschaftswert für das Objekt.

Parameter	Beschreibung
property_name (P)	interner Name der Objekteigenschaft
object_id (P)	Objektname (z.B. Benutzername, Gruppenname abhängig von der Objekt-Klassen-ID)
object_class_id (P)	Objekt-Klassen-ID 1: Benutzer 2: Gruppen 3: Organisationseinheiten 4: Tätigkeitsprofile 5: Wertemengen 6: Repositoryfelder

Parameter	Beschreibung
	7: Dokumentarten 8: Dokumentklassen 9: Berechtigungsprofile 10: Mengen 11: Workflows
property_value	Eigenschaftswert

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler, d.velop support kontaktieren
-140	kein Eigenschaftsname übergeben
-141	keine Objekt-ID übergeben
-142	keine Objekt-Klassen-ID übergeben
-240	ungültige Objekt-Klassen-ID übergeben
-241	unbekannten Eigenschaftsnamen übergeben



verfügbar ab d.3 Version 6.2

5.19.12 object_properties_get

Ermittelt die Eigenschaftswerte zu einer Eigenschaft oder zu einem Objekt. Werden beide Importparameter gesetzt, wird genau der Wert der Objekt-Eigenschaft name zum Objekt object_id zurückgegeben.

Die Anzahl der zurückgelieferten Ergebnisse wird in api_single_info geschrieben.

Folgende Arrays des Kontexts werden gefüllt:

Variable	Beschreibung
api_property_name	Eigenschaftsname
api_property_caption	Kurzbeschreibung der Eigenschaft
api_property_description	ausführliche Beschreibung der Eigenschaft
api_property_value	Wert der Eigenschaft
api_object_id	Objekt-ID der Eigenschaft (z.B. Benutzername, Gruppenname, ...)

Variable	Beschreibung
api_doc_status_info	Zusatzinfo zum Status (0/1)=(ungeprüft/geprüft, gesperrt/freigegeben)
api_doc_editor	Bearbeiter

Parameter	Beschreibung
object_id (O/P)	Objektname (z.B. Benutzername, Gruppenname abhängig von der Objekt-Klassen-ID)
object_class_id (O/P)	Objekt-Klassen-ID 1: Benutzer 2: Gruppen 3: Organisationseinheiten 4: Tätigkeitsprofile 5: Wertemengen 6: Repositoryfelder 7: Dokumentarten 8: Dokumentklassen 9: Berechtigungsprofile 10: Mengen 11: Workflows
property_name (O/P)	interner Name der Objekteigenschaft
language (O)	Wird hier die Sprache übergeben, werden <code>caption</code> und <code>description</code> in der entsprechenden Sprache gefüllt (falls vorhanden), sonst in der ersten vorhandenen Sprache.

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler, d.velop support kontaktieren
-143	weder Eigenschaftsname, noch Objekt-ID und Objekt-Klassen-ID übergeben
-240	ungültige Objekt-Klassen-ID übergeben
-241	unbekannten Eigenschaftsnamen übergeben



verfügbar ab d.3 Version 6.2

5.19.13 set_signature_data



wird ab Version 7.1 nicht mehr unterstützt
Bei Aufruf kommt Fehler -671.



verfügbar ab d.3 Version 7.0.1

Schreibt zu einem Dokument die Signaturinformationen in die Datenbank.

Parameter	Beschreibung
doc_id	Doku-ID des Dokuments
doc_status	Status des Dokuments (B, P, F, A)
archive_index	zugehöriger Archiv-Index

Über die globalen Kontextvariablen werden vor dem Aufruf die Signaturinformationen gesetzt:

Variable	Beschreibung
api_sig_docextension	Datei-Erweiterung des abhängigen Dokumentes, für das die Signatur gilt
api_sig_user	Anwender, der die Signatur durchgeführt hat
api_sig_version	welche Anwendung hat die Signatur und die Verifikation ausgeführt
api_sig_tstamp	gibt den Zeitpunkt an, wann der Anwender das Dokument signiert hat
api_sig_algorithm	dieser Wert steht für das verwendete Signaturverfahren
api_sig_digest	OID des HASH Verfahrens
api_sig_keylength	dieser Wert steht für die verwendete Schlüssellänge in Bit
api_sig_reason	Angabe des Benutzers, warum er dieses Dokument signiert hat
api_sig_type	Art der Signatur (leer für fortgeschrittene und qualifizierte Signaturen)
api_sig_reached_verif_level (0)	erreichte Prüftiefe (1-9)
api_sig_requested_verif_level (0)	gewünschte Prüftiefe (1-9)

Variable	Beschreibung
api_sig_content (0)	"a" für attached "d" für detached (Default-Wert, wenn leer) "e" für embedded
api_sig_provider (0)	Provider
api_sig_producer (0)	Hersteller
api_sig_certissuer	dieses Attribut enthält den Namen der Zertifizierungstelle, welche das Zertifikat für die Person ausgestellt hat
api_sig_certsubject	dieses Attribut enthält den Namen der Person, welche das Dokument signiert hat; auf diese Person ist das für die Signatur verwendete Zertifikat ausgestellt
api_sig_certvalidfrom	gibt den Zeitpunkt an, ab wann das Zertifikat der Person gültig ist
api_sig_certvalidto	gibt den Zeitpunkt an, bis wann das Zertifikat der Person gültig ist
api_sig_certserial	das ist die Seriennummer des Zertifikats; die Kombination aus CERTIssuer und CERTSerial ist eindeutig
api_sig_certalgorithm	Zertifikatsalgorithmus
api_sig_certrestriction (0)	textliche Angabe der Beschränkungen, die für das zugehörige Zertifikat gelten



Einschränkungen:

Im Gegensatz zur Client API Funktion `SetSignatureData` unterstützt die Server API Funktion "einfache" Signaturen nur eingeschränkt, da die Zertifikatsinformationen Pflichtparameter sind.

5.20 d3fc-Calls

Ab Version 6.2 ist es möglich, mit der ServerApi, also aus JPL heraus, d3fc-Calls abzusetzen. Gedacht ist dies, um `d.3 search`, `d.ecs rendition service` oder andere Server anzusprechen, die d3fc unterstützen. Es geht aber auch (z.B. zu Testzwecken) mit dem `d.3` Server selbst.



Dabei ist jedoch zu beachten, dass dafür zwei Serverprozesse nötig sind und dass diese beide für die Dauer des Calls für andere Aktivitäten blockiert sind. Ein solcher Einsatz ist daher Produktiv nicht ratsam.

Der Ablauf eines solchen Calls ist analog zum Aufruf mittels `d3remote.ocx` aufgebaut (siehe `d.3` API-Doku (`d3api.pdf`)).



Die d3fc-Calls sollen ausschließlich genutzt werden, um andere Dienste, die über d3fc kommunizieren, anzusprechen, nicht aber den Server selbst.
Mit den d3fc-Calls können keine administrativen Tätigkeiten (wie bspw. User anlegen, in optionale Felder schreiben) getätigt werden.

Die einzelnen Funktionen mit Parameterbeschreibung sind unten aufgeführt.



Vier Beispiele befinden sich am Ende dieses Kapitels.

- 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

5.20.1 d3fc_user_set

Mit dieser Funktion setzen Sie einen ausführenden Benutzer für den d3fc-Call.

Parameter	Beschreibung
user_name	Benutzer, der zu verwenden ist (default: d3_server)

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen

Rückgabewert	Bedeutung
-102	kein Benutzername angegeben

5.20.2 3fc_password_set

Passwort des ausführenden Benutzers für den d3fc-Call setzen.

Parameter	Beschreibung
password	Passwort (default: server_api_client)

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-137	kein Passwort angegeben

5.20.3 d3fc_remote_server_set

Name / IP des Hosts, der angesprochen werden soll.

Parameter	Beschreibung
remote_server	Hostname / IP (default: Inhalt der Config-Variable d3fc_remote_server_name)

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen

5.20.4 d3fc_port_set

Port des Hosts, der angesprochen werden soll.

Parameter	Beschreibung
port	Port (default: Inhalt der Config-Variable d3fc_remote_server_tcpip_port)

Rückgabewert	Bedeutung
0	alles OK

Rückgabewert	Bedeutung
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen

5.20.5 d3fc_server_set

Serverkürzel bei d.3 Prozessen.

Parameter	Beschreibung
server	Serverkürzel (default: Inhalt der Config-Variable d3fc_remote_server_kuerzel)

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen

5.20.6 d3fc_timeout_set

Zeit in Sekunden, die auf eine Antwort des Remoteservers gewartet werden soll.

Parameter	Beschreibung
timeout	Timeout in Sekunden (default: Inhalt der Config-Variable d3fc_remote_server_timeout)

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen

5.20.7 d3fc_function_name_set

Name des API-Calls.



Dieser Wert muss als erstes (vor allen Import-/Export-Parametern und Tabellen/-Dateien) gesetzt werden.

Parameter	Beschreibung
function_name	Name der d.3 Funktion

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-133	kein Funktionsname angegeben

5.20.8 d3fc_importing_set

Importparameter setzen.



Muss für jeden Parameter einmal aufgerufen werden.

Parameter	Beschreibung
param_name	Name des Parameters
param_value	Wert des Parameters

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-133	kein Funktionsname angegeben
-134	kein Parametername angegeben

5.20.9 d3fc_exporting_set

ExportParameter setzen.



Muss für jeden Parameter einmal aufgerufen werden, der nach dem Call abgefragt werden soll.

Parameter	Beschreibung
param_name	Name des Parameters

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-133	kein Funktionsname angegeben
-134	kein Parametername angegeben

5.20.10 d3fc_exporting_get

ExportParameter abfragen nach d3fc_execute.

Abfrage der Werte für Parameter, die vor dem Aufruf von d3fc_execute mit d3fc_exporting_set gesetzt wurden.

Der Wert wird in api_single_info geschrieben.

Parameter	Beschreibung
param_name	Name des Parameters

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-133	kein Funktionsname angegeben
-134	kein Parametername angegeben

5.20.11 d3fc_table_set_import_file

Import-Datei setzen.

Name (inklusive Pfad) der Datei, die der Call als Eingabe bekommt.



Dieser Wert kann nur alternativ zur Import-Tabelle gesetzt werden, nicht gleichzeitig.

Parameter	Beschreibung
filename	Dateiname

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-150	kein Dateiname angegeben
-250	Datei existiert nicht
-606	Import-Tabelle ist bereits gesetzt

5.20.12 d3fc_table_set_export_file

Export-Datei setzen.

Name (incl. Pfad) der Datei, in die die Ausgabe des Calls geschrieben werden soll.



Dieser Wert kann nur alternativ zur Export-Tabelle gesetzt werden, nicht gleichzeitig.

Parameter	Beschreibung
filename	Dateiname

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-150	kein Dateiname angegeben
-584	Datei existiert schon
-611	Export-Tabelle ist bereits gesetzt

5.20.13 d3fc_table_set_headline

Tabellenkopf setzen.



Dieser Wert kann nur alternativ zur Import/Export-Datei gesetzt werden, nicht gleichzeitig.

Parameter	Beschreibung
headline	Form: <code><head1><Leerzeichen><head2><Leerzeichen>...<headN><Leerzeichen><Punkt></code>



`doc_id doc_status` Platzhalter der Felder durch Leerzeichen getrennt. Am Ende folgen mindestens ein Leerzeichen (" ") und ein Punkt ("."). Die Platzhalter bestehen aus dem Feldnamen und sovielen Leerzeichen, dass die Feldinhalte hineinpassen. Die DokuID kann bis zu zehn Zeichen lang sein, also muss der Platzhalter mindestens zehn Zeichen lang sein. Der Name `doc_id` hat sechs Zeichen, also noch vier Leerzeichen anhängen (+ das trennende Leerzeichen): Der Status hat zwei Zeichen.

Der Name `doc_status` hat schon zehn Zeichen, also muss kein weiteres Leerzeichen angehängt werden

(außer dem trennenden Leerzeichen):

`1234567890_12_. doc_id doc_status . head_i_o "I"=Kopf der Import-Tabelle (Input) "O"=Kopf der Export-Tabelle (Output)`

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-135	keine Headline angegeben
-242	ungültiger Wert für <code>head_i_o</code> angegeben (muss I oder O sein)
-607	Import-Datei ist bereits gesetzt
-611	Export-Datei ist bereits gesetzt

5.20.14 d3fc_table_add

Tabellenzeile zur Import-Tabelle hinzufügen.



Die Headline muss bereits gesetzt sein.

Parameter	Beschreibung
n.a.	

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-135	Headline ist noch nicht gesetzt

5.20.15 d3fc_field_set

Feld in der aktuellen Tabellenzeile der Importtabelle setzen.



Die Headline muss bereits gesetzt sein.
d3fc_table_add muss bereits aufgerufen worden sein.

Parameter	Beschreibung
field_name	Name der Tabellenspalte in Headline
field_value	Wert des Feldes

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-135	Headline ist noch nicht gesetzt
-136	kein field_name angegeben
-613	Feld in Tabelle nicht vorhanden

5.20.16 d3fc_field_get

Feld in der aktuellen Tabellenzeile der Export-Tabelle lesen.



Feld muss vor d3fc_execute in der Headline gesetzt worden sein.
(d3fc_table_headline_set)
d3fc_first() / d3fc_next() muss bereits aufgerufen worden sein.

Parameter	Beschreibung
field_name	Name der Tabellenspalte in Headline

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-136	kein field_name angegeben
-612	Feld in Tabelle nicht vorhanden

5.20.17 d3fc_first

Erste Tabellenzeile der Export-Tabelle laden.

Parameter	Beschreibung
n.a.	

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-609	Tabellenende erreicht (Konstante EOT = -609)

5.20.18 d3fc_next

Nächste Tabellenzeile der Export-Tabelle laden

Parameter	Beschreibung
n.a.	

Rückgabewert	Bedeutung
0	alles OK
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-609	Tabellenende erreicht (Konstante EOT = -609)

5.20.19 d3fc_execute

Einen API-Call ausführen.



Rückgabewert (z.B. 0002 - falscher Benutzername/Passwort) wird in `api_single_info` geschrieben

Parameter	Beschreibung
n.a.	

Rückgabewert	Bedeutung
46	Fehler ungültige Proxy-Funktion beim Empfangen der Daten
45	Fehler kein Proxy-Support beim Empfangen der Daten
25	Fehler beim Senden der Datei <code>api_d3fc_import_file</code>
24	Fehler beim Empfangen der Daten: die gesendeten Daten haben die Länge 0
22	Fehler beim Empfangen der Daten: es wurde nichts gesendet
21	Fehler beim Senden der Datei <code>api_d3fc_import_file</code>
20	Fehler beim Öffnen der Zielfile für die empfangenen Daten
0	alles OK
-1	Fehler bei <code>fcCreate</code>
-2	Fehler bei <code>fcGetHeader</code>
-3	Fehler bei <code>fcConnect</code>
-4	Fehler bei <code>fcSend</code>
-10	Timeout bei <code>fcReceive</code> (keine Antwort innerhalb des definierten Timeout-Zeitraumes)
-15	Fehler vor dem Senden der Datei <code>api_d3fc_import_file</code> : Datei konnte nicht geöffnet werden oder nicht genügend Arbeitsspeicher
-21	Fehler beim Öffnen der übertragenen Datei
-22	Fehler beim Ablegen der übertragenen Datei unter dem Zielfilename (<code>api_d3fc_export_file</code>)
-609	Tabellenende erreicht (Konstante EOT = -609)



5.20.20 Beispiele

- ValidatePasswordForUser
- SearchDocumentInIDList
- ReceiveNote
- SendNote

Beispiel 1 - ValidatePasswordForUser



```
// -----
// Beispiel 1:
// ValidatePasswordForUser
// -----


// Variablen-deklaration
vars h_ReturnValue           // Fehlercode
vars h_User = "armin"         // Benutzer für d3fc-Aufruf
vars h_Password = "frikadelle" // Passwort für d3fc-Aufruf


// Verbindungs-optionen
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!")
}
```

Beispiel 2 - SearchDocumentInIDList



```
// -----
// Beispiel 2:
// SearchDocumentInIDList
// -----



// Variablen-deklaration
vars h_ReturnValue           // Fehlercode
vars h_counter = 0            // Schleifen-zähler
vars h_DokID = "Y000005174"   // zu suchendes Dokument


// Verbindungs-optionen
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


// Funktions-name
call api_function ("d3fc_function_name_set", "SearchDocumentInIDList")

    // Funktions-name


// 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


// Export-Parameter
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!")
}
```

Beispiel 3 - ReceiveNote



```
// -----
// Beispiel 3:
// ReceiveNote
// -----



// Variablendeclaration
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!")
}
```

Beispiel 4 - SendNote



```
// Beispiel 4:  
// SendNote  
// -----  
// Variablendeclaration  
vars h_ReturnValue                      // Fehlercode  
vars h_DokID = "Y000005177"                // DokumentID  
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!")  
}
```

5.21 JAVA

- `java_add_param`
- `java_clear_param`
- `java_call_static`

5.21.1 `java_add_param`

Diese Funktion fügt einen neuen Parameter für den Aufruf der nächsten Java-Funktion hinzu.

Die aufgerufene Java-Funktion erhält alle hinzugefügten Parameter als String-Array.

Die Parameter befinden sich in der Reihenfolge in dem String-Array, in der sie über diese Funktion hinzugefügt wurden.



verfügbar ab d.3 Version 7.2

Parameter	Beschreibung
<code>param_value</code>	dieser String wird als Parameter hinzugefügt

Rückgabewert	Bedeutung
0	alles OK
1	Java-Support nicht aktiviert / Java-VM kann nicht angesprochen werden
2	ein Fehler bei der JNI-Kommunikation (siehe Log-Viewer für weitere Informationen)
3	ein Fehler im d.3 Server ist aufgetreten (siehe Log-Viewer für weitere Informationen)

5.21.2 `java_clear_param`

Diese Funktion löscht alle bisher hinzugefügten Parameter (siehe `java_add_param`).



verfügbar ab d.3 Version 7.2

Parameter	Beschreibung
	keine

Rückgabewert	Bedeutung
0	alles OK
1	Java-Support nicht aktiviert / Java-VM kann nicht angesprochen werden
2	ein Fehler bei der JNI-Kommunikation (siehe Log-Viewer für weitere Informationen)
3	ein Fehler im d.3 Server ist aufgetreten (siehe Log-Viewer für weitere Informationen)

5.21.3 java_call_static

Diese Funktion ruft eine statische Funktion einer Java-Klasse auf. Die zuvor hinzugefügten Parameter werden übergeben. Die Klasse muss sich dabei im Java CLASSPATH befinden.



verfügbar ab d.3 Version 7.2

Die aufzurufende Methode muss folgende Signatur haben:

```
static int javaTest(FieldInterface field, String[] params)
```

Das entspricht der Signatur von Java-Hook Funktionen. Der Rückgabewert der Java-Funktion wird in das globale Feld `api_single_info` geschrieben.



Nach einem Aufruf dieser Funktion werden die hinzugefügten Parameter automatisch gelöscht. Ein expliziter Aufruf von `java_clear_params` ist nicht notwendig.

Parameter	Beschreibung
class_name	der vollqualifizierte Klassenname in JNI Notation; hier muss der Klassenname inklusive Namensraum angegeben werden. Punkte (.) werden durch Schrägstriche (/) ersetzt; Beispiel: "com/dvelop/test/
JavaHooks	dies entspricht der Java-Klasse <code>JavaHooks</code> im Package <code>com.dvelop.test</code>
method_name	der Name der Methode, die aufgerufen werden soll; die Methode muss statisch sein und die oben beschriebene Signatur aufweisen; Beispiel: "javaTest"

Rückgabewert	Bedeutung
0	alles OK
1	Java-Support nicht aktiviert / Java-VM kann nicht angesprochen werden
2	ein Fehler bei der JNI-Kommunikation (siehe Log-Viewer für weitere Informationen)
3	ein Fehler im d.3 Server ist aufgetreten (siehe Log-Viewer für weitere Informationen)

6 JPL-Styleguide

6.1 Ablage und Name

Alle Hookfunktionen sollten nach Möglichkeit in einer Datei zusammengefasst werden. Vermeiden Sie mehrere Dateien mit jeweils einzelnen Hook-Funktionen (z.B. `HookInsertEntry_10`).

Nach Möglichkeit eine Hook-Datei mit folgenden Eigenschaften:

- Name sollte den Firmen-Namen bzw. ein Kunden-Kürzel beinhalten.
- Die JPL-Datei sollte im d.3-Server-Verzeichnis in einem eigenen Bereich abgelegt werden:



C:\d3\d3server\hooks\d3P\testfirma.jpl

6.2 Kommentare

Kommentare dienen der besseren Lesbarkeit und Wartung.

Der Kommentar selber wird in JPL mit zwei `//` eingeleitet. Diese beiden Zeichen können sich in jeder beliebigen Spalte des Codes befinden. Sollten aber auf gleicher Ebene wie der Code liegen (Einrückung). Eine einzelne Kommentarzeile sieht wie folgt aus.



// Kommentarzeile

Mehrzeilige Kommentarzeilen sollten mit einer Start- und Endzeile geklammert sein.



```
//-----
// Kommentarzeile
// Kommentarzeile
//-----
```

6.3 Kopf

Der Kopf einer JPL-Datei, dies gilt für externe JPI-Scripte als auch für JPL-Dateien für Hook-Funktionen, muss folgende Informationen enthalten:

- Name der Hook-Funktion
- Wann erstellt und von Wem
- Grobe Beschreibung
- Änderungslegende

In der Änderungslegende stehen auch die Infos Wer und Wann. Des Weiteren kann auch mit einer Nummerierung gearbeitet werden. So kann im Kopf die Änderungsnr 007 referenziert werden. Im Code selber taucht diese Nummer ebenfalls auf. So können die betroffenen Zeilen sehr schnell gefunden werden.



```
//-----  
// 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
//-----
```

Der Code selber sollte natürlich auch hinreichend kommentiert werden.

Empfehlung:

- Größere Logische Blöcke z.B. Schleifen , Abfragen kommentieren
- Änderungsnummer aus dem Hookkopf sollte hier auftauchen

6.4 Variablen

- Generell
- Repositoryfelder und Dokumentarten

6.4.1 Generell

Die Definition von Variablen erfolgt immer im Kopf des Hooks oder einer Prozedur.

Deklaration von Variablen oder Konstanten innerhalb des Codes ist untersagt bzw. sollte auf ein Minimum eingeschränkt werden.

Variablen innerhalb der JPL-Datei werden mit `h_[VarName]` definiert.



h_Count oder h_SachBearbeiter oder h_VersNr.

6.4.2 Repositoryfelder und Dokumentarten

In den einzelnen Hookfunktionen wird immer wieder auf Dokumentarten und deren Felder zugegriffen. Um die Lesbarkeit des Codes zu vereinfachen, sollte nach folgendem Schema vorgegangen werden.

Im Kopf des Hooks werden folgende Konstanten definiert:

- Repository-Felder / h_DDF = Dok_Dat_Feld

h_DDF_‘Name’ = ,DB-Position’

Die Konstanten sind sortiert nach ,DB-Position’

- Kürzel der Dokumentarten DA = DokuArt

h_DA_‘NAME’ = ,Kürzel der Dokuart’

Die Konstanten sind sortiert nach ,Kürzel der dokuart’



```
/-----
// DB-Position der Repository-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_AAUFT         = "AAUFT"
```

Der Zugriff könnte dann wie folgt aussehen:



```
// 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]
```

6.5 Debugging

Jede Prozedur besitzt mindestens zwei Debuggingfunktionen, diese setzen den Start und das Ende der Funktion. Jeder weitere Aufruf der Debuggingfunktion enthält ebenfalls den Namen der Funktion.

```
$_
proc FunktionsName
{
    call api_log_debug ("FunktionsName start")
    // Variablen Deklaration
    ...
    // Code
    ...
    call api_log_debug ("FunktionsName Irgendeine Meldung")
    // Code
    ...
    call api_log_debug ("FunktionsName end")
}
```

6.6 Schleifen und Verzweigungen



Jede einzelne Ebene wird eingerückt.

- For-Schleife
- While-Schleife
- If-Abfrage

6.6.1 For-Schleife

Möglichst als Zähl-Variable keine Variablen mit nur einem Buchstaben als Namen verwenden! Hier sollte mit Variablen in der Form h_Count genutzt werden.

```
$_
vars h_Count
for h_Count = 0 while(string( h_Count, 1) != "") step 1
{
    Anweisungs-Block
}
```

6.6.2 While-Schleife

```
$_ vars h_Count = 1
while (string(h_Count,1) != " ")
{
    h_Count = h_Count+1
}
```

6.6.3 If-Abfrage

Wenn mehrere Bedingungen geprüft werden, steht jede Bedingung in einer einzelnen Zeile. Des Weiteren sollte mit Klammern gearbeitet werden; dies unterstützt die Lesbarkeit. Die Bedingungen sollten auch optisch untereinander stehen.

```
$_ if      (      (x > y) \
&& (z < x) \
|| (x > w))
{
    Anweisungs-Block
}
else
{
    Anweisungs-Block
}
```

7 Nützliche Funktionen

7.1 WriteNotizEntry

Diese Funktion liest die Werte aus dem 61er-Feld aus und schreibt sie in die Notizen.

```
 // 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
```

7.2 AddPrefixAndZeroes

Diese Funktion fügt einer übergebenen Zeichkette einen ebenfalls übergebenen Präfix und Nullen zu, um die Zeichenkette auf die Länge wf_MaxLength zu bringen. Beginnt die Zeichenkette bereits mit dem gewünschten Präfix, werden keine Änderungen vorgenommen.

```
$_ // 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
```

7.3 CutPrefixAndZeros

Die Funktion `CutPrefixAndZeros` entfernt von einer Zeichenkette das Präfix und die folgenden Nullen.

```
$_ 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
```

7.4 OpenSessionConnection

Die folgende Funktion öffnet eine ODBC-Verknüpfung.

```
$_ 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
}
```

7.5 GruppeAnwesend

Die Funktion GruppeAnwesend (h_Gruppe) ermittelt, ob mindestens ein Mitglied der Gruppe anwesend ist.



```

// Prüfen, ob min. 1 Mitglied der Gruppe anwesend

proc GruppeAnwesend( h_Gruppe )
{
    vars h_TmpFlag, h_MaxCount = 0, h_Count = 0, h_ReturnValue

    // Ermittlung aller Mitglieder einer Gruppe
    if (BenutzerIstGruppe (h_Gruppe ) == 1)
    {
        h_ReturnValue = api_function ("group_get_members", h_Gruppe)
        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
}
```

7.6 BenutzerAnwesend

Mit dieser Funktion wird geprüft, ob der übergebene Benutzer zur Zeit anwesend ist.



```

// 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
```



```
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
```

7.7 BenutzerIstGruppe

BenutzerIstGruppe ermittelt, ob der übergebene User eine Gruppe ist.



```
// 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
```

7.8 ErmittlungBenutzer

Die Funktion ErmittlungBenutzer bestimmt Benutzer anhand einer PESNr die im OptionalenFeld1 der Benutzerdaten hinterlegt wurde.

```

// 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

```

7.9 fillWmFromDbField

Funktion zum Ermitteln von Werten (=> Wertemenge) aus einer Datenbank anhand vorgegebener Kriterien (Eingabe: Tabelle, Spalte, Wert (Schlüssel) => Wertemenge wird mit den Werten anderer Spalte gefüllt).

```

//*****
// * Funktion zum Ermitteln von Werten (> Wertemenge)
// * aus einer Datenbank anhand vorgegebener Kriterien
// * (Eingabe: Tabelle, Spalte, Wert (Schlüssel) -->
// * 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, hRowCount

    DBMS ALIAS h_dbValue
    DBMS SQL SELECT Count (*) FROM :dbTable WHERE :dbColSrc \
                  like ':baseValue%'
    h_dbValue = @dmrowcount
    DBMS ALIAS

```

```

        if( h_dbValue &gt; 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

```

7.10 getValueFromDBField

Funktion zum Ermitteln eines Wertes aus einer Datenbank anhand vorgegebener Kriterien (Eingabe: Tabelle, Spalte, Wert (Schlüssel) --> Ausgabe: Wert anderer Spalte).



```

//*****
// * 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, hRowCount, 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
}

```

7.11 hasDtsReposId

Funktion, um herauszufinden, ob eine Dokumentart ein Repository besitzt (anhand der Repository ID).



```

// ****
// * 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
}
```

7.12 AvailableValues

Wertemenge, die alle Werte liefert, die bereits zu diesem dok_dat_feld (Dokuart abhängig) im d.3 System vorhanden sind; eingegebene Werte in dem dok_dat_feld werden mit in den Query eingebaut (mit LIKE 'xxx%').

```
//*****
// * 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
        // Wertemengeninhalt 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
```

8 Webservice-Funktionen

d.3 webservice_Funktionen



Sofern nicht anders angegeben, geben alle Funktionen bei Erfolg den String `OK` zurück, ansonsten eine Fehlermeldung.

Initialisierung	
<code>d3ws_startSession()</code>	Initialisierung, gibt eine Session-ID zurück (z.Zt. ohne Verwendung)
<code>d3ws_stopSession()</code>	Finalisierung (optional, entspricht neuer Initialisierung)

XML-Erzeugung (-> XML-Out-Buffer)	
<code>d3ws_xml_start()</code>	Beginn eines neuen XML-Dokuments, Löschen des bisherigen Puffers
<code>d3ws_xml_openNode(name)</code>	öffnet einen neuen Knoten, der mit <code>CloseNode</code> wieder geschlossen werden muss (also " <code><name></code> ")
<code>d3ws_xml_addNode(name, value)</code>	erzeugt einen kompletten Knoten (also " <code><name>value</name></code> ")
<code>d3ws_xml_closeNode(name)</code>	schließt einen Knoten, der mit <code>OpenNode</code> geöffnet wurde (also " <code></name></code> ")
<code>d3ws_xml_stop()</code>	beendet das XML-Dokument, das jetzt gespeichert oder versendet werden kann

Laden, Speichern, Aufruf Web-Service	
<code>d3ws_xml_saveToFile(filename)</code>	speichert den XML-Out-Buffer in die angegebene Datei
<code>d3ws_xml_loadFromFile(filename)</code>	lädt die angegebene Datei in den XML-In-Buffer
<code>d3ws_http_addParameter(name, value)</code>	fügt einen Parameter hinzu, der beim späteren Aufruf von <code>sendRequest</code> als POST-Parameter mitgegeben wird
<code>d3ws_http_setTraceFile(filename)</code>	protokolliert nachfolgenden Aufruf von <code>sendRequest</code> in einer Datei (zur Fehlersuche geeignet)
<code>d3ws_http_sendRequest(url)</code>	Sendet den XML-Out-Buffer an die angegebene URL und lädt das Ergebnis des Requests in den XML-In-Buffer. In der URL können GET-Parameter verwendet werden.

**Laden, Speichern, Aufruf Web-Service**

Wenn zuvor KEIN weiterer Parameter mitgegeben wurde (addParameter), wird das XML per POST namenlos mitgesendet. Andernfalls wird das XML per POST als Parameter "xml=" mitgesendet.

XML-Parser (-> XML-In-Buffer)**Navigation**

d3ws_xml_setRootNode()	setzt den Fokus auf den Root-Node des XML-Dokuments
d3ws_xml_setParentNode()	setzt den Fokus auf den Parent-Node des bisher fokussierten Nodes
d3ws_xml_setNode (expression)	<p>Setzt den Fokus auf einen beliebigen Node (relativ zum aktuell fokussierten Node)</p> <p>expression kann sein:</p> <ul style="list-style-type: none">• " / " für den Root-Node• " . . " für den Parent-Node• " 1 " – "n" für den n'ten Child-Node <p>(ggf, vorher mit countNodes durchzählen)</p> <ul style="list-style-type: none">• "name" für einen bestimmten Child-Node <p>eine beliebige Kombination der obigen Angaben (als Pfadangabe, getrennt mit " / ")</p>
d3ws_xml_setNodeByIndex (i)	Analog zu setNode("i"), allerdings kann hier wirklich ein Integer übergeben werden (setNode erwartet einen String)

Analyse

d3ws_xml_countNodes()	liefert die Anzahl der verfügbaren Child-Nodes unterhalb des gerade fokussierten Nodes
d3ws_xml_getnodeName (i)	liefert den Namen des i'ten Nodes

Zugriff auf enthaltene Werte

d3ws_xml_getnodeValue (expression)	<p>liefert den Inhalt des mit expression bezeichneten Nodes (relativ zum aktuell fokussierten Node);</p> <p>Verwendung von expression wie in setNode</p>
---------------------------------------	--



9 Index

D

Dateizugriffe 10
doc_field 11
doc_id 11

G

Globale Variablen 11, 21, 22, 22, 23, 23, 24, 24, 25, 25

H

Hook 10

L

Löschen eines Verzeichnisses 89

N

Notiz 107
Notizdatei 107
Notizdatei anhängen 107

V

Variablen global 11
Verknüpfungen 10

W

Workflow 10