

d.velop

DSR 8.1.0 (Dokumentation d.3
Server API 8.1.0)

Inhaltsverzeichnis

1. d.3 server scripting api (jpl)	4
1.1. Einleitung	4
1.1.1. Vereinbarungen	4
1.1.2. Über die d.3 Server API	4
1.1.3. Voraussetzungen	5
1.2. Allgemeine Informationen zur d.3 Server API	5
1.2.1. Überblick	5
1.2.2. Funktionsaufruf	8
1.2.3. Logging	8
1.2.4. Beispiele	9
1.3. Globale Variablen	12
1.3.1. Dokumentkontext	12
1.3.2. Benutzerkontext	13
1.3.3. Objektkontext	14
1.3.4. Workflowkontext	14
1.3.5. Postkorbkontext	15
1.3.6. Sperrkontext	15
1.3.7. Anwendungskontext	15
1.3.8. Sonstige Variablen	17
1.3.9. Error Handling	17
1.4. Funktionen	17
1.4.1. Übersicht über die Funktionen	17
1.4.2. Dokument	22
1.4.3. Akte	63
1.4.4. Verknüpfungen	65
1.4.5. Verzeichnisse	74
1.4.6. Status	79
1.4.7. Notizen	91
1.4.8. Wiedervorlage	94
1.4.9. Workflow	105
1.4.10. Dokumentart	110
1.4.11. Wertemengen	114
1.4.12. Benutzer	118
1.4.13. Gruppen	133
1.4.14. Rollen	139
1.4.15. Restriktionsmengen	146
1.4.16. Tiff-Funktionen	149
1.4.17. Dateizugriffe	152
1.4.18. Stringkonvertierung	161
1.4.19. Diverses	175
1.4.20. d3fc-Calls	191
1.4.21. JAVA	203
1.4.22. Erstellung von Async-Jobs	205
1.5. JPL-Styleguide	206
1.5.1. Ablage und Name	206
1.5.2. Kommentare	206
1.5.3. Kopf	207
1.5.4. Variablen	209
1.5.5. Debugging	210
1.5.6. Schleifen und Verzweigungen	211
1.6. Nützliche Funktionen	212
1.6.1. WriteNotizEntry	212
1.6.2. AddPrefixAndZeroes	213

1.6.3. CutPrefixAndZeros	213
1.6.4. OpenSessionConnection	213
1.6.5. GruppeAnwesend	215
1.6.6. BenutzerAnwesend	215
1.6.7. BenutzerIstGruppe	216
1.6.8. ErmittlungBenutzer	216
1.6.9. fillWmFromDbField	217
1.6.10. getValueFromDBField	218
1.6.11. hasDtsReposId	219
1.6.12. AvailableValues	220
1.7. Webservice-Funktionen	222
1.8. Weitere Informationsquellen und Impressum	223

1. d.3 server scripting api (jpl)

1.1. Einleitung

1.1.1. 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 [Sons- tiges]	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.

Anmerkung

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

Warnung

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

Anmerkung

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

Anmerkung

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

Anmerkung

Der Pfad zu einem Verzeichnis, Parameter und Dateinamen werden in dieser `Schriftart` dargestellt.

Anmerkung

So markierte Abschnitte zeigen Ihnen, sofern vorhanden, den Link zu einem dem Thema entsprechenden Video.

1.1.2. Über die d.3 Server API

Über die d.3 Server API können Sie direkt auf d.3 Server-Funktionen zurückgreifen. Dazu gehören:

- dokumentbezogene Informationen (Eigenschaften, Verknüpfungen ...)

- dokumentartbezogene Informationen (erweiterte Eigenschaften ...)
- benutzerbezogene Informationen (Gruppenzugehörigkeit ...)

Die oben genannten Funktionen können dabei gelesen und geändert werden.

Die Entwickler können über die Schnittstelle langzeitstabile Funktionen integrieren, die den Zugriff auf die d.3-Daten sichern. Außerdem stellt die API viele Hilfsfunktionen bereit, die die Umsetzung vereinfachen.

Diese Dokumentation steht Entwicklungspartnern der d.velop AG im Service Portal online bereit. Die Weitergabe dieser Dokumentation oder von Teilen daraus ist nicht gestattet. Bei Anfragen im Rahmen der Entwicklungspartnerschaft gilt stets nur die Onlinedokumentation.

Bitte beachten Sie, dass Ihre Software über diese Schnittstelle auch Zugriff auf die von Ihren Kunden im d.3ecm abgelegten und konfigurierten Daten erhält und Eingriff in die Abläufe im d.3ecm-System nimmt. Gehen Sie daher bitte sorgfältig vor und achten Sie darauf, dass Ihre Anwendung Teil eines bestehenden Zusammenspiels unterschiedlicher Anwendungen ist. Die unsachgemäße Verwendung dieser Schnittstelle kann veränderte Programmabläufe und Datenverlust zur Folge haben.

Die Software-Entwicklung mit dieser Programmierschnittstelle ist Individualentwicklung. Der von Ihnen erzeugte Programmcode fällt nicht unter die Pflege- und Supportbedingungen der Produkte der d.velop AG. Unser Support unterstützt Sie gerne, Ihre Anfragen sind jedoch kostenpflichtig, sofern sich die Anfrage nicht auf einen Fehler in unseren Produkten zurückführen lässt.

Alle Fragen zu den Voraussetzungen und zur Software-Entwicklung mit d.3ecm beantwortet Ihnen gerne das Technology Partner Management der d.velop AG.

1.1.3. Voraussetzungen

Warnung

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.

1.2. Allgemeine Informationen zur d.3 Server API

1.2.1. 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 Dokument-ID in Form der Variablen `api_doc_id` und damit der Dokumentenkontext gesetzt werden. Die nachfolgenden Funktionen arbeiten dann alle im Kontext dieser Dokument-ID. Werden nun Operationen wie Eigenschaftsänderungen etc. vorgenommen, beziehen sich diese immer auf diese Dokument-ID.

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 ()`

Anmerkung

Ab d.3 Version 8.0 ist im d.3 admin und den d.3-Anwendungen nicht mehr von Attributen, sondern von Eigenschaften die Rede. Die API-Calls, welche sich auf Attribute/Eigenschaften beziehen und in dieser Dokumentation beschrieben sind, wurden nicht umbenannt.

`api_hook_user_set (user_name)`

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

<code>api_hook_user_set (user_name)</code>	
Parameter	<code>user_name</code> Name des Benutzers, in dessen Namen nun gearbeitet wird
Rückgabe	0 fehlerfrei -1 <code>user_name</code> wurde nicht angegeben -2 <code>user_name</code> ist kein gültiger d.3-Benutzername

`api_hook_user_reset ()`

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

<code>api_hook_user_reset ()</code>	
Parameter	keine
Rückgabe	Name des Benutzers, der zuvor als Benutzer verwendet wurde.

`api_hook_user_get ()`

Die Funktion ermittelt den ausführenden Benutzer.

<code>api_hook_user_get ()</code>	
Parameter	keine
Rückgabe	Name des ausführenden Benutzers

Anmerkung

```

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

```

api_hook_user_master()

Diese Funktion setzt den "Master" als ausführenden Benutzer.

api_hook_user_master()	
Parameter	keine
Rückgabe	keine

Anmerkung

seit d.3-Version 6.0 HF2

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

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

api_clear_document_context()

leert alle dokumentbezogenen Felder

api_clear_document_context()	
Parameter	keine

api_clear_document_context()	
Rückgabe	0 fehlerfrei

api_clear_user_context()

leert alle benutzerbezogenen Felder

api_clear_user_context()	
Parameter	keine
Rückgabe	0 fehlerfrei

1.2.2. 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

Anmerkung

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	

1.2.3. 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	
<code>api_log_debug(log_message)</code>	Debug-Logging
<code>api_log_info(log_message)</code>	Info-Logging
<code>api_log_error(log_message)</code>	Fehler_Logging

1.2.4. Beispiele

Beispiel 1

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

Anmerkung

```
//-----
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
```

Beispiel 2

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

Anmerkung

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

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

  // Feld 63 setzen
  h_Count2 = 1
  for h_Count1 = 1 while (h_Count1 &lt;= h_ReturnValue) step 1
  {
    if (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
```

Anmerkung

```

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

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

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

```

1.3. Globale Variablen**1.3.1. Dokumentkontext**

Variable	Beschreibung
api_doc_id	Dokument-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 Mehrfacheigenschaftsfelder
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

Variable	Beschreibung
api_doc_text	Bemerkungstext <div style="border: 1px solid black; padding: 5px; margin: 5px 0;"> <p>Warnung Das globale Feld <code>api_doc_text</code> wird nicht mehr unterstützt.</p> <p>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> <p>Verwenden Sie stattdessen die Variable <code>api_doc_texts[1..4]</code>.</p> </div>
api_doc_ids[MAX_DOC_IDS]	Dokument-ID-Array
api_doc_owner	Besitzer
api_doc_file_name	Dateiname
api_doc_file_ext	Dateierweiterung
api_doc_encrypt	Verschlüsselung (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	Zeitstempel der letzten Änderung
api_doc_web_published	webveröffentlicht (0/1)
api_doc_workflow	in einem Leitweg befindlich (0/1)
api_links[MAX_LINKS]	über-/untergeordnete Dokumente
api_link_creators[MAX_LINKS]	Erzeuger der Verknüpfung (a: Aktenplan, m: manuell, l: a00-Datei) (für <code>link_get_children</code> und <code>link_get_parents</code>)
api_link_tstamps[MAX_LINKS]	Zeitstempel, wann die Verknüpfung angelegt wurde (für <code>link_get_children</code> und <code>link_get_parents</code>)
api_doc_types[MAX_DOC_TYPES]	Dokumentarten
api_classes[MAX_CLASSES]	Array mit allen zugeordneten Klassen
api_class_right[MAX_CLASSES]	Array mit den zugeordneten Rechten für die Klasse mit gleichem Index in <code>api_classes</code> (0=verweigert, 1=erteilt, 2=keine Angabe) Position 1: Lesen Position 2: Schreiben Position 3: Prüfen Position 4: Freigeben
api_color_code	Farbcode des Dokuments 0: nicht gesetzt 1-24: Farbcode <div style="border: 1px solid black; padding: 5px; margin: 5px 0;"> <p>Anmerkung neue Variable ab Version 7.2</p> </div>

1.3.2. Benutzerkontext

Variable	Beschreibung
api_users[MAX_USERS]	Benutzernamen
api_groups[MAX_GROUPS]	Gruppennamen
api_rolls[MAX_ROLLS]	Benutzernamen

Variable	Beschreibung
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
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

1.3.3. 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, ...)

1.3.4. 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
api_wfl_header_name	LeitwegName
api_wfl_step_recipient	StepEmpfänger
api_wfl_step_type	StepTyp
api_wfl_step_action_type	Aktionstyp

Variable	Beschreibung
api_wfl_step_action_string_1	Option1 der Aktion
api_wfl_step_action_string_2	Option2 der Aktion
api_wfl_path_initiator	Pfad-Initiator des Leitweges

1.3.5. 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)

1.3.6. 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
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

1.3.7. Anwendungskontext

In die globale Variable `d3server_letzte_api_applikation` wird die Applikations-ID der Anwendung geschrieben, die den Hook angestossen hat.

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

Anmerkung

```
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".

Anmerkung

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

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

Mit der unten stehenden Liste erhalten Sie einen Überblick über die gängigen Applikations-ID der aufrufenden Anwendungen:

Anwendung	ID
d.3 smart explorer	200
d.3 import	220
d.3 smart start	495
d.3 view	100
d.3 folder browser	499
d.capture batch	430
d.capture batch importer	433
d.capture dialog	437
d.classify	501
d.cold	550
d.ecs forms	908
d.3 virtual user	253
d.3 smart suite ce	020
d.3 presentation server	901
d.3 web webservice	247
d.3 web	260
d.3 image service	483

Anwendung	ID
d.exchange	460
d.link for microsoft outlook	465

1.3.8. Sonstige Variablen

Sonstige Variablen	Beschreibung
<code>api_converted_string</code>	nimmt den konvertierten String der Funktionen <code>string_...</code> auf
<code>api splitted_string [MAX_SPLIT]</code>	enthält die Teilstrings der Funktion <code>string_split</code>
<code>api_single_info</code>	enthält einen einzelnen Rückgabestring (z.B. einzelne Eigenschaft)
<code>api_data_type</code>	Datentyp bei Wertemengen
<code>api_list[]</code>	Array zur Aufnahme der Einträge von <code>directory_list</code> ab d.3 Version 6.0.1 Hotfix 1
<code>MAX_DOC_IDS</code>	Größe des DokID-Arrays
<code>RIGHT_ANY</code>	jegliches Recht
<code>RIGHT_READ</code>	Leserecht
<code>RIGHT_WRITE</code>	Schreibrecht
<code>RIGHT_VERIFY</code>	Prüfungsrecht
<code>RIGHT_RELEASE</code>	Freigaberecht

1.3.9. Error Handling

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

1.4. Funktionen

1.4.1. Übersicht über die Funktionen

- Dokument
- Akte
- Verknüpfungen
- Verzeichnisse
- Status
- Notizen
- Wiedervorlage
- Workflow
- Dokumentart
- Wertemengen
- Benutzer
- Gruppen
- Rollen

- TIFF-Funktionen
- Dateizugriffe
- Stringkonvertierung
- D3fc-Calls
- Diverses

Anmerkung

Die Bedeutung der Rückgabewerte aller Funktionen der d.3 Server API können auch den Dateien `fehler.txt` und `errors.txt` entnommen werden. Diese befinden sich auf dem d.3 Server Host im Verzeichnis `...\\d3server.prg`.

Dokument (Übersicht)

Dokumentfunktionen	Beschreibung
<code>attribute_get_single</code>	einzelnes Attribut
<code>attribute_get_all</code>	alles über Kontext
<code>attribute_update_single</code>	einzelnes Attribut
<code>attribute_update_all</code>	alles über Kontext
<code>attribute_get_valid_values</code>	Abfrage erlaubter Kenndaten
<code>attribute_test_value</code>	Wert auf Gültigkeit prüfen
<code>attribute_update_begin</code>	schreibt die aktuellen Attribute des Dokuments in die <code>api_doc_fields</code>
<code>attribute_update_execute</code>	verändert die <code>api_doc_fields</code> nach Aufruf von <code>attribute_update_begin</code>
<code>document_change_type</code>	Dokumentart wechseln
<code>document_get_valid_values</code>	erlaubte Kenndaten abfragen
<code>document_delete</code>	Dokument löschen
<code>document_find_ids</code>	Dokumente zu Kontext suchen
<code>document_get_permission</code>	Abfrageberechtigungen eines Benutzers auf ein Dokument
<code>document_publish_for_web</code>	markiert ein Dokument als (Nicht-) Web-Veröffentlicht
<code>document_dependent_add</code>	legt ein abhängiges Dokument ab bzw. überschreibt ein existierendes
<code>document_dependent_delete</code>	löscht ein abhängiges Dokument
<code>document_get_file_path</code>	ermittelt zu vorgegebenen Parametern den kompletten Pfad im d.3-Dokbaum unter dem das Dokument stehen müsste
<code>document_start_lifetime</code>	setzt die Lebensdauer eines Dokuments
<code>document_set_cache_days</code>	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
<code>document_send_to_dsearch</code>	sendet OCR-Informationen einer Dokumentversion an d.3 search
<code>add_inherit_doc_rights</code>	Funktion zur Vererbung von Rechten auf ein Dokument
<code>remove_inherit_doc_rights</code>	Funktion zum Löschen von vererbten Rechten auf ein Dokument

Akte (Übersicht)

Aktenfunktionen	Beschreibung
<code>folder_create</code>	Akte anlegen

Verknüpfungen (Übersicht)

Verknüpfungsfunktionen	Beschreibung
<code>link_get_parents</code>	übergeordnete Dokumente abfragen

Verknüpfungsfunktionen	Beschreibung
<code>link_get_children</code>	untergeordnete Dokumente abfragen
<code>link_documents</code>	zwei Dokumente miteinander verknüpfen
<code>link_delete</code>	Verknüpfung löschen
<code>link_exists</code>	Testen, ob zwei Dokumente verknüpft sind

Verzeichnisse (Übersicht)

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

Status (Übersicht)

Statusfunktionen	Beschreibung
<code>document_block</code>	Dokument sperren
<code>document_verify</code>	Dokument prüfen
<code>document_transfer</code>	Statustransfer durchführen

Notizen (Übersicht)

Notizfunktionen	Beschreibung
<code>note_add_file</code>	Dateiinhalte an Notizdatei anhängen
<code>note_add_string</code>	Text an Notizdatei anhängen

Wiedervorlage (Übersicht)

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

Workflow (Übersicht)

Workflowfunktionen	Beschreibung
<code>workpath_get_document_info</code>	Infos zu aktuellem Leitweg abfragen
<code>workpath_end_document</code>	Dokument aus Leitweg entfernen
<code>workpath_start_document</code>	Dokument in Leitweg stellen
<code>workpath_go_to_next_step</code>	Dokument im Workflow weiterschieben

Dokumentart (Übersicht)

Dokumentartfunktionen	Beschreibung
<code>document_type_get_long</code>	Dokumentart-Langtext bestimmen
<code>document_type_get_short</code>	Dokumentart-Kürzel bestimmen

Wertemengen (Übersicht)

Wertemengenfunktionen	Beschreibung
<code>predefined_values_get</code>	Wertemenge abfragen
<code>predefined_values_test</code>	Werte gegen Wertemenge prüfen

Benutzer (Übersicht)

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

Gruppen (Übersicht)

Gruppenfunktionen	Beschreibung
<code>group_get</code>	Gruppennamen abfragen
<code>group_get_members</code>	Mitglieder einer Gruppe abfragen
<code>group_get_verifiers</code>	Prüfer einer Gruppe abfragen
<code>group_get_doc_types</code>	Dokumentarten einer Gruppe abfragen

Rollen (Übersicht)

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

Restriktionsmengen (Übersicht)

<code>d3set_add_filter</code>	fügt einen Filter zu einer bestehenden Restriktionsmenge hinzu
<code>d3set_remove_filter</code>	entfernt einen Filter von einer bestehenden Restriktionsmenge
<code>d3set_get_filter</code>	ermittelt den Filter einer bestehenden Restriktionsmenge
<code>d3set_get_sets</code>	sucht über bestimmte Kriterien nach Restriktionsmengen-Zuordnungen und liefert diese in einer Tabelle zurück
<code>d3set_remove_set</code>	löscht eine bestehende Restriktionsmenge

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	vertiffet die Workflow-Protokolle zu einer Dokument-ID und erzeugt abhängige w1-Datei

Dateizugriffe (Übersicht)

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

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 Position 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
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

Diverses (Übersicht)

Diverse Funktionen	Beschreibung
restore_from_jukebox	von Jukebox wiedereinlagern

Diverse Funktionen	Beschreibung
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 Kommandozeilenaufwurf durch (Start eines externen Programms)
get_current_datetime	liefert den aktuellen Zeitstempel im gewünschten Format zurück
sleep	"schläft" für eine Anzahl an Millisekunden ohne Prozessorauslastung
process_get_id	Gibt den Wert der Prozess ID in der globalen Variable <code>api_single_info</code> (hexadecimal) 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
get_objectid_from_synchid	ermittelt aus einer Synchronisations-ID, die über das Transportsystem erstellt wurden, die interne ID und den Typen eines Stammdaten-Objektes

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_importing_set	Import-Parameter setzen
3fc_exporting_set	Export-Parameter setzen
d3fc_exporting_get	Export-Parameter abfragen nach <code>d3fc_execute</code>
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 Import-Tabelle 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

JAVA (Übersicht)

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

1.4.2. Dokument

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	<p>Position des firmenspezifischen Feldes, das abgefragt wird oder Name des allgemeinen Attributes:</p> <p>doc_date_import: Importdatum</p> <p>doc_last_alter_no: Nummer der letzten Änderung</p> <p>doc_publ_alter_no: Nummer der Version in Freigabe</p> <p>doc_status: Dokumentstatus</p> <p>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)</p> <p>doc_editor: Bearbeiter</p> <p>doc_type_short: Kürzel der Dokumentart</p> <p>doc_no: Dokumentnummer (zeich_nr)</p> <p>doc_var_no: Variantennummer</p> <p>doc_text_1: Bemerkungstext</p> <p>doc_owner: Besitzer</p> <p>doc_file_name: Dateiname</p> <p>doc_file_ext: Dateierweiterung</p> <p>doc_encrypt: Verschlüsselt ? (0/1)</p> <p>doc_cnt_access: Anzahl der Zugriffe</p> <p>doc_date_last_access: Datum des letzten Zugriffs</p> <p>doc_user_last_access: Benutzer des letzten Zugriffs</p> <p>doc_last_update_file: letzte Dateänderung</p> <p>doc_web_published: webveröffentlicht ? (0/1)</p> <p>doc_workflow: im Leitweg ? (0/1)</p> <p>doc_type_type: a für Akte und d für Dokument</p> <p>doc_color_code: Farbcode des Dokuments (0: nicht gesetzt; 1-24: Farbcode).</p>
	<p>Anmerkung neues Attribut doc_color_code ab Version 7.2.</p>
	Pflichtfeld
row_no	<p>Zeilennummer bei Mehrfachattributfeldern</p> <p>optional; Defaultwert ist 0</p>
doc_id	<p>Dokument-ID, für die das Attribut abgefragt wird; wird nichts angegeben, wird die ID aus dem Kontext verwendet</p> <p>Pflichtfeld</p>
user_name	<p>Name des Benutzers, der das Attribut abfragt</p> <p>Optional</p>

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

Anmerkung

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

Anmerkung

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

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

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

Anmerkung

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

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

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

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

Anmerkung

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

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

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

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

attribute_get_all

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
api_doc_file_ext	Dateierweiterung
api_doc_encrypt	Verschlüsselt ? (0/1)
api_doc_cnt_access	Anzahl Zugriffe

Globale Variable	Beschreibung
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	Farbcode des Dokuments (0: nicht gesetzt; 1-24: Farbcode)

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
-303	Benutzer hat kein Zugriffsrecht auf das Dokument

Anmerkung

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.

Anmerkung

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

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

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

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

attribute_update_single

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
<code>db_pos</code>	Position des firmenspezifischen Feldes, das abgefragt wird oder Name des allgemeinen Attributes: <code>doc_text_1</code> : Bemerkungstext (Pflichtfeld) <code>doc_color_code</code> : Farbcode des Dokuments (0: nicht gesetzt; 1-24: Farbcode).
	Anmerkung neues Attribut <code>doc_color_code</code> ab Version 7.2
	Pflichtfeld
<code>new_value</code>	Wert, den das Attribut erhalten soll wird bei <code>row_no = -1</code> ignoriert, ansonsten Pflichtfeld

Parameter	Beschreibung
row_no	<p>Zeilennummer bei Mehrfachattributfeldern</p> <p>0 = als nächste Zeile anhängen (default)</p> <p>-1 = bei "normalen" Attributfeldern: Wert aus dem entsprechenden <code>api_doc_field</code> verwenden und <code>new_value</code> ignorieren; bei Mehrfachattributfeldern: alle Zeilen des Mehrfachattributfeldes löschen</p> <p>Optional</p>
doc_id	<p>Dokument-ID, für die das Attribut abgefragt wird; wird nichts angegeben, wird die ID aus dem Kontext verwendet</p> <p>Pflichtfeld - entweder als Parameter oder im Kontext: <code>api_doc_id</code></p>
user_name	Name des Benutzers, der das Attribut speichert (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	<p>kein neuer Wert für das Pflichtfeld <code><\:db_pos></code> in der Dokumentart</p> <p><code><\:doc_type_short></code> angegeben:</p> <p><code>api_error_var[..] = "doc_type_short"</code></p> <p><code>api_error_value[..] = Kürzel der Dokumentart</code></p> <p><code>api_error_var[..] = "modifiable"</code></p> <p><code>api_error_value[..] = Kürzel der Dokumentart</code></p> <p><code>api_error_var[..] = "modifiable"</code></p> <p><code>api_error_value[..] = Feld ist modifizierbar (0:nein; 1: ja)</code></p> <p><code>api_error_var[..] = "doc_flags "</code></p> <p><code>api_error_value[..] = Feldeigenschaften</code></p> <p><code>api_error_var[..] = "data_type "</code></p> <p><code>api_error_value[..] = Datentyp des Attributes</code></p>
-201	ungültigen Benutzernamen vorgegeben
-202	ungültige DB-Position vorgegeben
-203	<p>Dokument <code><\:doc_id></code>s Dokumentart <code><\:doc_type_short></code> enthält nicht das Feld</p> <p><code><\:db-pos></code></p> <p><code>api_error_var[..] = "doc_type_short"</code></p> <p><code>api_error-value[..] = Ermitteltes Dokumentartkürzel zur DokumentID</code></p>
-204	ungültige Zeilennummer vorgegeben

Rückgabewert	Bedeutung
-206	<p>ungültigen Datentyp vorgegeben</p> <p>api_error_var[...] = "doc_type_short"</p> <p>api_error_value[...] = Kürzel der Dokumentart</p> <p>api_error_var[...] = "modifiable"</p> <p>api_error_value[...] = Feld ist modifizierbar (0:nein; 1: ja)</p> <p>api_error_var[...] = "doc_flags"</p> <p>api_error_value[...] = Feldeigenschaften</p> <p>api_error_var[...] = "data_type"</p> <p>api_error_value[...] = Datentyp des Attributes</p>
-215	<p>ungültigen Wert gegenüber Wertevorrat für das Feld <\:db_pos> der Dokumentart</p> <p><\:doc_type_short> angegeben</p> <p>api_error_var[...] = "doc_type_short"</p> <p>api_error_value[...] = Kürzel der Dokumentart</p> <p>api_error_var[...] = "modifiable"</p> <p>api_error_value[...] = Feld ist modifizierbar (0:nein; 1: ja)</p> <p>api_error_var[...] = "doc_flags"</p> <p>api_error_value[...] = Feldeigenschaften</p> <p>api_error_var[...] = "data_type"</p> <p>api_error_value[...] = Datentyp des Attributes</p>
-216	<p>ungültigen Wert gegenüber Plausibilitätskriterien für das Feld <\:db_pos> der Dokumentart</p> <p><\:doc_type_short> angegeben</p> <p>api_error_var[...] = "doc_type_short"</p> <p>api_error_value[...] = Kürzel der Dokumentart</p> <p>api_error_var[...] = "modifiable"</p> <p>api_error_value[...] = Feld ist modifizierbar (0:nein; 1: ja)</p> <p>api_error_var[...] = "doc_flags"</p> <p>api_error_value[...] = Feldeigenschaften</p> <p>api_error_var[...] = "data_type"</p> <p>api_error_value[...] = Datentyp des Attributes</p>
-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
	<p>api_error_var[...] = "doc_permission"</p> <p>api_error_value[...] = Berechtigungen auf das Dokument</p>
-501	Fehler beim Bestimmen der Dokumentart des Dokumentes <\:doc_id>

Rückgabewert	Bedeutung
-515	<p>Feld <\:db_pos> darf nicht geändert werden in Dokumentart <\:doc_type_short>"</p> <p>api_error_var[...] = "doc_type_short"</p> <p>api_error_value[...] = Kürzel der Dokumentart</p> <p>api_error_var[...] = "modifiable"</p> <p>api_error_value[...] = Feld ist modifizierbar (0:nein; 1: ja)</p> <p>api_error_var[...] = "doc_flags "</p> <p>api_error_value[...] = Feldeigenschaften</p> <p>api_error_var[...] = "data_type "</p> <p>api_error_value[...] = Datentyp des Attributes</p>
-516	<p>Fehler beim Ändern des Attributes; nur @dmrowcount Zeilen in der DB geändert</p> <p>api_error_var[...] = "doc_type_short"</p> <p>api_error_value[...] = Kürzel der Dokumentart</p> <p>api_error_var[...] = "modifiable"</p> <p>api_error_value[...] = Feld ist modifizierbar (0:nein; 1: ja)</p> <p>api_error_var[...] = "doc_flags "</p> <p>api_error_value[...] = Feldeigenschaften</p> <p>api_error_var[...] = "data_type "</p> <p>api_error_value[...] = Datentyp des Attributes</p>
< -9500	<p>Fehler in kundenspezifischer Hook-Funktion hook_upd_attrib</p> <p>api_error_var[...] = "hook_return_code"</p> <p>api_error_value[...] = Rückgabewert des Hooks</p>

Anmerkung

verfügbar ab d.3 Version 5.5.1.

Warnung

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.

Anmerkung

```

// -----
// Beispiel:
// Aktualisierung eines Attributes
// -----
// Variablendeklaration
    vars h_ReturnValue      // Fehlercode
vars h_DocIdRef = "Y000050257" // Dokumenten-ID
vars h_DB_PosRef = 2        // Datenbankposition des
betreffenen

                                // 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[]
vewenden,

                                // 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.

Anmerkung

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

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

api_doc_field[h_DB_PosRef] = ""

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

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

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

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

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

```

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

Anmerkung

```

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

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

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

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

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

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

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

    if (api_single_info != "")
    {

```

```

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

```

attribute_update_all

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

Parameter	Beschreibung
<code>doc_id</code>	Dokument-ID, für die die Attribute geändert werden; wird nichts angegeben, wird die ID aus dem Kontext verwendet Pflichtfeld
<code>user_name</code>	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 <code><\:db_pos></code> in der Dokumentart <code><\:doc_type_short></code> angegeben <code>api_error_var[..] = "doc_type_short"</code> <code>api_error_value[..] = Kürzel der Dokumentart</code> <code>api_error_var[..] = "db-position"</code> <code>api_error_value[..] = <Position> - fehlender Wert für Pflichtfeld</code>
-201	ungültigen Benutzernamen vorgegeben
-215	ungültigen Wert gegenüber Wertevorrat für das Feld <code><\:db_pos></code> der Dokumentart <code><\:doc_type_short></code> angegeben <code>api_error_var[..] = "doc_type_short"</code> <code>api_error_value[..] = Kürzel der Dokumentart</code> <code>api_error_var[..] = "db-position"</code> <code>api_error_value[..] = <Position> - ungültiger Wert für Wertevorrat</code>

Rückgabewert	Bedeutung
-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
-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

Anmerkung

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.

Anmerkung

```
// -----  
// Beispiel A:  
// Aktualisierung der Attributfelder 1 und 60 bei  
// gleichzeitigem Verlust anderer Attributwerte  
// -----  
  
// Variablendeklaration  
vars h_ReturnValue           // Fehlercode  
vars h_DocIdRef = "D000000153" // Dokumenten-ID  
  
// leert den Dokument-Kontext der Server API  
call api_clear_document_context ()  
  
// füllen des API-Dokument-Kontextes mit den neuen Attributwerten  
api_doc_field[1] = "42"  
api_doc_field_60[1] = "609"  
api_doc_field_60[2] = "607"  
  
// setzt die Attribute des Dokuments auf die Werte des  
// API-Dokument-Kontextes  
h_ReturnValue = api_function ("attribute_update_all",  
h_DocIdRef, "")  
  
// Fehlerbehandlung  
if (h_ReturnValue != 0)  
{  
    call api_log_error ("Fehlernummer :h_ReturnValue wurde  
        zurückgeliefert!")  
}
```

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.

Anmerkung

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

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

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

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

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

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

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

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

attribute_get_valid_values

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

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.

Anmerkung

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

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

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

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

attribute_test_value

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 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
-216	ungültigen Wert gegenüber Plausibilitätskriterien für das Feld <\:db_pos> 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 api_error_var[..] = "new_value" api_error_value[..] = ggf. aufbereiteter Wert
-303	Benutzer hat kein Zugriffsrecht auf das Dokument

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.

Anmerkung

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

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

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

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

attribute_update_begin**Anmerkung**

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.

Warnung

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

Parameter	Beschreibung
<code>doc_id</code>	Dokument-ID des Dokuments, dessen Attribute geändert werden sollen
<code>user</code>	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 <code>doc_id</code> angegeben
-304	Benutzer hat kein Recht die Attribute zu ändern
-640	zu viele rekursive Attributupdates
-641	unbekannte <code>doc_id</code> vorgegeben
-650	es wird bereits eine Attributaktualisierung über Server-API ausgeführt (Paar <code>attribute_update_begin/_execute</code>)

`attribute_update_execute`

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

`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	Dokumentartkürzel nicht vorgegeben
-201	ungültigen Benutzernamen vorgegeben
-205	ungültiges Dokumentartkürzel 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 <code>api_error_var[..] = "doc_permission"</code> <code>api_error_value[..] = Berechtigungen auf das Dokument</code>
-505	Fehler beim Speichern der Attribute
< -9500	Fehler in kundenspezifischer Hook-Funktion <code>hook_upd_attrib</code> <code>api_error_var[..] = "hook_return_code"</code> <code>api_error_value[..] = Rückgabewert des Hooks</code>

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

Anmerkung

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

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

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

h_DocIdRef, "")

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

document_delete

Funktion löscht ein Dokument aus dem d.3-System.

Parameter	Beschreibung
reason	Löschgrund, 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 optional, default: 0
del_file_always	Nutzdatei unabhängig vom Status löschen 0=nur Nutzdateien in Bearbeitung/Prüfung löschen 1=Nutzdatei in jedem Fall löschen optional, default:0
doc_id	ID des Dokuments, das gelöscht werden soll; wird nichts übergeben, wird die ID des Kontexts genutzt Pflichtfeld, falls nicht im Kontext (api_doc_id) angegeben
user_name	Benutzer, in dessen Namen das Dokument gelöscht werden soll; wird nichts angegeben, wird der ausführende Benutzer verwendet optional

Parameter	Beschreibung
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

Anmerkung
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 übergeben noch im Kontext gesetzt
-120	Löschgrund 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	Benutzer hat kein Recht das Dokument zu löschen <code>api_error_var[..] = "doc_permission"</code> <code>api_error_value[..] = Berechtigung</code>
-319	keine Lizenz, um privilegiert zu löschen
-320	Benutzer hat kein Recht, privilegiert zu löschen
<-9500	Fehler in kundenspezifischer Hook-Funktion <code>hook_delete</code> <code>api_error_var[..] = "hook_return_code"</code> <code>api_error_value[..] = Rückgabewert des Hooks</code>

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.

Anmerkung

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

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

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

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

document_find_ids

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-Variablen	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

Anmerkung

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

Anmerkung

Diese Funktion stellt nur eine rudimentäre Suchfunktionalität zur Verfügung.

Für Abbildung von komplexeren Such-Szenarien (mit Volltext, Rechteprüfung oder Wildcards) verwenden Sie bitte die `docSearch`-Funktionen, die in der d.3 hook-Dokumentation beschrieben werden.

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
-2	Funktion wurde falsch aufgerufen
-121	keine Suchkriterien vorgegeben oder nur die Dummy-Dokumentart

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.

Anmerkung

```

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

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

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

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

```

document_get_permission

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

Parameter	Bedeutung
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	ID des Dokuments, dessen Berechtigungen abgefragt werden; wird nichts übergeben, wird die ID des Kontextes verwendet Pflichtfeld – entweder als Parameter oder im Kontext als <code>api_doc_id</code>
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

Anmerkung

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.

Anmerkung

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

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

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

h_UserName )

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

document_publish_for_web

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

Parameter	Beschreibung
publish	0: Dokument nicht web-veröffentlich 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

Anmerkung

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.

Anmerkung

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

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

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

Im Fehlerfall wird die Fehlernummer zurückgegeben.

Anmerkung

```
// -----
// Beispiel 2:
// Dokument markieren als „nicht-web-veröffentlicht“
// -----
// Variablendeklaration
vars h_ReturnValue                               // Fehlercode
vars h_DokID = "Y000005174"                     // betroffenes Dokument

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

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

document_dependent_add

Anmerkung

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. Die zugehörige Datei darf dabei nicht im neuen Dokumentenbaum erzeugt werden, sondern muss am besten im temporären Verzeichnis liegen.

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 Dokument-ID 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

Legt ein abhängiges Dokument ab bzw. überschreibt ein existierendes.

Anmerkung

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

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

h_DocStatus, \

h_DocExt, \

                               h_DokID)

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

document_dependent_delete

Löscht ein abhängiges Dokument.

Parameter	Beschreibung
doc_status	Status (B, P, F), in dem das zu löschende 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 Dokument-ID vorgegeben
-139	keine doc_ext angegeben

Rückgabewert	Bedeutung
-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

Löscht ein abhängiges Dokument.

Anmerkung

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

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

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

document_get_file_path

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
<code>doc_id</code>	Stammdokument für das der Pfad ermittelt werden soll.
<code>status</code>	Status des Dokuments: "Bearbeitung", "Prüfung", "Freigabe" oder "Archiv"
<code>archive_index</code>	Index der Archiv-Version erforderlich für <code>status = "Archiv"</code>
<code>user_group</code>	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
-126	weder Status übergeben noch im Kontext gesetzt

Anmerkung

verfügbar ab d.3 Version 6.1.1

document_start_lifetime

Setzt die Lebensdauer eines Dokuments.

Parameter	Beschreibung
<code>doc_id</code>	Dokument-ID des Dokuments
<code>overwrite_old_date</code>	Überschreiben des aktuellen Wertes (default: 1)
<code>life_time_days</code>	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 <code>doc_id</code> angegeben
-621	für die Dokumentart ist nicht vorgesehen die Lebenszeit zu ändern und <code>overwrite_old_date</code> mit 0 übergeben
-622	Datenbank-Fehler; siehe Logviewer für Einzelheiten

Setzt die Lebensdauer des Dokuments:

Anmerkung

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

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

Rückgabewert	Bedeutung
-115	für Status Archiv keinen Index der Archiv-Version vorgegeben
-223	ungültigen Status angegeben (muss F oder A sein)
-623	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

Anmerkung

Diese Funktion ist ab d.3 Version 6.3 verfügbar.

document_send_to_dsearch

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 not 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

add_inherit_doc_rights

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)

Parameter	Beschreibung
<code>tstamp_expire</code>	Zeitstempel für Ablaufzeitpunkt (Format DD.MM.YYYY - hh:mm:ss); leer, wenn <code>right_flags[2] = 2</code>

Rückgabewert	Bedeutung
0	alles OK
3	fehlerhafte oder fehlende Parameter
120	Benutzer (<code>granter</code> bzw. <code>grantee</code>) existieren nicht
450	<code>granter</code> darf keine Rechte vererben
451	<code>granter</code> hat nicht die Rechte, die vererbt werden sollen
452	<code>grantee</code> hat bereits die Rechte auf das Dokument
9000	Datenbank-Fehler

Anmerkung

```
//-----  
  
-----//  
  
// 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 Doku-  
// ment  
  
// 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

```

remove_inherit_doc_rights

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

1.4.3. Akte

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_noundapi_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: asynchron, 1: synchron) 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

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.

Anmerkung

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

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

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

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

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

1.4.4. Verknüpfungen**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.

Anmerkung

`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>

Parameter	Beschreibung
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 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

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

Anmerkung

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

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

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

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

link_get_children

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.

Anmerkung

`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

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

Tritt ein Fehler auf, wird dieses ausgegeben.

Anmerkung

```

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

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

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

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

```

link_documents

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

Parameter	Beschreibung
use_folderplan	<p>Wenn doc_id_parent nicht angegeben ist</p> <p>0: LIN002-JOB anlegen mit den Erkennungsattributen in den api_doc_field's und api_doc_type_short und den Vererbungsattributen in ernie_inherit_field_name_1..4</p> <p>1: Aktenplan für das Dokument verwenden</p> <p>optional – default: 0</p>

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

Anmerkung

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“ dagegegen anhand der Aktenbildung vorgenommen.

Anmerkung

```
// -----  
// Beispiel A:  
// Direkte Verlinkung durch Angabe der übergeordneten doc_id  
// -----  
  
// Variablendeklaration  
vars h_ReturnValue                // Fehlercode  
vars h_DocIdRef = "Y000050257"    // untergeordnete  
vars h_DocIdRefParent = "Y000050501" // übergeordnete  
Dokumenten-ID  
  
// Funktionsaufruf:  
// direkte Verlinkung durch Angabe des parent  
h_ReturnValue = api_function ("link_documents",  
h_DocIdRefParent, \  
                                h_DocIdRef, \  
                                "", \  
                                0, \  
                                "")  
  
// Ergebnisausgabe und Fehlerbehandlung  
if (h_ReturnValue == 0)  
{  
    call api_log_info("Verknüpfung erfolgreich erstellt!")  
}  
else  
{  
    call api_log_error("Fehlernummer :h_ReturnValue wurde  
zurückgeliefert!")  
}
```

Anmerkung

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

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

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

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

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

link_exists

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 übergeordneten bzw. untergeordneten Dokuments
-219	über- und untergeordnete Dokument-ID sind identisch

Testen, ob zwei Dokumente verknüpft sind:

Anmerkung

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

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

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

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

link_delete

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 <code>api_doc_id</code>
doc_id_child	ID des untergeordneten Dokuments Wird die <code>doc_id</code> 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 <code>api_doc_id</code>
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

Rückgabewert	Bedeutung
-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 für über- bzw. untergeordnet
-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 <code>api_error_var[..] = "right_level"</code> <code>api_error_value[..] = "<Rechtestufe>"</code> <code>api_error_var[..] = "right_link"</code> <code>api_error_value[..] = "<Recht zum Aktenverknüpfen>"</code>
-522	die Dokumente sind nicht auf diese Weise verknüpft
<-9500	Fehler in kundenspezifischer Hook-Funktion <code>hook_unlink</code> <code>api_error_var[..] = "hook_return_code"</code> <code>api_error_value[..] = Rückgabewert des Hooks</code>

Anmerkung

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.

Anmerkung

```

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

```

1.4.5. Verzeichnisse**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

Anmerkung

verfügbar ab d.3 Version 6.0.1 Hotfix 1

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

Anmerkung

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

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

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

directory_delete

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

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

Anmerkung

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

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

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

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

directory_exist

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

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

Anmerkung

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

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

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

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

directory_list

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

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

Anmerkung

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

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

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

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

directory_count_files

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, d.velop-Support kontaktieren
-160	Verzeichnis nicht vorgegeben
-583	Fehler beim Zählen der Dateien

Anzahl der Dateien in einem Ordner.

Warnung

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

Anmerkung

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

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

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

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

1.4.6. Status

document_block

Diese Funktion sperrt oder entsperrt ein Dokument.

Parameter	Beschreibung
block	0: Dokument entsperren 1: Dokument sperren (default) optional
user_name	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 <code>api_dok_id</code>

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 <code>api_error_var[.] = "doc_permission"</code> <code>api_error_value[.] = "<Berechtigungen>"</code>
-507	Version in Freigabe ist bereits gesperrt <code>api_error_var[.] = "doc_permission"</code> <code>api_error_value[.] = "<Berechtigungen>"</code>
-508	Version in Freigabe ist nicht gesperrt <code>api_error_var[.] = "doc_permission"</code> <code>api_error_value[.] = "<Berechtigungen>"</code>
< -9500	Fehler in kundenspezifischer Hook-Funktion <code>hook_release_...</code> <code>api_error_var[.] = "hook_return_code"</code> <code>api_error_value[.] = Rückgabewert des Hooks</code>

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

Das zweite Beispiel entspermt das Dokument mit der ID `Y000050257` und gibt ggf. eine Fehlermeldung aus.

Anmerkung

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

Anmerkung

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

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

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

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

document_verify

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

Rückgabewert	Bedeutung
-307	Benutzer hat kein Recht, dieses Dokument zu prüfen <pre>api_error_var[...] = "doc_permission" api_error_value[...] = "<Berechtigungen>"</pre>
-509	Version im Status Prüfung ist bereits geprüft <pre>api_error_var[...] = "doc_permission" api_error_value[...] = "<Berechtigungen>"</pre>
-510	es existiert keine Version im Status Prüfung <pre>api_error_var[...] = "doc_permission" api_error_value[...] = "<Berechtigungen>"</pre>
< -9500	Fehler in kundenspezifischer Hook-Funktion hook_release_... <pre>api_error_var[...] = "hook_return_code" api_error_value[...] = Rückgabewert des Hooks</pre>

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

Anmerkung

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

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

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

document_transfer**Anmerkung**

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 <div style="border: 1px solid black; background-color: #e6f2ff; padding: 5px; margin: 10px 0;"> <p>Anmerkung geändert für Version 7.0: kein Pflichtfeld mehr, wenn im Parameter destination „Prüfung“ angegeben wird</p> </div>
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 d.3 async dann später durchführt (0: sofort (default); 1: später durch d.3 async) optional – default: 0
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>"

Rückgabewert	Bedeutung
-113	<p>Gruppenname nicht vorgegeben</p> <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> <div style="background-color: #e6f2ff; padding: 10px; margin-top: 10px;"> <p>Anmerkung neu in d.3 Version 7.0: Fehler -113 tritt nicht mehr im Status „Prüfung“ auf.</p> </div>
-114	Zielstatus nicht vorgegeben
-115	<p>Archivindex nicht vorgegeben</p> <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>
-116	<p>Änderungstext nicht vorgegeben</p> <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	<p>ungültigen Gruppenname vorgegeben</p> <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
-214	<p>ungültigen Zielstatus vorgegeben</p> <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>
-511	dieser Transfer ist nicht erlaubt
	<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>
-512	Es gibt eine neuere Version; um diese zu überschreiben muss sie erst in die Bearbeitung des Benutzers <\:user_group> übertragen werden
	<pre>api_error_var[..] = "doc_permission" api_error_value[..] = "<Berechtigungen>"</pre>
-513	Fehler <\:job_error> beim Anlegen des Jobs

Rückgabewert	Bedeutung
-514	<p>Fehler beim Vergleichen der Nutzdateien der alten und der neuen Version</p> <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>
< -9500	<p>Fehler in kundenspezifischer Hook-Funktion <code>hook_transfer_...</code></p> <pre>api_error_var[..] = "hook_return_code" api_error_value[..] = Rückgabewert des Hooks</pre>

Im ersten Beispiel wird das Dokument mit der angegebenen Dokument-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.

Anmerkung

```

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

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

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

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

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

```

Anmerkung

```
// -----  
// Beispiel B:  
// Ein Dokument in die Prüfung verschieben  
// -----  
  
// Variablendeklaration  
vars h_ReturnValue           // Fehlercode  
vars h_DocIdRef = "Y000050257" // Dokumenten-ID  
vars h_StatusRef = "Prüfung"  // Ziel-Status  
vars h_GroupRef = "test"     // Gruppenname  
vars h_asynchronus = 0       // Sofortiger Statustransfer  
  
// Funktionsaufruf  
h_ReturnValue = api_function ("document_transfer", h_StatusRef, \  
                             h_GroupRef, \  
                             "", \  
                             h_asynchronus,  
 \  
                             "", \  
                             "", \  
                             h_DocIdRef)  
  
// Ergebnisausgabe und Fehlerbehandlung  
if (h_ReturnValue == 0)  
{  
    call api_log_info("Dokument erfolgreich in den gewünschten  
Status \  
                             verschoben!")  
}  
else  
{  
    call api_log_error("Fehlernummer \  
                             :h_ReturnValue wurde zurückgeliefert!")  
}
```

Anmerkung

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

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

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

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

1.4.7. Notizen**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

Rückgabewert	Bedeutung
-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 hook_transfer_... <code>api_error_var[..] = "hook_return_code"</code> <code>api_error_value[..] = Rückgabewert des Hooks</code>

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.

Anmerkung

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

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

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

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

note_add_string

Diese Funktion fügt eine Zeile an die Notizdatei des Dokumentes 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: <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
-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 <code>hook_transfer_...</code> <code>api_error_var[..] = "hook_return_code"</code> <code>api_error_value[..] = Rückgabewert des Hooks</code>

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.

Anmerkung

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

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

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

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

1.4.8. Wiedervorlage**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

Parameter	Beschreibung
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
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

Anmerkung
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, d.velop-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"

Rückgabewert	Bedeutung
-208	ungültiges Datum vorgegeben <pre>api_error_var[..] = "date_activate_holdfile" api_error_value[..] = Aktivierungsdatum api_error_var[..] = "date_acknowledge_until" 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 <pre>api_error_var[..] = "recipient" / "sender" api_error_value[..] = "<Empfänger-/Sendername> - no access right ?"</pre>
-302	Fehler beim Ermitteln der Benutzerberechtigungen
-303	Benutzer hat kein Zugriffsrecht auf das Dokument <pre>api_error_var[..] = "recipient" / "sender" api_error_value[..] = "<Empfänger-/Sendername> - has no access right for document"</pre>
-315	kein Gruppenmitglied hat Zugriffsrecht auf das zu versendende Dokument
-316	Empfänger hat kein Zugriffsrecht; Sender kann Zugriffsrecht aber vererben
	Anmerkung neuer Rückgabewert in d.3 Version 7.0
-317	Sender hat kein Schreibrecht, kann aber Rechte vererben
	Anmerkung neuer Rückgabewert in d.3 Version 7.0.
-318	Empfänger hat kein Schreibrecht, kann aber Rechte erben
-450	Sender hat kein Recht zur Rechtevererbung
-451	der User besitzt nicht die Rechte, die er vererben will
-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 <pre>api_error_var[..] = "recipient" api_error_value[..] = "<Empfängername> - checked out" api_error_var[..] = "checkout_text" api_error_value[..] = Abwesenheitstext</pre>

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 „Y000050527“ 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.

Anmerkung

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

vars h_Subject = "Test von hold_file_send" // Betreff
vars h_DocIdRef = "Y000050527" // 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!")
}
}
```

Anmerkung

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

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

vars h_IgnoreCheckout = 1                        // Auch bei
Abwesenheit

                                                // des
                                                // Empfängers versenden

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

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

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

Anmerkung

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

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
recipient	Empfänger der Wiedervorlage optional
doc_id	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>
user_name	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
-302	Fehler beim Ermitteln der Benutzerberechtigungen
-303	Benutzer hat kein Zugriffsrecht auf das Dokument

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.

Anmerkung

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

Anmerkung

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

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

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

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

hold_file_delete

Funktion quittiert empfangene Wiedervorlagen und löscht gesendete Wiedervorlagen.

Anmerkung

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>	ID der Wiedervorlagen-Kette, zu der der Eintrag gehört Pflichtfeld
<code>sent_received</code>	gesendete (1) oder empfangene (2) Wiedervorlage verarbeiten (optional - default = 2)

Parameter	Beschreibung
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

Anmerkung

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.

Anmerkung

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

hold_file_get_document_info

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
api_hold_file_type	Typ (W=Workflow)

Parameter	Beschreibung
sender	Sender Pflichtfeld

Parameter	Beschreibung
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: <code>api_doc_id</code>
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, d.velop-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

Anmerkung

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.

Anmerkung

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

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

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

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

1.4.9. Workflow**workpath_get_document_info**

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)

Globale Variable	Beschreibung
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 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
-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

workpath_end_document

Funktion entfernt Dokument aus dem Leitweg.

Anmerkung

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
> 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
-523	das Dokument ist gerade nicht in einem Leitweg aktiv
-524	der Workflow über Leitwege wird nicht unterstützt für dieses Repository

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

Anmerkung

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

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

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

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

workpath_start_document

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

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.

Anmerkung

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

// Variablendeklaration
vars h_ReturnValue // Fehlercode
vars h_DocIdRef = "D000000436" // Dokumenten-ID
vars h_WflId = "000000000000MT059IT1100CP200" // ID des zu
startenden // Workflows

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

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

workpath_go_to_next_step

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

Anmerkung

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 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 Repository
-527	Fehler beim Bestimmen der aktuellen Position im Leitweg

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.

Anmerkung

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

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

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

Anmerkung

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

// Variablendeklaration
vars h_ReturnValue           // Fehlercode
vars h_DocRefID = "D000000284" // Dokumenten-ID
vars next_step_ref_id = "00000000000000MT089ITI1OCP200"
// ID des nächsten Workflowschritts

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

1.4.10. Dokumentart**document_type_get_long**

Diese Funktion bestimmt den Dokumentart-Langtext wahlweise zu einer Dokument-ID 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
<code>doc_type_short</code>	Kürzel, dessen Langtext abgefragt wird optional
<code>doc_id</code>	Dokument, dessen Dokumentartlangtext abgefragt wird optional - unter Umständen wird <code>api_doc_id</code> verwendet

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

Rückgabewert	Bedeutung
-101	weder Dokument-ID übergeben noch im Kontext gesetzt
-501	Fehler beim Bestimmen der Dokumentart des Dokuments <\:doc_id>

Das erste Beispiel gibt den Dokumentartnamen der per Dokument-ID 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.

Anmerkung

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

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

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

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

Anmerkung

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

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

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

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

document_type_get_short

Diese Funktion bestimmt das Dokumentart-Kürzel wahlweise zu einer Dokument-ID 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
-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

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

Das zweite Beispiel ruft die Funktion nicht mit einer Dokument-ID, sondern mit dem Dokumentartnamen auf.

Anmerkung

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

Anmerkung

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

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

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

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

1.4.11. Wertemengen**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.

Warnung

Die Funktion fragt nur statische Wertemengen ab.

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

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

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.

Anmerkung

```

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

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

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

// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
  if (api_single_info > 0)
  {
    for h_Count = 1 while h_Count <= api_single_info step 1
    {
      // Ausgabe des Ergebnisses mit Hilfe des Arrays
      // d3server_value_char,
      // da die Wertemenge im Beispiel alphanumerisch ist
      call api_log_info("Eintrag :h_Count: \

:d3server_value_char_allowed[:h_Count]")
    }
  }
  else
  {
    call api_log_info("Die Wertemenge enthält keine Einträge.
")
  }
}
else
{
  call api_log_error("Fehlernummer :h_ReturnValue wurde
zurückgeliefert! ")
}

```

Anmerkung

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

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

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

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

predefined_values_test

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
-2	Funktion wurde falsch aufgerufen
-119	weder Wertemengen-ID noch Wertemengen-Name vorgegeben
-218	ungültige Wertemengen-ID oder Wertemengen-Name vorgegeben

Anmerkung

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.

Anmerkung

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

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

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

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

1.4.12. Benutzer

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
user_type	1 Nicht-System-Benutzer (default)
	2 System-Benutzer (d3_server, ...)
	3 System- und Nicht-System-Benutzer

Anmerkung

Parameter verfügbar ab d.3 Version 6.0 HF 3

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

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

Anmerkung

```

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

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

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

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

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

```

user_is_system

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

Anmerkung

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.

Anmerkung

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

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

user_get_groups

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.

Anmerkung

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

Anmerkung
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

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.

Anmerkung

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

Anmerkung

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

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

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

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

user_get_verifier_groups

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

Rückgabewert	Bedeutung
-2	Funktion wurde falsch aufgerufen
-201	ungültigen Benutzernamen vorgegeben

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.

Anmerkung

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

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

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

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

user_get_rolls

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
user_name	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

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.

Anmerkung

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

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

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

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

user_get_doc_types

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; d.velop-Support kontaktieren

Anmerkung

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.

Anmerkung

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

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

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

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

user_get_data

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)

Globale Variable	Beschreibung
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 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

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.

Anmerkung

```

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

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

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

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

```

```

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

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

```

user_validate_password

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

Anmerkung

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).

Anmerkung

```
// -----
// Beispiel:
// Prüfen, ob Benutzername und Passwort zusammenpassen
// -----
// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_UserName = "armin"   // Benutzername
vars h_Password = "frikadelle" // Passwort
// Funktionsaufruf
h_ReturnValue = api_function ("user_validate_password",
h_Password, \
                        h_UserName)
// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
  call api_log_info ("Kombination aus Benutzername und Passwort
gültig!")
}
else
{
  call api_log_info ("Kombination aus Benutzername und Passwort
ungültig!")
}
```

user_get_favourites

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.

Anmerkung

Es werden maximal 500 Favoriten zurückgegeben.

Parameter	Beschreibung
user_name	Benutzer, zu dem die Favoriten gelesen werden sollen

Rückgabewert	Bedeutung
0	alles OK

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

Dokument-IDs der Favoriten eines Benutzers ausgeben.

Anmerkung

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

user_get_sid_rights

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

user_checkout

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 (<code>granter</code> bzw. <code>grantee</code>) existieren nicht
420	<code>call_user</code> ist nicht der Vorgesetzte von <code>check_user</code>
423	der ausgewählte Vertreter ist zur Zeit abwesend gemeldet
9000	Datenbank-Fehler (siehe LogViewer)

user_get_superior

Funktion zur Ermittlung der Vorgesetzten eines Benutzers.

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

Globale Variable	Bedeutung
<code>api_single_info</code>	Anzahl Elemente der folgenden Feldes
<code>api_users[]</code>	Liste der Vorgesetzten

Rückgabewert	Bedeutung
0	alles OK
120	Daten des Benutzers nicht im Cache gefunden

1.4.13. Gruppen

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
<code>language</code>	Sprache zur Ermittlung der Gruppen-Namen (Standard = 049)

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

Anmerkung

neu ab d.3 Version 7.0:

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

Anmerkung

```

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

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

// Funktionsaufruf
h_ReturnValue = api_function ("group_get")

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

```

group_get_members

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 black; padding: 5px; background-color: #e6f2ff;"> <p>Anmerkung neu in d.3 Version 7.0:</p> <p>Parameter:</p> <p><code>group_name</code> kann jetzt auch die Gruppen-ID sein</p> </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

Rückgabewert	Bedeutung
-113	keinen Gruppennamen vorgegeben
-213	ungültigen Gruppennamen vorgegeben

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.

Anmerkung

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

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

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

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

Anmerkung

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

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

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

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

group_get_verifiers**Anmerkung**

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.

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

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.

Anmerkung

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

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

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

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

group_get_doc_types

Anmerkung

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

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

Die Namen der Dokumentenarten 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.

Anmerkung

```

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

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

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

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

```

1.4.14. Rollen**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

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.

Anmerkung

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

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

// Funktionsaufruf
h_ReturnValue = api_function ("roll_get")

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

roll_get_names

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

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.

Anmerkung

```

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

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

// Funktionsaufruf
h_ReturnValue = api_function ("roll_get_names")

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

```

roll_get_users

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.

Anmerkung

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

Rückgabewert	Bedeutung
-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

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.

Anmerkung

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

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

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

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

Anmerkung

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

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

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

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

roll_get_data

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

Globale Variable	Beschreibung
api_rolld_id	ID der Rolle
api_rolld_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

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_date` 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 Variablendeklaration und der Funktionsaufruf auskommentiert.

Anmerkung

```

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

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

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

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

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

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

```

```

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

```

1.4.15. Restriktionsmengen

d3set_add_filter

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
1513	die Restriktionsmenge existiert nicht
9000	Datenbank-Fehler

d3set_remove_filter

Entfernt einen Filter von einer bestehenden Restriktionsmenge.

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 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.

Rückgabewert	Bedeutung
1512	kein Schreibrecht auf das Mengen-Dokument.
1504	Länge des Mengen-Filters der Restriktionsmenge überschreitet das Maximum von 150.000 Zeichen
9000	Datenbank-Fehler

d3set_get_filter

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.

d3set_get_sets

Anmerkung

neu in d.3 Version 8.0

Sucht über bestimmte Kriterien nach Restriktionsmengen-Zuordnungen und liefert diese in einer Tabelle zurück.

Als Suchkriterien können der Name der Restriktionsmenge (`set_name`), ein Objekt (`object_id`: Benutzer, Gruppe, Organisationseinheit) und die Objekt-Klasse (`object_class_id`) verwendet werden.

Wird `object_id` gesetzt, muss zwingend auch `object_class_id` gesetzt sein.

Parameter	Beschreibung
<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>object_class_id</code>	Objekt-Klasse (0: keine Objekt-Zuordnung, 1: d.3 Benutzer, 2: d.3 Gruppe, 4: Organisationseinheit)

Die Anzahl der zurückgelieferten Ergebnisse wird in `api_single_info` geschrieben. Es werden maximal 100 Zeilen zurückgegeben.

Folgende Arrays des Kontextes werden gefüllt:

Variable	Beschreibung
<code>api_set_doc_id</code>	Dokument-ID des Restriktionsmengen-Dokuments
<code>api_set_name</code>	Name der Restriktionsmenge
<code>api_set_object_id</code>	Benutzername, Gruppen-ID bzw. Organisationseinheits-ID
<code>api_set_object_class_id</code>	Objekt-Klassen-ID zu Wert aus <code>api_set_object_id</code> (0: keine Objekt-Zuordnung, 1: d.3 Benutzer, 2: d.3 Gruppe, 4: Organisationseinheit)

Rückgabewert	Bedeutung
0	alles ok
>0	Datenbank-Fehler
-1	Parameter ungültig (siehe d.3 Log-Datei)

d3set_remove_set

Anmerkung

neu in d.3 Version

Löscht eine bestehende Restriktions-Menge. Folgende Fälle sind dabei möglich:

Wenn nur der Parameter `set_name` gefüllt ist, werden alle Referenzen dieser Restriktions-Menge, also auch die Zuordnungen innerhalb von Dokumentklassen, entfernt.

Ist zusätzlich zum Parameter `set_name` auch der Parameter `object_id` gefüllt, wird nur die Menge aus dem System entfernt, die dem Objekt (Benutzer oder Gruppe) zugeordnet ist. Die eigentliche Menge bzw. ein Verweis auf die Existenz dieser Menge bleibt bestehen.

Wenn der Parameter `doc_id` gesetzt ist, wird die spezielle Menge entfernt. Dieser Parameter hat Vorrang zu `set_name` und `object_id`.

Parameter	Beschreibung
<code>user_name</code>	d.3-Benutzername, in dessen Name die Änderung durchgeführt werden soll; dieser d.3-Benutzer muss Änderungsrechte auf Restriktionsmengen besitzen
<code>doc_id</code>	Dokument-ID des Restriktionsmengen-Dokuments; 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
3	Pflichtparameter nicht gesetzt
110	Ungültige Dokument-ID vorgegeben
1506	das Mengen-Dokument konnte nicht gelöscht werden (siehe d.3 Log-Datei)
1507	Objektname existiert weder als Benutzer, Gruppe, Tätigkeitsprofil noch als Organisationseinheit
1509	Restriktionsmenge wird noch in einer Dokumentklasse verwendet
1511	keine Berechtigung zum Löschen von Restriktionsmengen
1513	die Restriktionsmenge existiert nicht
9000	Datenbank-Fehler
14020	Fehler beim Eintragen in das Audit-Trail

1.4.16. Tiff-Funktionen

`document_render`

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

Anmerkung

Diese Funktion ersetzt die Funktion `tiff_create`.

Unterschied: `tiff_create` erzeugt nur TIFF; `document_render` erzeugt wahlweise TIFF oder PDF.

Anmerkung

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>	Zielfeld; erforderlich, falls das erzeugte Render-Dokument nicht als abhängige <code>t1</code> oder <code>p1</code> 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

Parameter	Beschreibung
replace_doc	erzeugtes Render-Dokument 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 gerendert / zu dem ein abhängiges t1 bzw. p1 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
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 upd721.jpl 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, d.velop 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 Dokument-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 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
-587	Dokument konnte nicht vom externen Storage bereitgestellt werden
-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

tiff_create

Anmerkung

tiff_create erzeugt nur TIFF.

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

Funktion vertift Dokument oder Datei.

Parameter	Beschreibung
source	Quelldatei; erforderlich, falls nicht das Dokument <code>doc_id</code> (neueste Version) vertifft werden soll
destination	Zieldatei; erforderlich, falls das erzeugte Tiff nicht als abhängige t1 Datei zu <code>doc_id</code> 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 (<code>job_id</code> wird für <code>error_var[.]="job_id"</code> geliefert); diese Option steht nur für das dokumentbasierte Vertiffen zur Verfügung, d.h. <code>doc_id</code> muss vorhanden sein default: 0
replace_doc	Erzeugtes TIF ersetzt das Stammdokument im d.3 (siehe auch Parameter <code>ERLAUBE_STAMM_2_TIFF_KONVERT</code>) 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 Dokument-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 übergeben noch im Kontext gesetzt
-151	Quelldateiname nicht vorgegeben
-152	Zieldateiname nicht vorgegeben
-201	ungültiger Benutzername 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
-543	Fehler in der d.tiff Verbindung
-544	d.tiff 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 <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
<-9500	Fehler in kundenspezifischer Hookfunktion <code>hook_rendition_..</code>

Anmerkung

verfügbar ab d.3 Version 6.0

tiff_concat

Die Funktion hängt ein Tiff-Dokument an ein anderes.

Parameter	Beschreibung
source	Quell-Tiff-Datei
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

document_render_wfl_prot

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

Rückgabe- wert	Bedeutung
1	vertiftes 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

1.4.17. Dateizugriffe

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.

Anmerkung

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>text</code>	String, der in die Datei geschrieben werden soll
<code>mode</code>	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.
<code>count</code>	Anzahl der Zeichen, die mindestens geschrieben werden sollen. Ist <code>text</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

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.

Anmerkung

```

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

// Variablendeklaration
vars h_ResultValue // Fehlercode

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

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

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

```

file_add_line

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.

Anmerkung

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

Parameter	Beschreibung
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, wird APPEND benutzt.
count	Anzahl der Zeichen, die mindestens geschrieben werden sollen. Ist <line> 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

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.

Anmerkung

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

// Variablendeklaration
vars h_ReturnValue // Fehlercode

// Im Pfad müssen ":" und "\" mit vorgestelltem \ ergänzt werden.
vars h_Pfad = "c:\\text.txt"
vars h_Line = "Ich, dieser Text, wurde in die Datei geschrieben.
"

// Funktionsaufruf
h_ReturnValue = api_function ("file_add_line", h_Pfad, h_Line)

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

file_read_line

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
file_name	Dateiname
line	Startzeile
mode	Leserichtung: BOF=vorwärts, EOF=rückwärts
count	Anzahl Zeichen maximal lesen

Rückgabewert	Bedeutung
>=0	alles OK; entspricht Anzahl der gefundenen Zeichen
-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	ungültige Leserichtung vorgegeben
-550	Datei kann nicht zum Lesen geöffnet werden

Anmerkung

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.

file_concat

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

Parameter	Beschreibung
source	Quelldatei, deren Inhalt angehängt werden soll
destination	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

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
-554	Fehler beim Schreiben der Datei

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.

Anmerkung

```
// -----
// Beispiel: Datei kopieren
// -----
// Variablendeklaration
vars h_ReturnValue           // Fehlercode
vars h_Quelle = "c:\\text.txt" // Ausgangs-Datei
vars h_Ziel = "c:\\text2.txt" // Ziel-Datei

// Funktionsaufruf
h_ReturnValue = api_function ("file_copy", h_Quelle, h_Ziel)
// Fehlerbehandlung
if (h_ReturnValue != 0)
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
        zurückgeliefert! ")
}
```

file_delete

Diese Funktion löscht eine Datei.

Parameter	Beschreibung
<code>file_name</code>	Datei, die gelöscht 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
-150	Dateiname nicht vorgegeben
-250	Datei nicht gefunden
-350	Dateizugriff verweigert

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

Anmerkung

```
// -----
// Beispiel: Löschen einer Datei
// -----
// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_Pfad = "c:\\text2.txt" // Dateipfad
// Funktionsaufruf
h_ReturnValue = api_function ("file_delete", h_Pfad)
if (h_ReturnValue != 0)
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
        zurückgeliefert! ")
}
```

file_compare

Diese Funktion vergleicht zwei Dateien.

Parameter	Beschreibung
file_name1	Datei 1
file_name2	Datei 2

Rückgabewert	Bedeutung
4	Dateinhalt unterschiedlich
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

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.

Anmerkung

```

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

// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_Datei1 = "c:\\text.txt" // Datei 1
vars h_Datei2 = "c:\\text2.txt" // Datei 2

// Funktionsaufruf
h_ReturnValue = api_function ("file_compare", h_Datei1, h_Datei2)

// Ergebnisausgabe
if (h_ReturnValue == 4)
{
    call api_log_info("Die Dateien haben einen unterschiedlichen
        Inhalt")
}
if (h_ReturnValue == 3)
{
    call api_log_info("Die Dateien haben unterschiedliche
Dateigrößen")
}
if (h_ReturnValue == 0)
{
    call api_log_info("Die Dateien sind identisch")
}
// Fehlerbehandlung
if (h_ReturnValue < 0)
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
        zurückgeliefert! ")
}
}

```

file_rename

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, d.velop-Support kontaktieren
-151	Quelldateiname nicht vorgegeben
-152	Zieldateiname nicht vorgegeben
-555	Fehler beim Umbenennen der Datei

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

Anmerkung

```
// -----
// Beispiel:
// Umbenennen einer Datei
// -----
// Variablendeklaration
vars h_ReturnValue // Fehlercode
vars h_AlterName = "c:\\text.txt" // Alter Dateiname
vars h_NeuerName = "c:\\text2.txt" // Neuer Dateiname
// Funktionsaufruf
h_ReturnValue = api_function ("file_rename", h_AlterName,
h_NeuerName)

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

file_exist

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

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

Anmerkung

```

/ -----
// Beispiel:
// Prüfen, ob Datei existiert
// -----
// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_Pfad = "c:\\text.txt" // Pfad zur Datei
// Funktionsaufruf
h_ReturnValue = api_function ("file_exist", h_Pfad)
// Ergebnisausgabe
if (h_ReturnValue == 1)
{
    call api_log_info("Die Datei existiert. ")
}
if (h_ReturnValue == 0)
{
    call api_log_info("Die Datei existiert nicht. ")
}
// Fehlerbehandlung
if (h_ReturnValue < 0)
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
        zurückgeliefert! ")
}

```

1.4.18. Stringkonvertierung**string_to_lower**

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

Anmerkung

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

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.

Anmerkung

```
// -----
// Beispiel:
// Zeichenkette in Kleinbuchstaben konvertieren
// -----
// Variablendeklaration
vars h_ReturnValue
           // Fehlercode
vars h_Text = "Anton Berta Caesar Dora Emil"
           // Ursprünglicher String
// Funktionsaufruf
h_ReturnValue = api_function ("string_to_lower", h_Text)
// Ergebnisausgabe
if (h_ReturnValue == 0)
{
    call api_log_info("Konvertierter String:
:api_converted_string")
}
// Fehlerbehandlung
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
zurückgeliefert! ")
}
}
```

string_to_upper

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

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

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.

Anmerkung

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

// Variablendeklaration
vars h_ReturnValue
    // Fehlercode
vars h_Text = "Anton Berta Caesar Dora Emil"
    // Ursprünglicher String

// Funktionsaufruf
h_ReturnValue = api_function ("string_to_upper", h_Text)

// Ergebnisausgabe
if (h_ReturnValue == 0)
{
    call api_log_info("Konvertierter String:
:api_converted_string")
}
// Fehlerbehandlung
else
{
    call api_log_info("Fehlernummer :h_ReturnValue wurde
zurückgeliefert! ")
}
}
```

string_to_number

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 `api_converted_string` geschrieben.

Parameter	Beschreibung
<code>text</code>	String, der konvertiert werden soll
<code>data_type</code>	Datentyp; zulässig sind M=MONEY und N=NUMERIC; 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

string_to_date

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

Parameter	Beschreibung
text	String, der konvertiert werden soll

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

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

Anmerkung

```
// -----
// Beispiel:
// Umwandeln eines String-Datums in ein d.3-Datum
// -----
// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_Datum = "01.01.1972" // Ursprungsdatum als String
// Funktionsaufruf
h_ReturnValue = api_function ("string_to_date", h_Datum)
// Ergebnisausgabe
if (h_ReturnValue == 0)
{
    call api_log_info("Konvertiertes Datum:
:api_converted_string")
}
// Fehlerbehandlung
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
zurückgeliefert! ")
}
}
```

string_split

Diese Funktion splittet einen String an einem Separator auf. Die Teilstrings werden im globalen Array `api_splitted_string[]` gespeichert.

Anmerkung

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
text	String, der aufgesplittet werden soll
separator	Zeichen(kette), an dem / der gesplittet werden soll

Rückgabewert	Bedeutung
>0	Anzahl der Werte im Array

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

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.

Anmerkung

```
// -----
// Beispiel:
// Zeichenkette aufspalten
// -----
// Variablendeklaration
vars h_ReturnValue
    // Anzahl zurückgegebener Werte
vars h_String = "Anton|Berta|Caesar|Dora|Emil"
    // zu teilender String
vars h_Splitter = "|"
    // Trenn-Zeichen
vars h_Count
    // Zähler für Schleife
// Funktionsaufruf
h_ReturnValue = api_function ("string_split", h_String,
h_Splitter)

// Ergebnisausgabe
if (h_ReturnValue >= 0)
{
    call api_log_info("Der String wurde in :h_ReturnValue
Strings geteilt.")
    for h_Count = 1 while( h_Count <= h_ReturnValue ) step 1
    {
        call api_log_info("Teilstring: :(apiSplittedString
[:h_Count])")
    }
}
// Fehlerausgabe
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
zurückgeliefert!")
}
}
```

string_cut_leading_char

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

Anmerkung

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

In diesem Beispiel werden von dem String „AAAAAABCDEFG“ die führenden „A“s entfernt.

Der neue String – „BCDEFG“ – wird durch die Funktion in der globalen Variable `api_converted_string` gespeichert und ausgegeben.

Anmerkung

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

// Variablendeklaration
vars h_ReturnValue           // Fehlercode
vars h_Text = "AAAAAABCDEFG" // Ursprünglicher String
vars h_Char = "A"           // Abzutrennendes Anfangszeichen

// Funktionsaufruf
h_ReturnValue = api_function ("string_cut_leading_char", h_Text,
                             h_Char)

// Ergebnisausgabe
if (h_ReturnValue >= 0)
{
    call api_log_info("Konvertierter String:
:api_converted_string")
    call api_log_info("Es wurden :h_ReturnValue Zeichen entfernt.
")
}
// Fehlerausgabe
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
zurückgeliefert! ")
}
}
```

string_fill_leading_char

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

Parameter	Beschreibung
<code>text</code>	String, der aufgefüllt 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

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.

Anmerkung

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

// Variablendeklaration
vars h_ReturnValue // Fehlercode
vars h_Text = "12345" // Ursprünglicher String
vars h_Char = "0" // Zeichen, welches eingefügt wird
vars h_Length = 10 // Ziel-Länge des neuen Strings

// Funktionsaufruf
h_ReturnValue = api_function ("string_fill_leading_char", \
                             h_Text, h_Char, h_Length)

// Ergebnisausgabe
if (h_ReturnValue >= 0)
{
    call api_log_info("Konvertierter String:
:api_converted_string")
    call api_log_info("Es wurden :h_ReturnValue Zeichen
eingefügt")
}
// Fehlerausgabe
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
zurückgeliefert! ")
}
}
```

string_fill_trailing_char

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 als Füllzeichen 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

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.

Anmerkung

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

// Variablendeklaration
vars h_ReturnValue           // Fehlercode
vars h_Text = "AntonBertaCaesar" // Ursprünglicher String
vars h_Length = 25           // Ziel-Länge des neuen
Strings

// Funktionsaufruf
// ohne Angabe des Füll-Zeichens, daher wird das Leerzeichen
verwendet
h_ReturnValue = api_function ("string_fill_trailing_char",
h_Text, \
                               "", h_Length)

// Ergebnisausgabe
if (h_ReturnValue >= 0)
{
    call api_log_info("Konvertierter String:
:api_converted_string")
    call api_log_info("Es wurden :h_ReturnValue Zeichen
angehängt. ")
}
// Fehlerausgabe
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
zurückgeliefert! ")
}
}
```

string_replace

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

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

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.

Anmerkung

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

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
>0	erste gefundene Position (1. Zeichen=1)
-1	Panther Syntax Fehler, Details siehe d.3 logviewer

Rückgabewert	Bedeutung
-2	Funktion wurde falsch aufgerufen
-3	unbekannter Fehler; d.velop-Support kontaktieren
-107	Text nicht vorgegeben

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.

Anmerkung

```
// -----
// Beispiel:
// Suchen innerhalb von Zeichenketten
// -----
// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_Text = "Anton Berta Caesar Dora Emil Friedrich Gustav"
                               // Ursprüngliche Zeichenkette
vars h_TextToFind = "Caesar" // Text, der gesucht werden
soll
vars h_SearchDirection = 1    // Suchrichtung
vars h_CharWhereToStart      // Zeichen, ab dem gesucht
wird
// Funktionsaufruf
h_ReturnValue = \
api_function ("string_get_position", h_Text, \
                               h_TextToFind, \
                               h_SearchDirection, \
                               h_CharWhereToStart)

// Ergebnisausgabe
if( h_ReturnValue > 0 )
{
    call api_log_info("Die Zeichenkette wurde an Position \
:h_ReturnValue) gefunden. ")
}
else if (h_ReturnValue == 0)
{
    call api_log_info("Die Zeichenkette wurde NICHT gefunden. ")
}
// Fehlerausgabe
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
zurückgeliefert! ")
}
}
```

string_mask_special_char

Diese Funktion maskiert die JAM-Sonderzeichen eines Strings.

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

Parameter	Beschreibung
text	String, der maskiert werden soll

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

api_str_sub

Anmerkung

verfügbar ab d.3 Version 7.0.1

Anmerkung

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 portablem 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 <code>startIdx</code> 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

api_str_sub_utf8

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

api_str_sub_ansi

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

api_str_len

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
inString	die Zeichenfolge

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

api_str_len_utf8

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

api_str_len_ansi

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

api_str_to_utf8

Konvertiert eine Zeichenfolge in das UTF8-Format.

Parameter	Beschreibung
ansiInString	Zeichenfolge die nach UTF8 konvertiert werden soll.
numCodepage	Numerische 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

api_str_from_utf8

Anmerkung

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	Numerische Id der Ziel-Codepage (1250-1258); wird dieser Parameter nicht angegeben, wird die <code>DEFAULT_CODEPAGE</code> angenommen

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

api_str_is_utf8

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

Parameter	Beschreibung
<code>inString</code>	Zeichenfolge, die überprüft werden soll

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

`api_str_is_ascii`

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
<code>inString</code>	die Zeichenfolge, die überprüft werden soll

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

`api_str_cut_fract_utf8`

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
<code>inString</code>	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

`api_str_hash`

Anmerkung

verfügbar ab d.3 Version 7.1.1

Anmerkung

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="background-color: #ffe6e6; padding: 5px; margin-top: 10px;"> <p>Warnung</p> <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)	<p>Mit diesem Parameter kann angegeben werden, dass d.3 automatisch ein Salting des Hashes vornimmt.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tbody> <tr> <td style="text-align: center;">0</td> <td>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.</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Ein in d.3 fest eingebauter Salt wird in den Hash-Wert eingerechnet. Alle d.3 Installationen erstellen bei diesem Modus den gleichen Hash-Wert für die gleiche übergebene Zeichenfolge (<code>value</code>).</td> </tr> <tr> <td style="text-align: center;">2</td> <td>Ein Salt, der für die jeweilige d.3 Installation spezifisch ist, wird in den Hash-Wert eingerechnet. D.h. zwei d.3 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 d.3 gateway verwendet wird.</td> </tr> </tbody> </table> <div style="background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> <p>Anmerkung</p> <p>Um eine eigene Salting-Methode zu realisieren (beispielsweise zum Einrechnen des Benutzernamens in einen Passwort-Hash), kann der eigene Salt direkt an den <code>value</code>-Parameter angehängen werden. Diese Technik lässt sich selbstverständlich auch mit den d.3 Salt-Modi kombinieren.</p> </div>	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.	1	Ein in d.3 fest eingebauter Salt wird in den Hash-Wert eingerechnet. Alle d.3 Installationen erstellen bei diesem Modus den gleichen Hash-Wert für die gleiche übergebene Zeichenfolge (<code>value</code>).	2	Ein Salt, der für die jeweilige d.3 Installation spezifisch ist, wird in den Hash-Wert eingerechnet. D.h. zwei d.3 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 d.3 gateway verwendet wird.
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.						
1	Ein in d.3 fest eingebauter Salt wird in den Hash-Wert eingerechnet. Alle d.3 Installationen erstellen bei diesem Modus den gleichen Hash-Wert für die gleiche übergebene Zeichenfolge (<code>value</code>).						
2	Ein Salt, der für die jeweilige d.3 Installation spezifisch ist, wird in den Hash-Wert eingerechnet. D.h. zwei d.3 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 d.3 gateway verwendet wird.						

Anmerkung

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 d.3 logviewer
	Beispielsweise der Übergabe einer ungültigen Algorithmus-Kennung

1.4.19. Diverses

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

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

restore_from_history

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

Parameter	Beschreibung
aktion_id	Der Wert des Parameters wird seit d.3 server Version 8.0 ignoriert.
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
version_id	Versions-ID der wiederherzustellenden Version. Nur relevant, wenn nicht das komplette Dokument gelöscht ist. Wird eine -1 vorgegeben, so wird die zuletzt aus dem Status Freigabe gelöschte Version wiederhergestellt, falls vorhanden.

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 <code>hook_insert_exit_d3_20</code>

send_email

Die Funktion ermöglicht das Versenden einer E-Mail.

Warnung

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

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

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
-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 <code>api_error_var[.] = "doc_permission"</code> <code>api_error_value[.] = "<Berechtigung>"</code>
-540	SMTP_SUPPORT ist nicht aktiviert <code>api_error_var[.] = "smtp_support"</code> <code>api_error_value[.] = "SMTP_SUPPRT"</code>
-541	Die SMTP DLL ist nicht verfügbar <code>api_error_var[.] = "smtp_dll_available"</code> <code>api_error_value[.] = "SMTP_DLL_VERFUEGBAR"</code>
-542	Fehler beim Senden der E-Mail

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 Repository. Tritt ein Fehler auf, wird die Fehlernummer ausgegeben.

Anmerkung

```

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

// Variablendeklaration
vars h_ReturnValue // Fehlercode
vars h_RecipientUser = "dvelop" // Empfänger
vars h_SubjectText = "Dies kommt in den Betreff" // Betreff der Email
vars h_DokId = "D000000153" // zu versendendes
// zu verlinkendes Dokument
vars h_AttachDokId = 0 // Dokument soll verlinkt
werden
vars h_Sender = "user1" // Absender
vars h_BodyFile = "c:\\text.html" // Datei mit Text für die Email
vars h_MailFormat = "html" // Formatierten Text
verschicken
varsh_CCRecipientUser = info@d-velop.de // Empfänger einer eMail-Kopie
(CC)
varsh_BCCRecipientUser = "d_velop" // Empfänger versteckter
//eMail- Kopie (BCC)
varsh_AttachFile = "c:\\attach.doc" // anzuhängende Datei
varsh_AttachDep = "T1,P1" // anzuhängende abhängige Dateien
vars h_DocStatus = "Be" oder "Pr" oder "Fr" oder "Ar"
vars h_ArchiveIndex

//Empfänger setzen
api_email_cc_recipients[1] = h_CCRecipientUser
api_email_bcc_recipients[1] = h_BCCRecipientUser

// Funktionsaufruf
h_ReturnValue = api_function ("send_email", h_RecipientUser, \
                                h_SubjectText, \
                                "", \
                                h_DokId, \
                                h_Sender, \
                                h_BodyFile, \
                                h_MailFormat, \
                                h_AttachDokId, \
                                h_DocStatus, \
                                h_ArchiveIndex, \
                                h_AttachDep, \
                                h_AttachFile, 0, 1,
1)

// Ergebnisausgabe
if (h_ReturnValue == 0)

```

```

{
    call api_log_info("Die Email wurde erfolgreich verschickt.")
}
// Fehlerausgabe
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
        zurückgeliefert!")
}

```

lock_token_acquire

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 (api_lock_token enthält das vergebene token, api_lock_ttl 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 object_id übergeben
-131	kein object_name übergeben
-255	ungültiges Token übergeben
-600	Fehler beim Löschen eines alten Tokens. Näheres im Logviewer
-601	Objekt bereits gesperrt (api_lock_user und api_lock_info 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

lock_token_release

Gibt eine Sperre für ein Objekt in der DB frei.

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 identifizieren. 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	Token, das zuvor mit lock_token_acquire 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
-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)

start_ext_program

Führt einen Kommandozeilenaufruf durch (Start eines externen Programms)

z. B. notepad.exe C:\\test.txt c:\\test1.txt

Anmerkung

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 Dokument-ID vorgegeben
226	ungültiger Wert für <code>publish</code> , 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

Führt einen Kommandozeilenaufruf durch (Start eines externen Programms)

Anmerkung

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

// Variablendeklaration
vars h_ReturnValue                               // Fehlercode
vars h_Program = "notepad.exe"                   // zu startende
Anwendung
vars h_Params = "c:\\test.txt"                   // Startparameter

// Funktionsaufruf
h_ReturnValue = api_function ("start_ext_program", h_Program,
                              h_Params)

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

get_current_datetime

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
Tag	%d	1/2-stellig	[1..12]
	%0d	2-stellig	[01..31]
	.*d	ganzer Name	Wednesday
Wochentag	%3d	Kurzname	Wed
	%.d	numerisch	[1..7]
	.*d	numerisch	[1..366]
Stunde	%h	1/2-stellig	[1..12] bzw. [0..23]

Datumsformat	String	Beschreibung	Beispiel
	%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

Anmerkung

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

Aktuellen Zeitstempel ausgeben.

Anmerkung

```
// -----
// Beispiel: Aktuellen Zeitstempel ausgeben
// -----
// Variablendeklaration
vars h_ReturnValue // Fehlercode
vars h_Format =
    "Heute ist der %0d. %3m %4y, es ist %0h\:%0M\:%0s.%0S \Uhr"
    // Format der Ausgabe

// Funktionsaufruf
h_ReturnValue = api_function ("get_current_datetime", h_Format)
// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    call api_log_info(api_single_info)
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
    zurückgeliefert!")
}
```

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

Anmerkung

```
// -----
// Beispiel: Aktuellen Zeitstempel ausgeben
// -----
// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_Format =
    "Heute ist der %d.%m.%4y %h%\:0M%\:0s, 0 Uhr"
                                // Format der Ausgabe

// Funktionsaufruf
h_ReturnValue = api_function ("get_current_datetime", h_Format)
// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    call api_log_info(api_single_info)
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
        zurückgeliefert!")
}
}
```

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

Anmerkung

```
// -----
// Beispiel:
// Aktuellen Zeitstempel ausgeben
// -----
// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_Format = "Heute ist der \"%d.%m.%4y %h%\:0M%\:0s %p\", 1"
                                // Format der Ausgabe

// Funktionsaufruf
h_ReturnValue = api_function ("get_current_datetime", h_Format)

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

sleep

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

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

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

https://docs.prolifics.com/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 <code>api_single_info</code>)
-1	Panther Syntax Fehler, Details siehe d.3 logviewer
-2	Funktion wurde falsch aufgerufen
-127	kein regulärer Ausdruck übergeben
-128	kein Testwert übergeben

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

Anmerkung

```
// -----  
  
// Beispiel 1:  
  
// regulären Ausdruck testen  
  
// „api_single_info“ wird auf 1 gesetzt (für gültig)  
  
// ----- vars h_ReturnValue  
  
vars h_Format = "[0-9]{3,6}"  
  
vars h_Wert = "1234"  
  
h_ReturnValue = api_function ("regular_expression_test", h_For-  
mat, h_Wert)  
  
call api_log_info("regular_expression_test")  
  
if (h_ReturnValue == 0)  
{  
  
call api_log_info(api_single_info)  
  
}  
  
else  
  
{  
  
call api_log_error("Fehlernummer :h_ReturnValue wurde zurückge-  
liefert!")  
  
}
```

Beispiel 2; regulärer Ausdruck ungültig

Anmerkung

```
// -----
// Beispiel 2:
// regulären Ausdruck testen
// „api_single_info“ wird auf 0 gesetzt
// (für nicht gültig, weil ein Buchstabe enthalten ist).
// -----
// Variablendeklaration
vars h_ReturnValue                               // Fehlercode
vars h_Format = "[0-9]\{3,6\}" , "1a345" // Format der Ausgabe

// Funktionsaufruf
h_ReturnValue = api_function ("regular_expression_test",
h_Format)

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

Beispiel 3; regulärer Ausdruck ungültig

Anmerkung

```
// -----
// Beispiel 3:
// regulären Ausdruck testen
// „api_single_info“ wird auf 0 gesetzt
// (für nicht gültig, weil nur 2 Zeichen)
// -----
// Variablendeklaration
vars h_ReturnValue                               // Fehlercode
vars h_Format = "[0-9]\{3,6\}" , "12" // Format der Ausgabe

// Funktionsaufruf
h_ReturnValue = api_function ("regular_expression_test",
h_Format)

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

object_property_set

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

Parameter	Beschreibung
<code>property_name (P)</code>	interner Name der Objekteigenschaft
<code>object_id (P)</code>	Objektname (z.B. Benutzername, Gruppenname abhängig von der Objekt-Klassen-ID)
<code>object_class_id (P)</code>	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
<code>property_value</code>	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

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
<code>api_property_name</code>	Eigenschaftsname
<code>api_property_caption</code>	Kurzbeschreibung der Eigenschaft
<code>api_property_description</code>	ausführliche Beschreibung der Eigenschaft
<code>api_property_value</code>	Wert der Eigenschaft
<code>api_object_id</code>	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

set_signature_data

Anmerkung

wird ab Version 7.1 nicht mehr unterstützt

Bei Aufruf kommt Fehler -671.

Schreibt zu einem Dokument die Signaturinformationen in die Datenbank.

Parameter	Beschreibung
doc_id	Dokument-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_verification_level (0)	erreichte Prüftiefe (1-9)
api_sig_requested_verification_level (0)	gewünschte Prüftiefe (1-9)
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_ceralgorithm	Zertifikatsalgorithmus
api_sig_certrestriction (0)	textliche Angabe der Beschränkungen, die für das zugehörige Zertifikat gelten

Anmerkung

Einschränkungen:

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

get_objectid_from_synchid

Ermittelt aus einer Synchronisations-ID, die über das Transportsystem erstellt wurden, die interne ID und den Typen eines Stammdaten-Objektes.

globale Variable	Beschreibung
api_md_object_id	Interne ID des Stammdaten-Objektes
api_md_object_type	Typ des Stammdaten-Objektes

Parameter	Beschreibung
synch_id	Synchronisations-ID

Rückgabewert	Bedeutung
0	Alles OK.
-5009	Leere oder nicht existente Synchronisations-ID

1.4.20. 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.

Anmerkung

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).

Warnung

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 nachfolgend aufgeführt.

Anmerkung

Vier Beispiele befinden sich am Ende dieses Kapitels.

- `d3fc_user_set`
- `3fc_password_set`
- `d3fc_remote_server_set`
- `d3fc_port_set`
- `d3fc_server_set`
- `d3fc_timeout_set`
- `d3fc_function_name_set`
- `d3fc_importing_set`
- `d3fc_exporting_set`
- `d3fc_exporting_get`
- `d3fc_table_set_import_file`
- `d3fc_table_set_export_file`
- `d3fc_table_set_headline`
- `d3fc_table_add`
- `d3fc_field_set`
- `d3fc_field_get`
- `d3fc_first`
- `d3fc_next`
- `d3fc_execute`

d3fc_user_set

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
-102	kein Benutzername angegeben

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

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

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

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

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

d3fc_function_name_set

Name des API-Calls.

Anmerkung

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

d3fc_importing_set

Importparameter setzen.

Anmerkung

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

d3fc_exporting_set

ExportParameter setzen.

Anmerkung

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

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

d3fc_table_set_import_file

Import-Datei setzen.

Name (inklusive Pfad) der Datei, die der Call als Eingabe bekommt.

Anmerkung

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

d3fc_table_set_export_file

Export-Datei setzen.

Name (incl. Pfad) der Datei, in die die Ausgabe des Calls geschrieben werden soll.

Anmerkung

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

d3fc_table_set_headline

Tabellenkopf setzen.

Anmerkung

Dieser Wert kann nur alternativ zur Import/Export-Datei gesetzt werden, nicht gleichzeitig.

Parameter	Beschreibung
headline	Form: <head1><Leerzeichen><head2><Leerzeichen>...<headN><Leerzeichen><Punkt>

Anmerkung

`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 Dokument-ID 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

d3fc_table_add

Tabellenzeile zur Import-Tabelle hinzufügen.

Anmerkung

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

d3fc_field_set

Feld in der aktuellen Tabellenzeile der Importtabelle setzen.

Anmerkung

Die Headline muss bereits gesetzt sein.

`d3fc_table_add` muss bereits aufgerufen worden sein.

Parameter	Beschreibung
<code>field_name</code>	Name der Tabellenspalte in Headline

Parameter	Beschreibung
<code>field_value</code>	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 <code>field_name</code> angegeben
-613	Feld in Tabelle nicht vorhanden

d3fc_field_get

Feld in der aktuellen Tabellenzeile der Export-Tabelle lesen.

Anmerkung

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
<code>field_name</code>	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 <code>field_name</code> angegeben
-612	Feld in Tabelle nicht vorhanden

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)

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)

d3fc_execute

Einen API-Call ausführen.

Anmerkung

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 Zielfeile 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 Zielfeilenamen (<code>api_d3fc_export_file</code>)
-609	Tabellenende erreicht (Konstante EOT = -609)

Beispiele

- `ValidatePasswordForUser`
- `SearchDocumentInIDList`
- `ReceiveNote`
- `SendNote`

Beispiel 1 - ValidatePasswordForUser**Anmerkung**

```

// -----
// Beispiel 1:
// ValidatePasswordForUser
// -----
// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_User = "armin"       // Benutzer für d3fc-Aufruf
vars h_Password = "frikadelle" // Passwort für d3fc-Aufruf
// Verbindungsoptionen
call api_function ("d3fc_user_set", h_User)
    // Benutzer; default: d3_server
call api_function ("d3fc_password_set", h_Password)
    // Passwort; default: sever_api_client
call api_function ("d3fc_remote_server_set", "host")
    // Hostname; default: d3fc_remote_server_name
call api_function "d3fc_port_set", "3400")
    // Port; default: d3fc_remote_server_tcpip_port
call api_function ("d3fc_server_set", "Y")
    //Kürzel; default: d3fc_remote_server_kuerzel
// Funktionsname
call api_function ("d3fc_function_name_set",
    "ValidatePasswordForUser")
//Funktionsname
// ImportParameter
call api_function ("d3fc_importing_set", "user", h_User)
//zu prüfender Benutzername
call api_function ("d3fc_importing_set", "password", h_Password)
// zu prüfendes Passwort
// Funktionsaufruf
h_ReturnValue = api_function ("d3fc_execute")
// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    if (api_single_info == 0)
        call api_log_info("Kombination aus Benutzer und
            Passwort gültig")
    else
        call api_log_error("Fehlernummer :api_single_info \
            wurde von ValidatePasswordForUser \
            zurückgeliefert!")
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
        zurückgeliefert!")
}

```

Beispiel 2 - SearchDocumentInIDList**Anmerkung**

```

// -----
// Beispiel 2:
// SearchDocumentInIDList
// -----
// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_counter = 0         // Schleifenzähler
vars h_DokID = "Y000005174" // zu suchendes Dokument
//Verbindungsoptionen
call api_function ("d3fc_user_set", "user")
// Benutzer; default: d3_server
call api_function ("d3fc_password_set", "password")
// Passwort; default: sever_api_client
call api_function ("d3fc_remote_server_set", "host")
// Hostname;
// default: d3fc_remote_server_name
call api_function ("d3fc_port_set", "3400")
// Port;
// default: d3fc_remote_server_tcpip_port
call api_function ("d3fc_server_set", "Y")
// Kürzel;
// default: d3fc_remote_server_kuerzel

//Funktionsname
call api_function ("d3fc_function_name_set",
"SearchDocumentInIDList")
// Funktionsname

//Import-Tabelle
call api_function ("d3fc_table_set_headline", "doc_id      .",
"I")
//Tabellenkopfzeile
call api_function ("d3fc_table_add")
// Tabellenzeile hinzufügen
call api_function ("d3fc_field_set", "doc_id", h_DokID)
//Wert für eine Spalte in dieser Zeile

// ExportParameter
call api_function ("d3fc_exporting_set ", "number")
//Anzahl Treffer

// Export-Tabelle
call api_function ("d3fc_table_set_headline",
"doc_id doc_status .", "O")
// Tabellenkopfzeile
// Funktionsaufruf
h_ReturnValue = api_function ("d3fc_execute")
// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
vars h_numberReturned
vars h_ret
call api_log_info("SearchDocumentInIDList lieferte ReturnCode
\
:api_single_info")

```

```
// ExportParameter abfragen
call api_function ("d3fc_exporting_get", "number")
h_numberReturned = api_single_info
call api_log_info(":api_single_info Treffer")
h_Ret = api_function ("d3fc_first")
    // Erste Zeile holen
while (h_Ret != EOT)
    //EOT ist Konstante für Rückgabe -609
{
    for h_counter=1 while h_counter <= h_numberReturned step 1
    {
        call api_log_info("(h_counter). Treffer:")
        call api_function ("d3fc_field_get", "doc_id")
            // Wert abfragen
        call api_log_info("doc_id=:api_single_info")
        call api_function ("d3fc_field_get", "doc_status")
        call api_log_info("doc_status=:api_single_info")
    }
    h_Ret = api_function ("d3fc_next ")
        // nächste Zeile
    }
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
        zurückgeliefert!")
}
}
```

Beispiel 3 - ReceiveNote**Anmerkung**

```

// -----
// Beispiel 3:
// ReceiveNote
// -----
// Variablendeklaration
vars h_ReturnValue          // Fehlercode
vars h_DokID = "Y000005177" // DokuID
vars h_FileName = "c:\\\\notiz.txt" // Speicherort für Notiz
//Verbindungsoptionen
call api_function ("d3fc_user_set", "user")
    // Benutzer; default: d3_server
call api_function ("d3fc_password_set", "password")
    // Passwort; default: sever_api_client
call api_function ("d3fc_remote_server_set", "host")
    // Hostname;
    // default: d3fc_remote_server_name
call api_function ("d3fc_port_set", "3400")
    // Port;
    // default: d3fc_remote_server_tcpip_port
call api_function ("d3fc_server_set", "Y")
    // Kürzel;
    // default: d3fc_remote_server_kuerzel
// Funktionsname
call api_function ("d3fc_function_name_set", "ReceiveNote")
    // Funktionsname
// ImportParameter
call api_function ("d3fc_importing_set", "doc_id", h_DokID)
    // Dokument
// Exportfile
call api_function ("d3fc_table_set_export_file", h_FileName)
    // Dateiname, unter dem die
    // Notiz abgelegt wird
// Funktionsaufruf
h_ReturnValue = api_function ("d3fc_execute")
// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    call api_log_info("Notizdatei wurde unter
        :h_FileName gespeichert.")
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
        zurückgeliefert!")
}

```

Beispiel 4 - SendNote**Anmerkung**

```

// Beispiel 4:
// SendNote
// -----
// Variablendeklaration
vars h_ReturnValue                               // Fehlercode
vars h_DokID = "Y000005177"                     // DokuID
vars h_FileName = "c:\\notiz.txt"              // Speicherort für
Notiz
//Verbindungsoptionen
call api_function ("d3fc_user_set", "user")
// Benutzer; default: d3_server
call api_function ("d3fc_password_set", "password")
// Passwort; default: sever_api_client
call api_function ("d3fc_remote_server_set", "host")
// Hostname;
// default: d3fc_remote_server_name
call api_function ("d3fc_port_set", "3400") // Port; default:
// d3fc_remote_server_tcpip_port
call api_function ("d3fc_server_set", "Y") // Kürzel; default:
// d3fc_remote_server_kuerzel
//Funktionsname
call api_function ("d3fc_function_name_set", "SendNote")
//Funktionsname
//ImportParameter
call api_function ("d3fc_importing_set", "doc_id", h_DokID)
//Dokument
//Importfile
call api_function ("d3fc_table_set_import_file", h_FileName)
//Dateiname der Notiz
// Funktionsaufruf
h_ReturnValue = api_function ("d3fc_execute")
// Ergebnisausgabe und Fehlerbehandlung
if (h_ReturnValue == 0)
{
    call api_log_info("Notizdatei :h_FileName wurde gespeichert.")
}
else
{
    call api_log_error("Fehlernummer :h_ReturnValue wurde
zurückgeliefert!")
}

```

1.4.21. JAVA

- java_add_param
- java_clear_param
- java_call_static

java_add_param

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.

Anmerkung

verfügbar ab d.3 Version 7.2

Parameter	Beschreibung
param_value	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)

java_clear_param

Diese Funktion löscht alle bisher hinzugefügten Parameter (siehe [java_add_param](#)).

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)

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.

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.

Anmerkung

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 JavaHooks im Package com.dvelop.test
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)

1.4.22. Erstellung von Async-Jobs

Warnung

ab d.3 Version 8.0

d3async_job_open

Warnung

ab d.3 Version 8.0

Öffnet einen Async Job.

Mit `d3async_job_set_attribute` bzw. `d3async_job_set_lin002_attribute` können zusätzliche Async Job Parameter gesetzt werden.

Der Async Job wird dann abschließend über `d3async_job_close` in der Datenbank erstellt.

Parameter	Beschreibung
<code>doc_id</code>	Dokument-ID
<code>job_type</code>	Async Job Typ (z.B. LIN002)
<code>user_name</code>	Benutzername, in dessen Name der Job ausgeführt werden soll. Wenn leer, wird der Benutzername aus dem Kontext des Hooks übernommen.

Rückgabewert	Bedeutung
0	alles ok
-680	kein Job-Typ vorgegeben

d3async_job_set_attribute

Setzt Async Job Attribute.

Parameter	Beschreibung
<code>attr_name</code>	Async Job Attributname
<code>attr_value</code>	Wert für Async Job Attribut

Rückgabewert	Bedeutung
0	alles ok
-681	kein Attribut-Name vorgegeben

d3async_job_set_lin002_attr

Setzt die Job-Attribute speziell für LIN002-Jobs und kann somit auch nur bei LIN002-Jobs aufgerufen werden.

Parameter	Beschreibung
attr_type	muss einen der folgende Werte haben ernie_job_attr_name: Erkennungsattribute für das übergeordnete Dokument inherit_field_name_1: Vererbungsattribute vom untergeordneten Dokument auf das übergeordnete Dokument, wobei die Attribute des übergeordneten Dokumentes nicht überschrieben werden inherit_field_name_2: Vererbungsattribute vom untergeordneten Dokument auf das übergeordnete Dokument, wobei die Attribute des übergeordneten Dokumentes überschrieben werden inherit_field_name_3: Vererbungsattribute vom übergeordneten Dokument auf das untergeordnete Dokument, wobei die Attribute des untergeordneten Dokumentes nicht überschrieben werden inherit_field_name_4: Vererbungsattribute vom übergeordneten Dokument auf das untergeordnete Dokument, wobei die Attribute des untergeordneten Dokumentes überschrieben werden
attr_name	Attributname (z.B. "dok_dat_feld_1")
attr_value	Wert für Dokument-Attribut

Rückgabewert	Bedeutung
0	alles ok
-681	kein Attribut-Name vorgegeben

d3async_job_close

Schließt einen mit `d3async_job_open` eröffneten Job ab, schreibt diesen in die Datenbank und setzt den Job-Status so, dass der d.3 async Prozess diesen starten kann.

`api_single_info` gibt bei Erfolg die Job-Nummer zurück.

Rückgabewert	Bedeutung
0	Async Job wurde erstellt und zur Verarbeitung freigegeben
-682	Async Job konnte nicht erstellt werden

1.5. JPL-Styleguide

1.5.1. 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:

Anmerkung

```
C:\d3\d3server\hooks\d3P\testfirma.jpl
```

1.5.2. 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.

Anmerkung

```
// Kommentarzeile
```

Mehrzeilige Kommentarzeilen sollten mit einer Start- und Endzeile geklammert sein.

Anmerkung

```
//-----  
// Komentarzeile  
// Komentarzeile  
//-----
```

1.5.3. Kopf

Der Kopf einer JPL-Datei, dies gilt für externe JPL-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 Änderungsnummer 007 referenziert werden. Im Code selber taucht diese Nummer ebenfalls auf. So können die betroffenen Zeilen sehr schnell gefunden werden.

Anmerkung

```

//-----
// 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 Sachbearbieters.
//
//      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

1.5.4. Variablen

- Generell
- erweiterte Eigenschaften-Felder und Dokumentarten

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.

Anmerkung

`h_Count` oder `h_SachBearbeiter` oder `h_VersNr`.

Erweiterte Eigenschaften-Felder und Dokumentarten

In den einzelnen Hook-Funktionen 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:

- erweiterte Eigenschaften-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'

Anmerkung

```

/-----
// DB-Position der erweiterte Eigenschaften-Felder
//-----
vars h_DDF_MANDANT          = 2
vars h_DDF_BUCHUNGSKREIS   = 3
vars h_DDF_OBJEKT          = 7
vars h_DDF_DOKUTYP         = 25
//-----
// Dokument/Aktenarten
//-----
vars h_DA_AAUF              = "AAUF"
```

Der Zugriff könnte dann wie folgt aussehen:

Anmerkung

```

// DB-Zugriff
DBMS SQL SELECT a.dok_dat_feld_:h_DDF_RENAME \
          FROM firmen_spezifisch a          \
          WHERE a.kue_dokuart = :+h_DOK_AREMP \
          AND a.dok_dat_feld_:h_DDF_RECHNR=
          :+dok_dat_feld[h_DDF_RECHNR]
```

1.5.5. Debugging

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.

Anmerkung

```

proc FunktionsName
{
  call api_log_debug ("FunktionsName start")
  // Variablendeklaration
  ...
  // Code
  ...
  call api_log_debug ("FunktionsName Irgendeine Meldung")
  // Code
  ...
  call api_log_debug ("FunktionsName end")
}

```

1.5.6. Schleifen und Verzweigungen**Anmerkung**

Jede einzelne Ebene wird eingerückt.

- For-Schleife
- While-Schleife
- If-Abfrage

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.

Anmerkung

```

vars h_Count
for h_Count = 0 while(string( h_Count, 1) != "") step 1
{
  Anweisungs-Block
}

```

While-Schleife**Anmerkung**

```

vars h_Count = 1
while (string(h_Count,1) != "")
{
  h_Count = h_Count+1
}

```

If-Abfrage

Wenn mehrere Bedingungen geprüft werden, steht jede Bedingung in einer einzelnen Zeile. Desweiteren sollte mit Klammern gearbeitet werden; dies unterstützt die Lesbarkeit. Die Bedingungen sollten auch optisch untereinander stehen.

Anmerkung

```

if ( (x > y) \
    && (z < x) \
    || (x > w))
{
    Anweisungs-Block
}
else
{
    Anweisungs-Block
}

```

1.6. Nützliche Funktionen**1.6.1. WriteNotizEntry**

Diese Funktion liest die Werte aus dem 61er-Feld aus und schreibt sie in die Notizen.

Anmerkung

```

//Werte aus 61er-Feld lesen und in die Notizdatei schreiben

proc WriteNotizEntry( h_DokID )
{
    // Pfad bereitstellen UNIX
    // d.3-Archiv-ID in den Pfad einbauen!

    vars h_Pfad      = "/opt/d3/d3server/hooks/d3:
(d3fc_server_kuerzel)/
                        work/"
    vars h_FileName = h_Pfad ## h_DokID ## "_Notiz.txt"
    vars h_Count = 1
    vars h_ReturnValue
    vars h_Flag = 0
    // Lesen der 60-er Felder
    for h_Count = 1 while (dok_dat_feld_61[h_Count] != "") step 1
    {
        // Schreiben in eine tmp-Datei mit Doku-ID als Dateinamen
        h_ReturnValue = \
        api_function ("file_add_line", h_FileName, \
                    \":(dok_dat_feld_61[:h_Count])\\"", "APPEND" )
        h_Flag = 1
    }
    if( h_Flag == 1 )
    {
        // Schreiben der Datei in den Notiz-Eintrag
        h_ReturnValue = api_function ("note_add_file", \
                                    h_FileName, h_DokID, "TEST1"
        )
    }

    // Löschen der Datei
    if (h_ReturnValue == 0)
        call api_function ("file_delete", h_FileName)
    }
}
} // end of WriteNotizEntry

```

1.6.2. AddPrefixAndZeroes

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.

Anmerkung

```
// Einer Zeichenkette einen Präfix und Nullen anfügen

string proc AddPrefixAndZeros (wf_String, wf_Prefix,
wf_MaxLength)
{
  if (wf_String(1, @length(wf_Prefix)) != wf_Prefix)
  {
    while (@length(wf_String) < wf_MaxLength)
      wf_String = "0" ## wf_String
      wf_String = wf_Prefix ## wf_String
    }
  return wf_String
} // end of AddPrefixAndZeros
```

1.6.3. CutPrefixAndZeros

Die Funktion `CutPrefixAndZeros` entfernt von einer Zeichenkette das Präfix und die folgenden Nullen.

Anmerkung

```
string proc CutPrefixAndZeros( wf_String, wf_Prefix,
wf_MinLength )
{
  if (wf_String(1, @length(wf_Prefix)) == wf_Prefix)
  {
    wf_String = wf_String(@length(wf_Prefix)+1) // Cut Prefix
    while (wf_String(1,1) == "0" && @length(wf_String)
    >; wf_MinLength)
      wf_String = wf_String(2)
    }
  return wf_String
} // end of CutPrefixAndZeros
```

1.6.4. OpenSessionConnection

Die folgende Funktion öffnet eine ODBC-Verknüpfung.

Anmerkung

```

proc OpenSessionConnection( h_Umgebung )
{
  vars h_tmpUmgebung, h_Count, h_Session, h_DBReturnCode

  call api_function ("string_to_lower", h_Umgebung)
  h_tmpUmgebung = api_converted_string

  // Ermittlung des ODBC-Namens abhängig von der Umgebung
  for h_Count = 1 while(h_Count <= h_MaxConnectCount) step 1
  {
    if (h_tmpUmgebung == h_ODBC_Name[h_Count] \
        (@length(h_ODBC_Name[h_Count])))
      break
  }
  if (h_Count <= h_MaxConnectCount)
  {
    h_Session = h_SessionName[h_Count]

    call api_log_info ("CONNECTION-STATUS für :h_Session " \
                      ## dm_is_connection( h_Session ))
    if (dm_is_connection( h_Session ) == 0)
    {
      DBMS DECLARE :h_Session CONNECTION FOR USER
      :+h_ODBC_USER \
          [h_Count] \
          PASSWORD :+h_ODBC_PW[h_Count] \
          DATASOURCE :h_ODBC_Name[h_Count]
      h_DBReturnCode = @dmretcode
      DBMS WITH CONNECTION :h_Session COMMIT
    }

    if (dm_is_connection( h_Session ) == 1)
    {
      // Test-Abfrage gegen Datenbank!
      vars h_TmpValue
      DBMS WITH CONNECTION :h_Session ALIAS h_TmpValue
      DBMS WITH CONNECTION :h_Session SQL SELECT NAME \
          FROM :(h_TableOwner[h_Count]). \
          VWKI_KDNR_ADR \
          WHERE KDNR = 4711
      h_DBReturnCode = @dmretcode
      DBMS WITH CONNECTION :h_Session ALIAS
      DBMS WITH CONNECTION :h_Session COMMIT
      call api_log_info ("1.Fehlercode: :(@dmengerrcode) \
                        :(@dmengerrmsg) \
                        :(@dmretcode) :(@dmretmsg) !")
      if (h_DBReturnCode != 0)
      {
        DBMS CLOSE CONNECTION :h_Session
        DBMS DECLARE :h_Session CONNECTIN FOR USER \
            :+h_ODBC_USER[h_Count] \
            PASSWORD :+h_ODBC_PW[h_Count] \
            DATASOURCE :h_ODBC_Name[h_Count]
      }
    }
  }
}

```

```

        DBMS WITH CONNECTION :h_Session COMMIT
        call api_log_info (2.Fehlercode: :(@dmengerrcode) \
                        :(@dmengerrmsg) : \
                        :(@dmretcode) :(@dmretmsg) !")
        //return -1005
        // msglib.usr ( DB-ERROR Siehe Log-File
        // für weitere Informationen)
    }
}
return h_Count
}
return -1
} // end of OpenSessionConnection

```

1.6.5. GruppeAnwesend

Die Funktion `GruppeAnwesend(h_Gruppe)` ermittelt, ob mindestens ein Mitglied der Gruppe anwesend ist.

Anmerkung

```

// 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

```

1.6.6. BenutzerAnwesend

Mit dieser Funktion wird geprüft, ob der übergebene Benutzer zur Zeit anwesend ist.

Anmerkung

```
// 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
```

Anmerkung

```
proc BenutzerAnwesend (h_SachBe)
{
  DBMS ALIAS h_TmpFlag
  DBMS SQL SELECT checkout_user_flag \
    FROM benutzer \
    WHERE benutzername = :+h_SachBe
  h_MaxCount = @dmrowcount
  DBMS ALIAS
  if (h_MaxCount != 0 &&& h_TmpFlag != 1)
    return 1
  else
    return 0
} // end of BenutzerAnwesend
```

1.6.7. BenutzerIstGruppe

BenutzerIstGruppe ermittelt, ob der übergebene User eine Gruppe ist.

Anmerkung

```
// Prüft, ob h_SachBe eine Benutzergruppe ist.
proc BenutzerIstGruppe (h_SachBe)
{
  vars h_TmpFlag, h_MaxCount = 0
  DBMS ALIAS h_TmpFlag
  DBMS SQL SELECT benutzergruppe \
    FROM benutzergruppen \
    WHERE benutzergruppe = :+h_SachBe
  h_MaxCount = @dmrowcount
  DBMS ALIAS
  return h_MaxCount
} // end of BenutzerIstGruppe
```

1.6.8. ErmittlungBenutzer

Die Funktion ErmittlungBenutzer bestimmt Benutzer anhand einer PESNr die im OptionalenFeld1 der Benutzerdaten hinterlegt wurde.

Anmerkung

```
// Bestimmung des Benutzers anhand einer PESNr
string proc ErmittlungBenutzer (h_PESNr)
{
  if (h_PESNr != "")
  {
    vars h_MaxCount = 1
    vars h_Benutzer = ""
    DBMS ALIAS h_Benutzer
    DBMS SQL SELECT benutzername benutzer_feld_1 \
      FROM benutzer \
      WHERE benutzer_feld_1=:h_PESNr
    h_MaxCount = @dmrowcount
    DBMS ALIAS
    if (h_MaxCount > 0)
      return h_Benutzer
  }
  return ""
}
} // end of ErmittlungBenutzer
```

1.6.9. fillWmFromDbField

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).

Anmerkung

```

//*****
// * Funktion zum Ermitteln von Werten (--&gt; Wertemenge)
// * aus einer Datenbank anhand vorgegebener Kriterien
// * (Eingabe: Tabelle, Spalte, Wert (Schlüssel) --&gt;
// * Wertemenge wird mit
// * den Werten anderer Spalte gefüllt)
//*****

proc fillWmFromDbField (dbTable, dbColSrc, dbColTarget,
baseValue, \
                        h_repos_id )
{
  call api_log_info ("+++++++ HOOK: fillWmFromDbField START +++++
+++")
  vars h_dbValue, h_rowCount

  DBMS ALIAS h_dbValue
  DBMS SQL SELECT Count (*) FROM :dbTable WHERE :dbColSrc \
              like ':(baseValue)%'
  h_dbValue = @dmrowcount
  DBMS ALIAS

  if( h_dbValue > h_MAX_AMOUNT_VALUES )
  {
    d3server_value_char_allowed = 'Zu viele Werte'
    d3server_repos_id_allowed = ':h_repos_id'
    return
  }
  else if (h_dbValue == 0)
  {
    d3server_value_char_allowed = ''
    d3server_repos_id_allowed = ':h_repos_id'
    return
  }

  // wenn das Limit nicht überschritten worden ist,
  // dann zeige alle Möglichkeiten in der Wertemenge an
  DBMS ALIAS d3server_value_char_allowed,
d3server_repos_id_allowed
  DBMS SQL SELECT DISTINCT :dbColTarget, `:h_repos_id`
              FROM :dbTable WHERE :dbColSrc like ':(baseValue)%'
  DBMS ALIAS

  call api_log_info ("+++++++ HOOK: fillWmFromDbField ENDE +++++
++")
} // end of fillWmFromDbField

```

1.6.10. getValueFromDBField

Funktion zum Ermitteln eines Wertes aus einer Datenbank anhand vorgegebener Kriterien (Eingabe: Tabelle, Spalte, Wert (Schlüssel) --> Ausgabe: Wert anderer Spalte).

Anmerkung

```

//*****
// * Funktion zum Ermitteln eines Wertes aus einer Datenbank
// * anhand
// * vorgegebener Kriterien (Eingabe: Tabelle, Spalte, Wert
// * (Schlüssel)
// * --> Ausgabe: Wert anderer Spalte)
//
//*****

string proc
getValueFromDBField (dbTable, dbColSrc, dbColTarget, \
                    baseValue )
{
  call
  api_log_info ("+++++++ HOOK: getValueFromDBField START ++++++")

  vars h_dbValue, h_rowCount, dummy

  //
  // Wenn Datensätze doppelt auftreten (Schlüssel): bei der SQL Count
  // Abfrage wird gruppiert (Schlüssel) und die Anzahl der
  //
  // zurückgegebenen Datensätze (aus 2 mach 1) ausgewertet (h_dbValue)
  DBMS ALIAS
  DBMS SQL SELECT :dbColSrc, Count(*) \
    FROM :dbTable \
    WHERE :dbColSrc = :+baseValue \
    GROUP BY :dbColSrc
  h_dbValue = @dmrowcount
  DBMS ALIAS

  if (h_dbValue > 1)
  {
    return h_ERR_TOO_MANY_VALUES
  }

  // Wenn es einen oder keinen Wert gibt
  DBMS ALIAS h_dbValue
  DBMS SQL SELECT DISTINCT :dbColTarget FROM :dbTable WHERE \
    :dbColSrc = :+baseValue
  DBMS ALIAS

  call api_log_info
  ("+++++++ HOOK: getValueFromDBField ENDE ++++++")
  return h_dbValue
}

```

1.6.11. hasDtsReposId

Funktion, um herauszufinden, ob eine Dokumentart ein Repository besitzt (anhand der Repository ID).

Anmerkung

```

//*****
// * Funktion um herauszufinden, ob eine Dokumentart ein
Repository
// * besitzt (anhand der Repository ID)
//
//*****
*

proc hasDtsReposId (h_dts, h_repos_id)
{
call api_log_info "+++++++ HOOK: hasDtsReposId START +++++++
+")
// Ermittle, ob es in der DB einen Eintrag gibt, der besagt,
// ob ein Repository einer Dokumentart zugewiesen ist
vars h_count = 0, retVal = 0
DBMS ALIAS h_count
DBMS SQL SELECT Count(*) \
    FROM fispe_titel_dokuart \
    WHERE kue_dokuart = :+h_dts AND repository_id = :+h_repos_id
DBMS ALIAS

    // Es gibt eine Zuweisung...
    if (h_count > 0)
    {
        retVal = 1
    }
    // Es gibt KEINE Zuweisung ...
    else
    {
        retVal = 0
    }
call api_log_info ("+++++++ HOOK: hasDtsReposId ENDE +++++++
+")
    return retVal
}

```

1.6.12. AvailableValues

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%').

Anmerkung

```

//
*****
// * 20.06.2006
// * Wertemenge, die alle Werte liefert, die bereits zu diesem
//
* dok-dat-feld (Dokuart abhängig) im d3 System vorhanden sind;
//
* Eingegebene Werte in dem Dok_dat_feld werden mit in den Query
// * eingebaut (mit LIKE 'xxx%')
//
*****

proc AvailableValues (h_ReposID, h_User, h_DoctypeShort)
{
  call api_log_info ("+++++++ HOOK: DynWM
                    AvailableValues START ++++++")

  vars h_ddf
  vars h_AnzRow, h_LikeClause = ""
  vars h_MaxCount = 1500

  // ermittle dok_dat_feld zu der repos_id

  DBMS ALIAS h_ddf
  DBMS SQL SELECT DISTINCT doc_field_nr_pref FROM >
                    field_repository WHERE \
                    repository_id = :h_ReposID
    h_AnzRow = @dmrowcount
  DBMS ALIAS

  // ermittle Anzahl der gefundenen Werte
  // (d.h. Größe der Wertemenge)
  // Eingabe im dokdatfeld mit in den Query eingebaut
  // (für Einschränkungen)

  if (dok_dat_feld[:h_ddf] != "")
    h_LikeClause = "and dok_dat_feld[:h_ddf] LIKE ':"
                  (dok_dat_feld[:h_ddf])%'"

  DBMS ALIAS h_AnzRow
  DBMS SQL SELECT count (DISTINCT dok_dat_feld[:h_ddf]) \
                    FROM firmen_spezifisch \
                    WHERE kue_dokuart = ':h_DoctypeShort' :h_LikeClause
  DBMS ALIAS

  // wenn das Limit überschritten worden ist,
  // dann wird ein Hinweis als
  // Wertemengeninhalte ausgegeben (macht Sinn bei sehr
  // großen Wertemengen,
  // wie z.B. 10000 Einträgen oder mehr)

  if( h_AnzRow > h_MaxCount )
  {

```

```

call api_log_info ("Mehr als :h_MaxCount Werte gefunden
                  :h_AnzRow")
d3server_value_char_allowed = "Mehr als
                              :h_MaxCount Werte gefunden :h_AnzRow"
d3server_repos_id_allowed   = h_ReposID
return
}
else if( h_AnzRow == 0 )
{
  call api_log_info ("Keinen Wert gefunden")
  d3server_value_char_allowed
= 'Es sind keine Werte vorhanden.'
  d3server_repos_id_allowed   = h_ReposID
  return
}

// wenn das Limit nicht überschritten worden ist,
// dann zeige alle Möglichkeiten
// in der Wertemenge an Eingabe im dokdatfeld mit in den
// Query eingebaut (für Einschränkungen)

DBMS
ALIAS d3server_value_char_allowed, d3server_repos_id_allowed
DBMS SQL SELECT DISTINCT dok_dat_feld :h_ddf, ':h_ReposID' \
      FROM firmen_spezifisch \
      WHERE kue_dokuart = ':h_DoctypeShort' :h_LikeClause
DBMS ALIAS
call api_log_info ("+++++++ HOOK: >
                  DynWM AvailableValues ENDE ++++++")
} // end of AvailableValues

```

1.7. Webservice-Funktionen

d.3 webservice Funktionen

Anmerkung

Sofern nicht anders angegeben, geben alle Funktionen bei Erfolg den String OK zurück, ansonsten eine Fehlermeldung.

Initialisierung	
d3ws_startSession()	Initialisierung, gibt eine Session-ID zurück (z.Zt. ohne Verwendung)
d3ws_stopSession()	Finalisierung (optional, entspricht neuer Initialisierung)

XML-Erzeugung (-> XML-Out-Buffer)	
d3ws_xml_start()	Beginn eines neuen XML-Dokuments, Löschen des bisherigen Puffers
d3ws_xml_openNode (name)	öffnet einen neuen Knoten, der mit CloseNode wieder geschlossen werden muss (also "<name>")
d3ws_xml_addNode (name, value)	erzeugt einen kompletten Knoten (also "<name>value</name>")
d3ws_xml_closeNode (name)	schließt einen Knoten, der mit OpenNode geöffnet wurde (also "</name>")
d3ws_xml_stop()	beendet das XML-Dokument, das jetzt gespeichert oder versendet werden kann

Laden, Speichern, Aufruf Web-Service	
d3ws_xml_saveToFile (filename)	speichert den XML-Out-Buffer in die angegebene Datei
d3ws_xml_loadFromFile (filename)	lädt die angegebene Datei in den XML-In-Buffer
d3ws_http_addParameter (name, value)	fügt einen Parameter hinzu, der beim späteren Aufruf von <code>sendRequest</code> als POST-Parameter mitgegeben wird
d3ws_http_setTraceFile (filename)	protokolliert nachfolgenden Aufruf von <code>sendRequest</code> in einer Datei (zur Fehlersuche geeignet)
d3ws_http_sendRequest (url)	<p>Sendet den XML-Out-Buffer an die angegebene URL und lädt das Ergebnis des Requests in den XML-In-Buffer.</p> <p>In der URL können GET-Parameter verwendet werden.</p> <p>Wenn zuvor KEIN weiterer Parameter mitgegeben wurde (<code>addParameter</code>), wird das XML per POST namenlos mitgesendet.</p> <p>Andernfalls wird das XML per POST als Parameter <code>"xml="</code> mitgesendet.</p>

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 <code>countNodes</code> 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 <code>setNode ("i")</code> , allerdings kann hier wirklich ein Integer übergeben werden (<code>setNode</code> 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 <code>setNode</code></p>

1.8. Weitere Informationsquellen und Impressum

Wenn Sie Ihre Kenntnisse rund um die d.velop-Software vertiefen möchten, besuchen Sie die digitale Lernplattform der d.velop academy unter <https://dvelopacademy.keelelearning.de/>.

Mithilfe der E-Learning-Module können Sie sich in Ihrem eigenen Tempo weiterführende Kenntnisse und Fachkompetenz aneignen. Zahlreiche E-Learning-Module stehen Ihnen ohne vorherige Anmeldung frei zugänglich zur Verfügung.

Besuchen Sie unsere Knowledge Base im d.velop service portal. In der Knowledge Base finden Sie die neusten Lösungen, Antworten auf häufig gestellte Fragen und How To-Themen für spezielle Aufgaben. Sie finden die Knowledge Base unter folgender Adresse: <https://kb.d-velop.de/>

Das zentrale Impressum finden Sie unter <https://www.d-velop.de/impressum>.