



FormBuilder
Anleitung

Impressum

yourIT GmbH
Häselstr. 10
72336 Balingen

Fon: +49 7433 30098-0
Fax: +49 7433 30098-15
E-Mail: info@yourit.de
URL: <https://www.yourit.de>

Geschäftsführer:
Dipl. Inform. (FH) Ralf Ströbele
Dipl.-Kfm. Thomas Ströbele
Registergericht: Amtsgericht Stuttgart, HRB 421 035
Ust-ID: DE244014959
(Stand 12.05.2012)

Support

Sie erreichen unseren Support jeweils Montag bis Freitag von 8:00 bis 17:00 Uhr.

Produkt hotline

Fon +49 7433 30098-0
Fax +49 7433 30098-15
support@yourit.de

Revisionshistorie

Datum	Revision	Dokumentänderungen
2021-06-11	1	Erster Entwurf
2021-10-14	2	Ergänzung mit technischer Dokumentation

Inhalt

Grundlagen	4
Voraussetzungen	4
Datenaustausch	5
Erstellung von Formularen	6
Erstellen einer Formularvorlage	6
Vorgangformulare / Kundenformulare erstellen	8
Positionieren und Generieren	9
DOCUframe-Positionierung	9
Positionierung per Drag&Drop	9
Formular generieren	10
Datenspeicher	11
Zusatzfunktionen	12
Integration von Formularen in das Portal	12
Erstellung von Folgeformularen	12
Integration des Kunden oder der Aufgabe in das Formular	13
Änderung des Formularstils mit CSS	13
Basisdaten	14
Makros und ihre Aufgaben	21
Glossar	45

Grundlagen

Herzlich willkommen bei yourIT FormBuilder - einem Modul, das Sie in DOCUframe verwenden können.

Mit der yourIT FormBuilder können Sie Formulare einfach und schnell erfassen. Integrierte Funktionen zur Erstellung aller notwendigen Elemente helfen Ihnen, schnell und einfach sehr komplexe Formulare zu erstellen. Die mit dem erstellten Formular erhobenen Daten werden in DOCUframe gespeichert und Sie haben die Möglichkeit, diese Daten weiter zu verwenden.

Voraussetzungen

Für die Verwendung von yourIT FormBuilder gelten folgende Voraussetzungen:

- DOCUframe R3 von GSDSoftware oder höher.
- Eine aktive Internetverbindung über WLAN.
- Der FormBuilder Service muss auf einem Server erreichbar sein. Eventuell müssen Sie hierzu Anpassungen in der Firewall vornehmen.

Systemvoraussetzungen

Technische Voraussetzungen für die yourIT FormBuilder-App:

Web Browser

Plattform	Google Chrome, Safari
Betriebssystem	64-bit Windows 7/8/10 oder höher OS X El Capitan 10.11 oder höher
Prozessor	SSE3-fähiger Intel Pentium 4-Prozessor oder höher
Speicher	16Mb (sowie zusätzlicher Speicher für weitere Ressourcen bei Bedarf)
Internet-Browser	Google Chrome, Safari(v11.1 oder höher)

iPhone/iPad

Plattform	iOS
Betriebssystem	iOS 11.3 oder höher
Internet-Browser	Safari
Speicher	16Mb (sowie zusätzlicher Speicher für weitere Ressourcen bei Bedarf)

Android

Plattform	Android
-----------	---------

Betriebssystem	Android 5.0 (Lollipop)
Internet-Browser	Google Chrome für Android
Speicher	16Mb (sowie zusätzlicher Speicher für weitere Ressourcen bei Bedarf)

Datenaustausch

Wie gelangen die Daten in die Formularansicht des FormBuilder und wieder zurück?

Für den Datenaustausch zwischen der FormBuilder-App und Ihrem System ist der FormBuilder-Service verantwortlich. Er fordert diese Daten an und sendet sie wieder zurück an die Benutzeroberfläche.

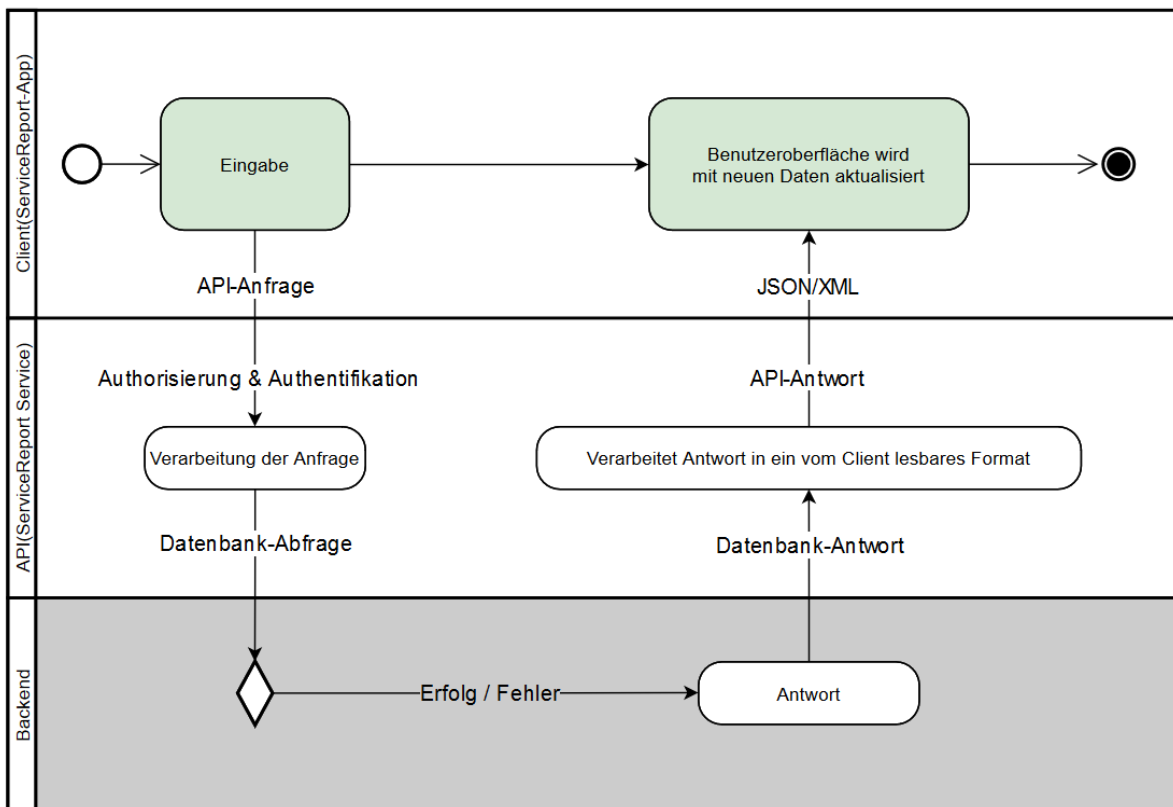


Abbildung 1. Datenaustausch über die API


Daher erstellen Sie im DOCUframe ein Formular und Elemente mit den notwendigen Funktionen wie Abmessungen, Wert, Position usw. Diese Informationen werden über **API** als **JSON**-Objekt an FormBuilder gesendet.

Der Dienst erstellt dann eine **HTML**-Datei mit der entsprechenden Darstellung und sendet den Namen dieser Datei zurück an DOCUframe. DOCUframe erstellt einen Link, auf dem die HTML-Datei visualisiert wird.

Erstellung von Formularen


Es gibt zwei Möglichkeiten, ein Formular zu erstellen. Eine besteht darin, eine Vorlage zu erstellen, deren Kopien Sie später für einen bestimmten Kunden verwenden. Die andere ist, direkt beim Kunden ein neues Formular zu erstellen. Sie können ein solches Formular nicht als Vorlage für andere Kunden verwenden.


Erstellen einer Formularvorlage

Unter **Formulare** **Formular-Vorlagen** sehen Sie eine Liste der bereits vorhandenen Formulare. In der oberen rechten Ecke befinden sich zwei Schaltflächen. Durch Klicken auf  können Sie eine PDF-Datei auswählen, die Formularfelder enthält. Auf diese Weise erstellen Sie ein Formular, das dieselben Elemente und Werte enthält, wie die ausgewählte PDF-Datei.

Name	Beschreibung	Elemente	Typ	Vorlage	Ersteller	Erstellungszeit
		303	PDF	✓ TRUE		10.10.2019 11:19:17
		303	PDF	✓ TRUE		10.10.2019 15:01:21
		6	HTML	✓ TRUE		14.10.2019 14:53:11
		0	PDF	✓ TRUE		14.10.2019 14:54:46
		0	PDF	✓ TRUE		14.10.2019 14:56:13
		0	PDF	✓ TRUE		14.10.2019 14:57:59
		303	PDF	✓ TRUE		14.10.2019 14:58:51
		303	PDF	✓ TRUE		14.10.2019 15:20:21
		139	PDF	✓ TRUE		18.10.2019 10:42:11
		302	PDF	✓ TRUE		17.10.2019 14:53:22
		302	PDF	✓ TRUE		17.10.2019 15:03:08
		302	PDF	✓ TRUE		17.10.2019 15:03:51

Abbildung 2. Liste der Formularvorlage

Durch Klicken auf  wird eine leere Formularvorlage erstellt, die dann in der Liste der Formularvorlagen angezeigt wird.

Um Elemente zu der neu erstellten Vorlage hinzuzufügen, klicken Sie auf die Vorlage. Durch Klicken auf  können Sie dann auswählen, welche Art von Element (z.B. Textbox, Kontrollkästchen...) zur Formularvorlage hinzugefügt werden soll.

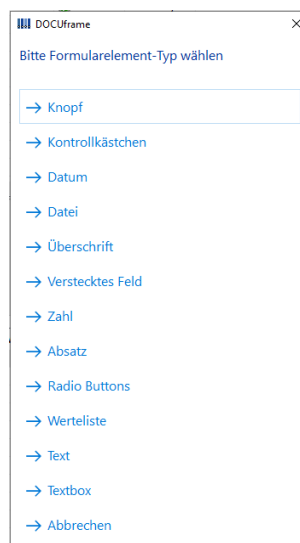


Abbildung 3. Elementtypauswahl



Wenn Sie sich nicht sicher sind, wie das Element aussieht, können Sie in der Liste der Formularvorlagen eine mit dem Namen "Alle Elemente" finden. Dieses Formular enthält alle Arten von Elementen.

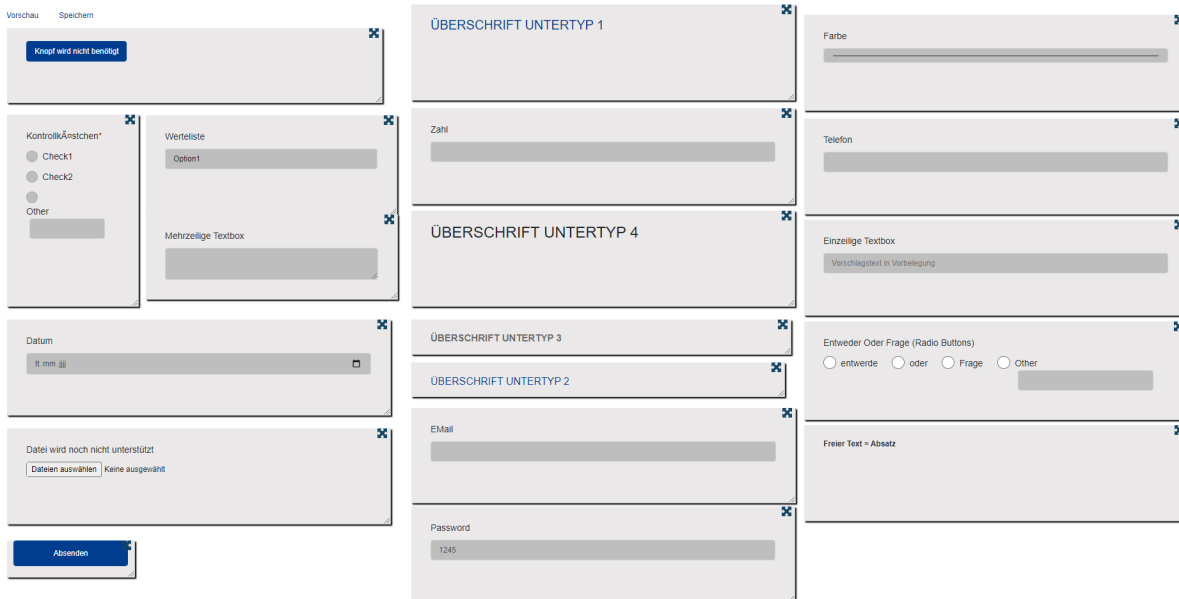


Abbildung 4. Form mit allen Arten von Elementen

Je nach Elementtyp können Sie die Eigenschaften dieses Elements anpassen.

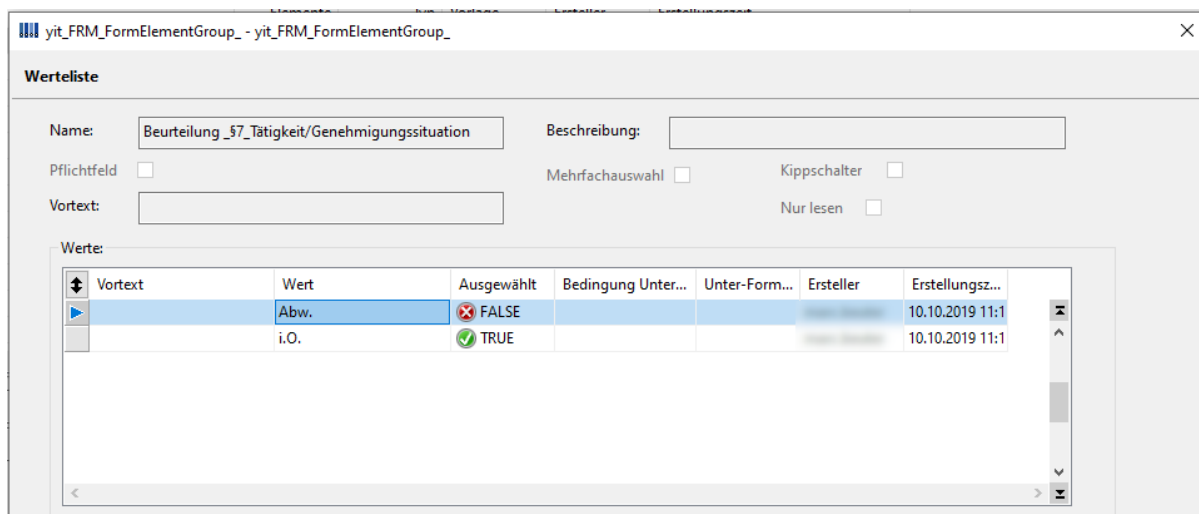







Abbildung 5. Elementdetails

Sie können beispielsweise der Werteliste die Optionen hinzufügen, die angezeigt werden sollen. Sie können festlegen, ob mehr als ein Feld ausgewählt werden kann, ob dieses Feld im Formular ein Pflichtfeld ist, ob dieses Feld schreibgeschützt ist usw.



Schaltflächen zum Arbeiten mit Formularelementen

-  ein neues Element hinzufügen
-  das ausgewählte Element löschen
-  ein Element nach oben scrollen
-  ein Element nach unten scrollen
-  automatische **Positionierung** von Elementen

Vorgangformulare / Kundenformulare erstellen


Jeder Kunde hat eine Registerkarte **Formulare**. Es gibt eine Liste aller Formulare, die der Kunde benötigt. Durch Klicken auf  können Sie eine der angebotenen Optionen auswählen. Sie können auch ein neues Formular ohne Vorlage erstellen - das Verfahren ist dasselbe wie beim Erstellen einer neuen Formularvorlage.



Abbildung 6. Reiterkarte beim Kunden

Wir können aber auch eine bereits erstellte Vorlage verwenden. In diesem Fall wird ein neues Formular erstellt, das mit der ausgewählten Vorlage identisch ist.



Die ObjectID der Vorlage und des Kundenformulars unterscheiden sich.

Bei Aufgabe gibt es auch eine Registerkarte "Formular" und auf die gleiche Weise können wir eine vorhandene Formularvorlage hinzufügen oder eine neue für diese Aufgabe erstellen.

Positionieren und Generieren

Es gibt zwei Möglichkeiten, Elemente in einem Formular zu positionieren. Wenn die Elemente gleich hoch und breit sein und untereinander liegen sollen, können Sie die automatische DOCUframe-Positionierung verwenden. Alternativ können die Elemente auch per Drag&Drop platziert werden.

DOCUframe-Positionierung

Wenn Sie auf  klicken, werden mehrere Dialoge geöffnet, in denen wir Folgendes angeben können:

- linke Startkoordinate des Formulars
- die obere Startkoordinate des Formulars
- Standardelementhöhe
- Standardelementbreite

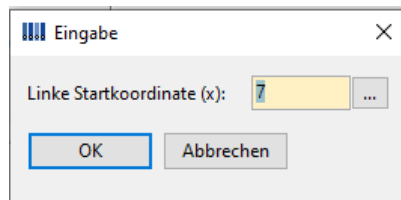


Abbildung 7. Positionierungsdialog



Hinweis:

Sie können die Koordinaten und Abmessungen des Elements auch direkt ändern, indem Sie mit der rechten Maustaste auf die gewünschte Zelle klicken und dann auf **Bearbeiten** klicken.

Positionierung per Drag&Drop

Oberhalb der Liste der Elemente befinden sich drei Schaltflächen

- **Formular generieren**
- **Downloadlink generieren**
- **Positionierung erstellen**

Wie der Name schon sagt, wird beim Klick auf **Positionierung erstellen** ein Link in der Textbox „Positionierungs-URL“ erstellt. Um eine Drag&Drop-Ansicht zu öffnen, kopieren Sie diesen Link in einen Internetbrowser.



Abbildung 8. Positionierungs-URL

Wenn Sie die Koordinaten der Elemente zuvor nicht angegeben haben, befinden sich alle Elemente in

der unteren rechten Ecke. Mit  können Sie die Elemente auf dem Formular bewegen und beliebig platzieren.



Die grauen Rahmen

Die grauen Rahmen auf den Elementen dienen nur der leichteren Positionierung. Im generierten Formular werden sie nicht angezeigt.

Wenn Sie überprüfen möchten, wie das generierte Formular aussehen würde, können Sie dies ganz einfach mit einem Klick auf [Vorschau](#) tun.

Einzeilige Textbox

Vorschlagstext in Vorbelegung

Mehrzeilige Textbox

Password

1245

E-Mail

Abbildung 9. Anzeige ohne graue Rahmen

Wenn Sie mit der Positionierung der Elemente fertig sind, können Sie auf [Speichern](#) klicken. Wenn Sie zu DOCUframe zurückkehren, sollten Sie neue Koordinatenwerte und Elementabmessungen sehen.

Ist dies nicht der Fall, versuchen Sie, die Anzeige mit der Taste [F5](#) zu aktualisieren.

Formular generieren

Nachdem die Elemente auf eine der beiden im vorherigen Kapitel beschriebenen Weisen positioniert wurden, erstellen Sie mit einem Klick auf [Formular generieren](#) einen Link in der Textbox „Formular-URL“.

Der Aufruf des generierten Links führt zu dem generierten Formular. Wir können den Link an den Kunden senden oder in ein Portal oder eine Website einbauen.

Datenspeicher

Nachdem der Kunde das Formular ausgefüllt und die Daten versendet hat, werden diese Daten gespeichert und sind auf dem Reiter **Auswertung** des angegebenen Formulars zu finden.

Formular-Elemente Kategorien Vorschau Auswertung _yit_FRM_Formular_Auswertung Versionshistorie				
Auswertung manuell		Formular aus Auswertung kopieren		
Name	Beschreibung	Typ	Vortext	Wert
text box	test	Textbox - Textarea		fff
test datum		Datum -		2021-06-09
zahl input		Zahl		1
name test	beschreibung test	Werteliste		option1
		Absatz -	blabla	
click		Knopf		rgregr

Abbildung 10. Anzeige von Antworten

Wenn der Kunde das Formular erneut ausfüllt, werden die aktuellen Daten durch neue ersetzt.

Zusatzfunktionen

Der FormBuilder ermöglicht uns viele weitere zusätzliche Optionen, um die erstellten Formulare besser zu verwalten. In diesem Kapitel werden wir einige davon auflisten und erklären.

Integration von Formularen in das Portal

Wie bereits im Kapitel [Datenaustausch](#) erwähnt, sendet DOCUframe alle notwendigen Informationen an den Java-Service, um eine HTML-Formulardatei zu erstellen. Der Service sendet die generierte HTML-Datei zurück an DOCUframe und erstellt einen Link, mit dem das Formular visualisiert wird.

Diesen Link für das Formular können wir über einen [iframe](#) Tag in das Portal oder die Website einbetten.

KUNDENSELBSTAUSKUNFT

Sehr geehrte Kundin,
sehr geehrter Kunde,

herzlich Willkommen bei der Kundenselbstauskunft der PÜG mbH. Mithilfe dieses Formulars prüfen wir jährlich Ihre Unternehmensdaten ab, um eine beständige Aktualität zu gewährleisten.

Sollte sich bei Ihren Daten etwas verändert haben, tragen Sie dies bitte in das dafür vorgesehene Feld unten ein. Gibt es keine Änderungen, dann bitte das Feld freilassen.

Vielen Dank für Ihre Unterstützung!

The image shows a web form for customer self-service. It consists of several input fields arranged in a grid-like structure. The fields are: 'Kundennummer:' (a small text box), 'Name des Unternehmens:*' (a long text box), 'Straße:' (a long text box), 'Hausnummer:' (a medium text box), 'PLZ:' (a small text box), 'Ort:' (a long text box), 'Telefonnummer:' (a long text box), and 'Faxnummer:' (a long text box). Each field has a light blue header and a grey input area.

Abbildung 11. ein im Portal eingebettetes Formular

Erstellung von Folgeformularen

Sie können mehrere Formulare verknüpfen, sodass, wenn ein Kunde eines ausfüllt, das andere Formular automatisch geladen wird. Nachdem der Kunde in DOCUframe den Submit-Button geklickt hat, beginnt der Prozess, in dem das [JSON](#)-Objekt erstellt wird. Dieses [JSON](#)-Objekt enthält Informationen, ob das folgende Formular existiert und wenn ja, welches es ist.


Es kann immer die gleiche Reihenfolge sein, kann aber beispielsweise von den Antworten aus dem vorherigen Formular abhängen.



Wenn Sie das Formular in ein Portal oder eine Website integrieren oder Folgeformulare erstellen möchten, kontaktieren Sie uns bitte.

Integration des Kunden oder der Aufgabe in das Formular

Wenn Sie zuvor Vorgangsformulare oder Kundenformulare erstellt haben und Sie möchten, dass die Elemente mit Werten aus dem zugeordneten Objekt (d.h. dem entsprechenden Kunden oder der entsprechenden Aufgabe) gefüllt werden, dann folgen Sie den nächsten Schritten:

- öffnen Sie das Formular
- wählen Sie Bezugsklasse
- Doppelklick auf ein Element
- klicken Sie unter Stammdaten auf 
- Wählen Sie die Kundeneigenschaft oder Aufgabeeigenschaft aus, für die Sie den Wert des ausgewählten Elements im Formular wünschen

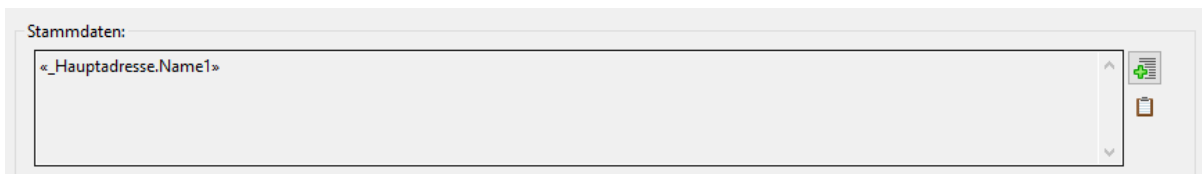



Abbildung 12. Element mit Stammdaten

Nachdem Sie das Formular generiert haben, sehen Sie, dass die Elemente mit Stammdaten die Standardeigenschaften des Objekts haben.

Änderung des Formularstils mit CSS

Sie können den Stil der Elemente im Formular selbst ändern. Dazu klicken Sie auf [Formulare](#)  [CSS Bearbeiten](#), daraufhin wird Ihnen die Benutzeroberfläche zum Arbeiten mit CSS-Stilen Ihrer Formulare angezeigt.

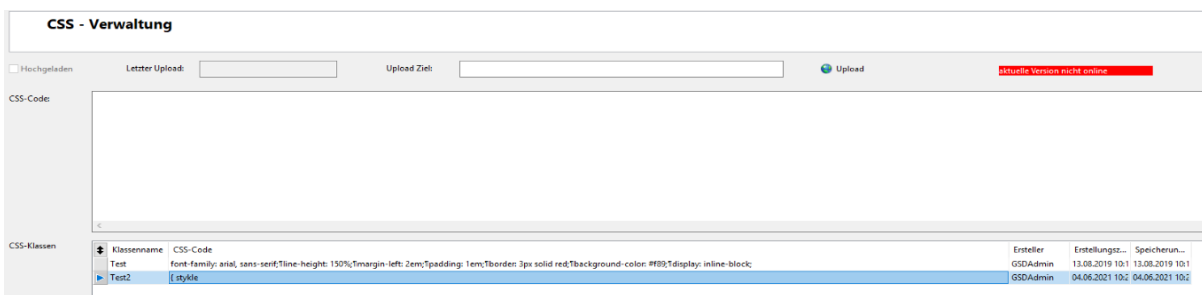



Abbildung 13. Benutzeroberfläche zum Arbeiten mit CSS-Stilen

Hier wird eine Liste aller zuvor erstellten CSS-Klassen angezeigt. Durch Doppelklick auf eine CSS-Klasse können Sie den CSS-Code ändern und mit einem Klick auf  eine neue Klasse erstellen.

Unter der Eigenschaft "Klasse" können Sie die erstellten Klassen zu den Formularelementen hinzufügen. Dann werden die Stile, die Sie in die Klasse eingegeben haben, auf die ausgewählten Elemente

angewandt.

Formular-Elemente						
Kategorien						
Vorschau						
Auswertung						
_yit_FRM_Formular_Auswertung						
Versionshistorie						
Name	Beschreibung	Typ	Vortext	Wert	Klasse	
test box	test	Textbox - Textarea			Test	
test datum		Datum -			Test	
zahl input		Zahl		0	Test	
name test	beschreibung test	Werteliste			Test	
		Absatz -	blable		Test	
click		Knopf		rgregr	Test	

Abbildung 14. Elemente mit CSS-Klasse



Wenn die von Ihnen geladenen Stile nicht auf die Elemente angewendet werden, versuchen Sie, dem CSS-Stil die Eigenschaft **important** hinzuzufügen. Beispiel: Anstelle von `background: black;` schreiben Sie `background: black !important;`.

Basisdaten

Wenn Ihnen die Daten einzelner Elemente sehr wichtig sind und Sie deren Wert im DOCUframe-Feld des Objekts, mit dem das Formular verknüpft ist, speichern oder den Wert dieses Feldes ändern möchten, sind Basisdaten hilfreich. Außerdem können Sie anhand der Basisdaten einen bestimmten Prozess in Abhängigkeit von dem im Formular eingegebenen Wert starten.

Beim Klick auf [Formulare](#) [Basisdaten-Vorlagen](#) wird Ihnen eine Liste der bisher erstellten Basisdaten angezeigt. Ein Klick auf erzeugt eine neue Basisdatei. Sie können eine beliebige Nummer und einen beliebigen Namen eingeben. Wählen Sie bei [Werttyp](#) den Elementtyp aus, zu dem Sie diese Basisdatei hinzufügen möchten.

Geben Sie den Prozessnamen in die Eingabe mit der Bezeichnung "Prozess" ein. Dieser Prozess wird nach dem Absenden des Formulars aufgerufen. Weitere Informationen zu den Prozessen in den Basisdaten finden Sie im Kapitel [Basisdaten und Prozesse](#).

Basidaten Vorlage - 0_

Nummer:
 Name:
 Werttyp:
 Prozess:
 Überschreiben

Abbildung 15. Informationen zur Basisdatei

Wenn Sie das Kontrollkästchen "Überschreiben" markieren, ersetzt der neue Wert den alten, das heißt, Sie haben nur Zugriff auf den neuesten Basisdatenwert. Wenn Ihnen alle Werte dieser Basisdaten wichtig sind, dann lassen Sie das Kontrollkästchen unmarkiert.

Example 1. Kontrollkästchen "Überschreiben" markieren

Stellen Sie sich beispielsweise das folgende Szenario vor. Sie haben die Basisdatei *Anzahl Mitarbeiter* angelegt. Im ersten Jahr waren es 50, aber im Folgejahr änderte der Kunde beim Ausfüllen des Formulars die Mitarbeiterzahl auf 60.

Wenn Sie beim Anlegen dieser Basisdatei **Überschreiben** markiert haben, wird der Wert der Basisdatei *Anzahl Mitarbeiter* 60 sein.

Wenn das Kontrollkästchen unmarkiert geblieben ist, haben Sie jetzt einen neuen Wert, 60, im Basisdateifeld `yit_FRM_Wert_` gespeichert, und im Feld `yit_FRM_Archiv_Objekt_` befindet sich ein altes Basisdateiobjekt mit dem Wert 50, das Sie gerade geändert haben.

Wir können nun auf ein Element in der Elementliste des ausgewählten Formulars doppelklicken, um die Details zu öffnen. Gehen Sie auf den Reiter **Basisdaten** und wählen Sie dort mit einem Klick auf **Basisdaten auswählen** die zuvor erstellte Basisdatei aus.

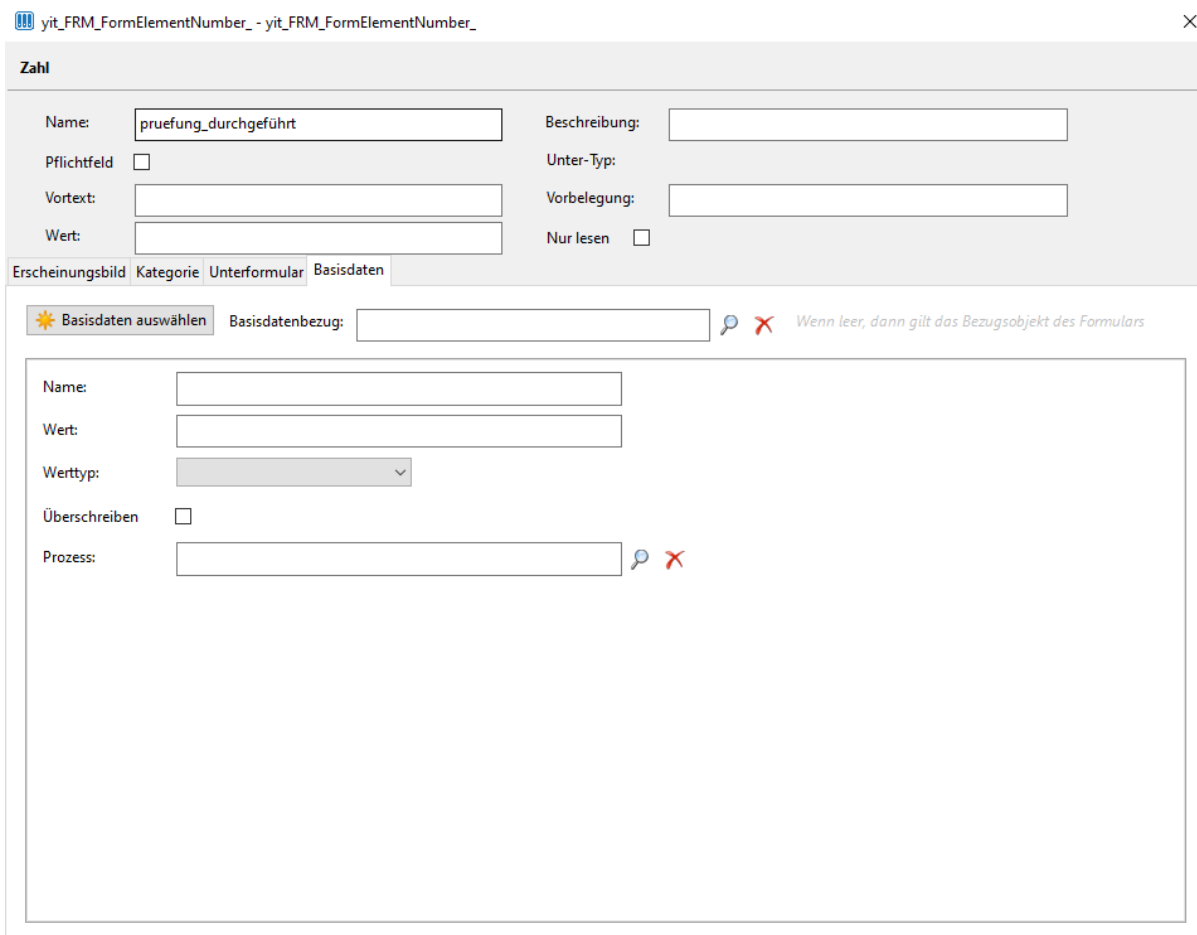



Abbildung 16. Hinzufügen einer Basisdatei zu einem Element

Basisdaten und Prozesse

Im DOCUframe in der Symbolleiste öffnen Sie mit einem Klick auf **Einstellungen** **Prozesse** eine Liste aller Prozesse. Dies sind Funktionen, die automatisch aufgerufen werden, wenn bestimmte Bedingungen erfüllt sind.

Wenn Sie in Abhängigkeit vom Wert des Basisdatenfeldes eine bestimmte Funktion aufrufen möchten, legen Sie zunächst einen neuen Prozess an. Wählen Sie den entsprechenden Prozessnamen aus und tragen Sie ihn in das Feld **Name** ein.

Klicken Sie im Bereich **Regeln** auf , um eine neue Regel hinzuzufügen. Jede Regel stellt einen Fall dar, auf den der Prozess unterschiedlich reagieren soll. Jede Regel hat also **Bedingungen** und **Aktionen**. Wenn die Bedingungen erfüllt sind, werden Aktionen aufgerufen.

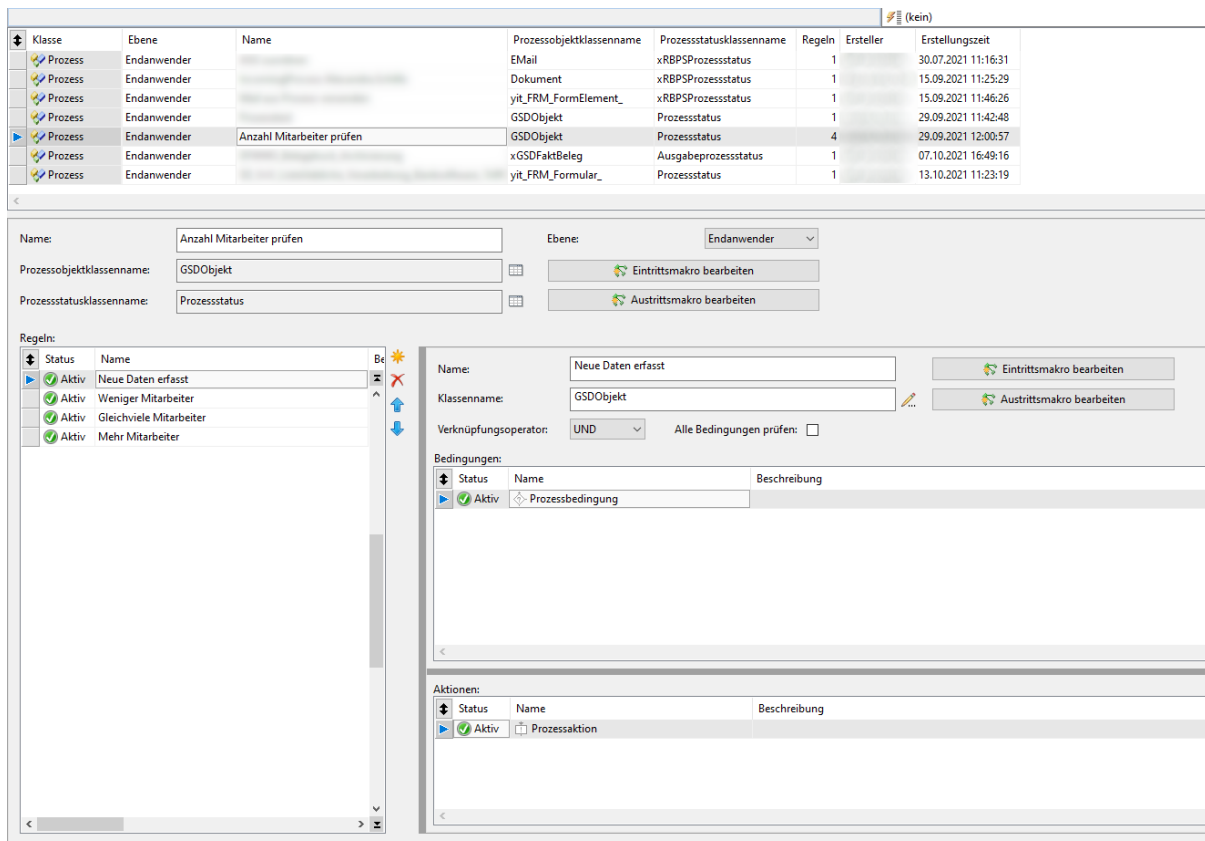



Abbildung 17. Prozesserstellung

Wenn Sie beispielsweise dem Element *Anzahl der Mitarbeiter* im Formular eine Basisdatei hinzugefügt haben und auf eine Wertänderung reagieren möchten, wäre eine gute Wahl der Regeln:

- weniger Mitarbeiter
- mehr Mitarbeiter
- genauso viele Mitarbeiter wie letztes Jahr
- erste Dateneingabe

Nachdem Sie die Regeln erstellt haben, können Sie Bedingungen hinzufügen. Unter der Regel *weniger Mitarbeiter* sollen Sie prüfen, ob der neue Basisdatenwert kleiner ist als der vorherige.

Klicken Sie im Bereich **Bedingungen** auf . Wählen Sie aus der Liste der Bedingungen **Prozessbedingung** aus. Öffnen Sie die neu erstellte Bedingung mit Doppelklick und klicken Sie auf **Makro bearbeiten**, um die Bedingungsfunktion zu schreiben.



Sie können verschiedenen Bedingungen und Aktionen auswählen.

Der Makroeingabeparameter ist ein Objekt vom Typ **Prozessstatus**. Dieses Objekt enthält (unter anderem) ein Feld namens **Prozessobjekt**. Dort wird ein Objekt vom Typ **yit_FRM_FormElementText_** abgelegt. Dies ist eigentlich ein Element des Formulars, dem wir die Basisdatei beigefügt haben. Dieses Formularelement hat folgende Felder:


- **yit_FRM_Value_** - aktueller Wert
- **yit_FRM_Source_** - vorheriger Wert

So können Sie ganz einfach feststellen, ob die Mitarbeiterzahl gestiegen, gesunken oder gleich geblieben ist. Ein Beispiel für eine *weniger Mitarbeiter*-Bedingung könnte sein:

Example 2. Weniger Mitarbeiter - Bedingung

```
BOOL ProcessCondition( Prozessstatus &Status )
  yit_FRM_FormElementText_ oElement,oSource;
  STRING Value;
  BOOL bRet;

  oElement = Status.Prozessobjekt;
  oSource = oElement.yit_FRM_Source_;
  Value = oElement.yit_FRM_Value_;
  IF(oSource.yit_FRM_Value_.ToInt() > oElement.yit_FRM_Value_.ToInt())
    bRet = TRUE;
  ELSE
    bRet = FALSE;
  ENDIF
  RETURN( bRet );
```

Jetzt schreiben Sie nur noch ein Makro, das ausgeführt wird, wenn die Funktion aus der **Bedingung** **TRUE** zurückgibt. Klicken Sie im Feld **Aktionen** auf  und falls Sie ein Makro schreiben möchten, wählen Sie **Prozessaktion**.

Dieses Makro hat den gleichen Eingabeparameter wie das Makro unter **Bedingungen** und Sie können leicht auf den Wert des Formularelements zugreifen.



Ein wichtiges Feld in der Klasse **yit_FRM_FormElementText_** ist **yit_FRM_Basisdaten_Bezugsobjekt_**, in dem das Referenzobjekt für dieses Feld gespeichert wird.

Das Bezugsobjekt ist ein Objekt, das wir mit dem Formularelement verknüpft haben, indem wir im Formular **Bezugsklasse** und dann im Formularelement **Bezugselement** auswählen. Dies wird im Kapitel [Basisdaten in einer Datenbank speichern](#) näher erläutert.

Ein gutes Aktionsbeispiel für die *Weniger Mitarbeiter*-Regel wäre:

Example 3. Weniger Mitarbeiter - Aktion

```
BOOL ProcessAction( Prozessstatus &Status )
    Mitteilung oM;
    DBSTRINGSET To;
    yit_FRM_FormElement_ Element = Status.Prozessobjekt;
    Kunde customer = Element.yit_FRM_Basisdaten_Bezugsobjekt_;

    To.Add("test_username");
    oM.Send(To,"Die Mitarbeiterzahl hat sich im Vergleich zum Vorjahr
    verringert. Kunde: " + customer.Name1 + " " + customer.Name2,
    "", "");
    RETURN( TRUE );
```

Basisdaten in einer Datenbank speichern

Mit Basisdaten können Sie wichtige Informationen in einer Datenbank speichern. Sie möchten beispielsweise die "Anzahl Mitarbeiter" im Objekt **Kunde** speichern, dieses Feld existiert jedoch nicht in der DOCUframe-Datenbank. Um nicht ständig die Datenbank ändern und Felder hinzufügen zu müssen, können wir hierfür Basisdaten verwenden.

Nachdem Sie die Vorlage für das Formular erstellt haben, fügen Sie die Elemente und die **Bezugsklasse** hinzu. Sie können dieses Formular mit einem Objekt der Klasse verknüpfen, die in der **Bezugsklasse** aufgeführt ist. Darüber haben wir bereits im Kapitel [Integration des Kunden oder der Aufgabe in das Formular](#) geschrieben.

Auch nachdem Sie eine Vorlage für die Basisdaten erstellt haben, vergeben Sie einen passenden Namen und Typ. Diese Basisdatei können Sie dem Formularelement hinzufügen. Darüber haben wir auch im Kapitel [Basisdaten](#) geschrieben.

Doppelklicken Sie auf das ausgewählte Formularelement und öffnen Sie das Register **Basisdaten**. Hier ist das Feld **Basisdatenbezug**. Wenn dieses Feld leer bleibt, ist das Bezugsobjekt für das Formularelement das Objekt, zu dem Sie das Formular hinzugefügt haben.



Mit anderen Worten, Sie speichern die Basisdaten als Feld in dem Objekt, mit dem das Formular verknüpft ist. Falls dies nicht gewünscht ist, können Sie auch ein Unterobjekt des Formularbezugsobjekts auswählen. In diesem Kapitel erklären wir, wie und geben einige Beispiele.

Sie haben beispielsweise ein Formular zu einem Objekt der Klasse **Aufgabe** hinzugefügt, möchten aber die Basisdaten **Anzahl der Mitarbeiter** zu einem Objekt der Klasse **Kunde** speichern, das auf der Aufgabe unter dem Feld **_Hauptadresse** gespeichert wird. Dann haben Sie im Formularfeld **Bezugsklasse** **Aufgabe** und im Feld **Basisdatenbezug** im Elementfeld unter dem Reiter **Basisdaten** **_Hauptadresse**.



Jedes Objekt hat ein Feld `yit_FRM_Basisdaten_` und dort können beliebige Basisdaten gespeichert werden. Dieses Feld enthält eine **Liste von Objekten** der Klasse `yit_FRM_Basisdaten_`. Jedes Objekt dieser Klasse enthält unter anderem folgende Felder:

- `yit_FRM_Name_` - der Name, den wir den Basisdaten gegeben haben
- `yit_FRM_Wert_` - der Wert des Elements, zu dem die Basisdaten hinzugefügt wurden

Eine mögliche Situation ist beispielsweise: Beim Kunden unter der Nummer "K-20841" hat das erste Element des Feldes `yit_FRM_Basisdaten_` die Felder:

- `yit_FRM_Name_`: "Anzahl Mitarbeiter"
- `yit_FRM_Wert_`: 50

Diese Basisdatenfelder haben Sie ausgefüllt, indem Sie ein Formular mit dem Bezugsobjekt `Aufgabe` mit der Nummer "AUF-034117" ausgefüllt haben.

Makros und ihre Aufgaben

In diesem Kapitel finden Sie eine Liste der von uns verwendeten Makros und eine Erläuterung ihrer Aufgaben. Die Makros sind alphabetisch geordnet.

yit_FRM_Add_Basisdaten_To_FormElement_

Bezeichnung

- erstellt neue Basisdaten aus der ausgewählten Basisdatenvorlage und hängt sie an das Formularelement an

Parameter

- `yit_FRM_FormElement_ &oElement` - Element, dem die Basisdatei zugeordnet wird
- `yit_FRM_Basisdaten_Vorlage_ oVorlage` - eine Basisdatenvorlage, aus der die Basisdatei erstellt und das Element aus dem Formular hinzugefügt wird
- `STRING &sFehlertext` - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt

Rückgabewert

- `INT iRC` - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung

yit_FRM_Add_CSS_Class_

Bezeichnung

- erstellt eine neue CSS-Klasse und fügt sie der CSS-Datei hinzu

Parameter

- `HDIALOG hDialog` - der Dialog, zu dem dieses Makro hinzugefügt wird
 - wird verwendet, um die Klassenliste zu aktualisieren, wenn eine neue Klasse erfolgreich hinzugefügt wurde
- `yit_FRM_CSS_File_ &oCSS` - CSS-Datei, zu der eine neue Klasse hinzugefügt wird und das Element aus dem Formular hinzugefügt wird
- `STRING &sFehlertext` - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt

Rückgabewert

- `INT iRC` - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung
 - -1 - der Benutzer hat einen bereits vorhandenen Klassennamen ausgewählt

yit_FRM_Auswertung_Manuell

Bezeichnung

- kopiert aktuelle Werte aus Formularelementen und speichert diese in der Formularauswertung

Parameter

- `yit_FRM_Formular_ &oForm` - ein Formular, in dessen Auswertung aktuell die Werte seiner Elemente gespeichert sind
- `STRING &sFehlertext` - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt

Rückgabewert

- `INT iRC` - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung
 - 2 - Fehler beim Kopieren des Formularelements

yit_FRM_Btn_Add_Basisdaten_

Bezeichnung

- öffnet einen Dialog zur Auswahl einer Basisdatenvorlage, erstellt die Basisdaten aus dieser Vorlage und speichert sie im aktuellen Element

Parameter

- `yit_FRM_FormElement_ &oElement` - Element, dem die Basisdatei zugeordnet wird
- `STRING &sFehlertext` - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt

Rückgabewert

- `INT iRC` - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung
 - 1 - der Benutzer hat den Vorgang abgebrochen

yit_FRM_Btn_Add_Process_

Bezeichnung

- öffnet einen Prozessauswahldialog und ordnet den ausgewählten Prozess den Basisdaten zu

Parameter

- `yit_FRM_Basisdaten_ &oBasisdaten` - die Basisdaten, an die der Prozess anschließt
- `STRING &sFehlertext` - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt

Rückgabewert

- **INT** iRC - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung
 - 1 - der Benutzer hat den Vorgang abgebrochen

yit_FRM_Build_CSS_File

Bezeichnung

- konvertiert Text aus CSS-Kategoriefeldern in gültigen CSS-Code

Parameter

- **yit_FRM_CSS_File_** &oFile - css-Dateiobjekt, aus dem die erstellten Kategorien gelesen werden
- **STRING** &sCode - CSS-Code durch Konvertierung erhalten
- **STRING** &sFehlertext - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt

Rückgabewert

- **INT** iRC - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung

yit_FRM_Build_FormElementButton_

Bezeichnung

- erstellt ein Button-JSON-Objekt und fügt dieses Element der Liste der Formularelemente hinzu

Parameter

- **yit_FRM_Formular_** oForm - ein Formular, dem eine Schaltfläche hinzugefügt wird
 - spielt in diesem Makro keine Rolle, nur zum Debuggen
- **yit_FRM_FormElementButton_** &oButton - ein Buttonobjekt, das dem Formular hinzugefügt wird
- **HJSON** &hData - eine Liste von JSON-Objekten, die Formularelemente bezeichnen
- **STRING** &sFehlertext - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt

Rückgabewert

- **INT** iRC - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung

yit_FRM_Build_FormElementFile_

Bezeichnung

- erstellt ein "upload file" JSON-Objekt und fügt dieses Element der Liste der Formularelemente hinzu

Parameter

- `yit_FRM_Formular_oForm` - ein Formular, dem ein Dateifeld hinzugefügt wird
 - spielt in diesem Makro keine Rolle, nur zum Debuggen
- `yit_FRM_FormElementFile_oFile` - ein Dateiobjekt, das dem Formular hinzugefügt wird
- `HJSON &hData` - eine Liste von JSON-Objekten, die Formularelemente bezeichnen
- `STRING &sFehlertext` - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt

Rückgabewert

- `INT iRC` - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung

`yit_FRM_Build_FormElementGroup_`

Bezeichnung

- erstellt ein JSON-Group-Objekt und fügt dieses Element der Liste der Formularelemente hinzu
 - Group-Objekt kann sein: checkbox-group, radio-group oder select-group.

Parameter

- `yit_FRM_Formular_oForm` - ein Formular, dem Grroupelemente hinzugefügt werden
 - spielt in diesem Makro keine Rolle, nur zum Debuggen
- `yit_FRM_FormElementGroup_oGroup` - ein Groupobjekt, das dem Formular hinzugefügt wird
- `HJSON &hData` - eine Liste von JSON-Objekten, die Formularelemente bezeichnen
- `STRING &sFehlertext` - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt

Rückgabewert

- `INT iRC` - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung

`yit_FRM_Build_FormElementGroup_PDF`

Bezeichnung

- erstellt ein JSON-Group-Objekt aus einem geladenen PDF-Dokument und fügt dieses Element der Liste der Formularelemente hinzu

Parameter

- `yit_FRM_Formular_` oForm - ein Formular, dem Groupnelemente hinzugefügt werden
 - spielt in diesem Makro keine Rolle, nur zum Debuggen
- `yit_FRM_FormElementGroup_` &oGroup - ein Groupobjekt, das dem Formular hinzugefügt wird
- `HJSON` &hData - eine Liste von JSON-Objekten, die Formularelemente bezeichnen
- `STRING` &sFehlertext - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt

Rückgabewert

- `INT` iRC - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung

`yit_FRM_Build_FormElementNumber_`

Bezeichnung

- erstellt ein NummerInput JSON-Objekt und fügt dieses Element der Liste der Formularelemente hinzu

Parameter

- `yit_FRM_Formular_` oForm - ein Formular, dem das Zahlenfeld hinzugefügt wird
 - spielt in diesem Makro keine Rolle, nur zum Debuggen
- `yit_FRM_FormElementNumber_` &oNumber - ein Nummerobjekt, das dem Formular hinzugefügt wird
- `HJSON` &hData - eine Liste von JSON-Objekten, die Formularelemente bezeichnen
- `STRING` &sFehlertext - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt

Rückgabewert

- `INT` iRC - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung

`yit_FRM_Build_FormElementText_`

Bezeichnung

- erstellt ein TextInput JSON-Objekt und fügt dieses Element der Liste der Formularelemente hinzu
 - Textobjekt kann sein: Datum, Kopfzeile, Absatz, Textbereich oder verstecktes Element

Parameter

- `yit_FRM_Formular_` oForm - ein Formular, dem Text hinzugefügt wird
 - spielt in diesem Makro keine Rolle, nur zum Debuggen
- `yit_FRM_FormElementText_` &oText - ein Textobjekt, das dem Formular hinzugefügt wird

- **HJSON** &hData - eine Liste von JSON-Objekten, die Formularelemente bezeichnen
- **STRING** &sFehlertext - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt

Rückgabewert

- **INT** iRC - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung

yit_FRM_Build_FormElementText_PDF_

Bezeichnung

- erstellt ein Text-JSON-Objekt aus einem geladenen PDF-Dokument und fügt dieses Element der Liste der Formularelemente hinzu
 - Textobjekt kann sein: Datum, Kopfzeile, Absatz, Textbereich oder verstecktes Element

Parameter

- **yit_FRM_Formular_** oForm - ein Formular, dem Text hinzugefügt wird
 - spielt in diesem Makro keine Rolle, nur zum Debuggen
- **yit_FRM_FormElementText_** &oText - ein Textobjekt, das dem Formular hinzugefügt wird
- **HJSON** &hData - eine Liste von JSON-Objekten, die Formularelemente bezeichnen
- **STRING** &sFehlertext - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt

Rückgabewert

- **INT** iRC - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung

yit_FRM_Build_Formular_

Bezeichnung

- konvertiert alle Elemente aus dem Formular in ein JSON-Objekt

Parameter

- **yit_FRM_Formular_** &oForm - Formular, dessen Elemente umgewandelt werden
- **HJSON** &hData - eine Liste der konvertierten JSON-Formularelementen
- **STRING** &sFehlertext - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt

Rückgabewert

- **INT** iRC - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung

yit_FRM_Build_Formular_PDF

Bezeichnung

- wandelt alle Elemente aus dem PDF erhaltenen Formular in ein JSON-Objekt um

Parameter

- **yit_FRM_Formular_** &oForm - Formular, dessen Elemente umgewandelt werden
- **HJSON** &hData - eine Liste der konvertierten JSON-Formularelementen
- **STRING** &sFehlertext - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt

Rückgabewert

- **INT** iRC - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung

yit_FRM_Build_Stammdaten_String_

Bezeichnung

- ändert den Wert des **sValue**, der den Memberpfad enthält, mit dem richtigen Wert dieses Felds im Bezugsobjekt

Parameter

- **DBOBJECT** oBezugsobjekt - das Objekt, dessen Feldwert gesucht wird
- **STRING** &sValue- eine Zeichenfolge, die anfangs die Zeichenfolge enthält: [\[Memberpfad\]](#) und nach der Ausführung des Makros den wahren Wert des Felds im Referenzobjekt enthält, zu dem der Memberpfad führt
- **STRING** &sFehlertext - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt

Rückgabewert

- **INT** iRC - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung
 - -1 - Der Membername des Bezugsobjekt existiert nicht; der angegebene Memberpfad existiert nicht

yit_FRM_CheckInsert_OID_

Bezeichnung

- prüft, ob ein Element mit dem Namen "OID" im Formular vorhanden ist, falls es nicht existiert - fügt das Element "OID" und andere Elemente aus dem gesendeten PDF hinzu

Parameter

- **yit_FRM_Formular_** &oForm - ein Formular, zu dem Elemente hinzugefügt werden, wenn kein Element namens "OID" vorhanden ist
- **PDF** &oPDFIn- PDF, aus dem die Elemente geladen werden
- **STRING** &sFehlertext - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt

Rückgabewert

- **INT** iRC - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung
 - 1 - Fehler beim Erstellen des PDF-Elements

yit_FRM_Create_Formular_

Bezeichnung

- erstellt ein neues Formular oder eine Formularvorlage

Parameter

- **yit_FRM_Formular_** &oForm - das Objekt, in dem das neue Formular gespeichert wird
- **BOOL** bVorlage
 - wenn bVorlage **TRUE** ist, wird eine neue Formularvorlage erstellt
 - wenn es **FALSE** ist, wird ein neues Formular erstellt
- **STRING** &sFehlertext - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt
- **DBOBJECT** oBezugsobjekt
 - Referenzobjekt, das mit dem neu erstellten Formular verknüpft wird
 - Optionaler Parameter mit dem Standardwert **null**

Rückgabewert

- **INT** iRC - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung
 - 1 - der Benutzer hat den Vorgang abgebrochen

yit_FRM_Copy_Form_

Bezeichnung

- kopiert Elemente eines Formulars in ein anderes

Parameter

- **yit_FRM_Formular_** oSourceForm - das Formular, aus dem es kopiert wurde

- `yit_FRM_Formular_ &oDestForm` - das Formular, in das es kopiert wird
- `STRING &sFehlertext` - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt
- `BOOL bFromAuswertung`
 - wenn `bFromAuswertung TRUE` ist, wird es aus der Auswertung kopiert
 - wenn `bFromAuswertung FALSE` ist, wird es von den Formularelementen kopiert
 - Optionaler Parameter mit dem Standardwert `FALSE`

Rückgabewert

- `INT iRC` - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung

yit_FRM_Copy_FormElement

Bezeichnung

- kopiert ein Formularelement in ein anderes Formularelement

Parameter

- `yit_FRM_FormElement_ &oSource` - das Formularelement, aus dem es kopiert wurde
- `yit_FRM_FormElement_ &oTarget` - das Formularelement, in das es kopiert wird
- `DBSTRINGSET &ElementValues` - der Wert des Formularelements
 - in Groupbox - ein Set von markierten Labels
- `STRING &sFehlertext` - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt

Rückgabewert

- `INT iRC` - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung

yit_FRM_Create_Download_Link_

Bezeichnung

- erstellt ein PDF-Element aus dem Formular, speichert den Link zum PDF im Formularfeld `yit_FRM_Link`

Parameter

- `yit_FRM_Formular_ &oForm` - Formular, dessen PDF und Link dazu erstellt wird
- `STRING &sFehlertext` - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt

Rückgabewert

- **INT** iRC - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung
 - -1 - Fehler bei der Erstellung des Formulars

yit_FRM_Create_Form_From_JSON_

Bezeichnung

- erstellt ein neues Formular oder eine neue Formularvorlage aus einem JSON-Objekt

Parameter

- **HJSON** hResponse - JSON, aus dem ein neues Formular erstellt wird
- **yit_FRM_Formular_** &oForm - neu erstelltes Formular
- **BOOL** bVorlage
 - wenn bVorlage **TRUE** ist, wird eine Formularvorlage erstellt
 - wenn bVorlage **FALSE** ist, wird ein Formular erstellt
- **STRING** &sFehlertext - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt

Rückgabewert

- **INT** iRC - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung
 - -1 - OID konnte nicht geschrieben werden

yit_FRM_Create_FormElement_

Bezeichnung

- öffnet eine Liste von Elementtypen, nachdem der Elementtyp ausgewählt wurde, erstellt ihn und fügt ihn dem Formular hinzu

Parameter

- **yit_FRM_Formular_** &oForm - ein Formular, dem ein neues Element hinzugefügt wird
- **STRING** &sFehlertext - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt - **HDIALOG** hDialog - der Dialog, zu dem dieses Makro hinzugefügt wird
 - wird verwendet, um die Elementliste zu aktualisieren, wenn ein neues Element erfolgreich hinzugefügt wurde
- **STRING** sListName - die Kennung des Feldes, in dem sich die Liste der Elemente befindet

Rückgabewert

- **INT** iRC - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde

- 0 - erfolgreiche Ausführung
- -1 - der Benutzer hat den Vorgang abgebrochen

yit_FRM_Create_FormElement_ByCopy

Bezeichnung

- erstellt neue Elemente, die durch Klonen erstellt wurden

Parameter

- `yit_FRM_FormElement_ &oSource` - Element, das geklont und kopiert wird
- `yit_FRM_FormElement_ &oNewFormElement` - ein neues Element im Formular, das durch Klonen erstellt wurde - `yit_FRM_FormElement_ &oTarget` - ein Element zum Speichern der Auswertung
- `DBSTRINGSET &ElementValues` - der Wert des Formularelements
 - in Groupbox - ein Set von markierten Labels
- `DBSTRINGSET &Coordinates` - Position des neu erstellten Elements
- `STRING &sFehlertext` - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt

Rückgabewert

- `INT iRC` - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung

yit_FRM_CreateUpdate_FormElement_From_Response_

Bezeichnung

- erstellt oder aktualisiert Formularelemente aus dem erhaltenen JSON-Objekt

Parameter

- `yit_FRM_Formular_ &oForm` - ein Formular, dessen Elemente aktualisiert / erstellt werden
- `HJSON hChild` - JSON-Objekt, aus dem neue Elemente erstellt oder alte Elemente aktualisiert werden
- `STRING &sFehlertext` - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt

Rückgabewert

- `INT iRC` - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung

yit_FRM_CSS_Select_Class_

Bezeichnung

- öffnet den Dialog zur Auswahl der CSS-Klasse und ordnet die ausgewählte Klasse dem aktuellen Element zu

Parameter

- `yit_FRM_FormElement_` &oElement - Element, zu dem wir die CSS-Klasse hinzufügen

Rückgabewert

- `INT` iRC - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung

yit_FRM_Del_CSS_Class_**Bezeichnung**

- entfernt die CSS-Klasse aus der CSS-Datei

Parameter

- `HDIALOG` hDialog - der Dialog, zu dem dieses Makro hinzugefügt wird
 - wird verwendet, um die Klassenliste zu aktualisieren
- `yit_FRM_CSS_File_` &oCSS - CSS-Datei, aus der die CSS-Klasse entfernt wird
- `STRING` &sFehlertext - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt

Rückgabewert

- `INT` iRC - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung
 - -1 - Keine Klasse ausgewählt

yit_FRM_FormElement_Open_Dialog_Modal**Bezeichnung**

- findet den Wert des Elements des Referenzobjekts unter dem angegebenen Pfad

Parameter

- `yit_FRM_FormElement_` oElement - das Element, dessen Dialog geöffnet wird
- `yit_FRM_Formular_` oForm
 - Ein Referenzobjekt, das einem offenen Element zugeordnet ist
 - Optionaler Parameter mit dem Standardwert `null`

Rückgabewert

- **INT** iRC - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung

yit_FRM_Get_Formulare_By_Object

Bezeichnung

- findet alle Formulare, die einem bestimmten Referenzobjekt zugeordnet sind

Parameter

- **DBOBJECT** oBezugsobjekt - Referenzobjekt, dessen Formulare gefunden werden
- **yit_FRM_Formular_Set** &FormSet - gefundene Formulare
- **STRING** &sFehlertext - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt

Rückgabewert

- **INT** iRC - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung

yit_FRM_Get_Stammdaten_String_

Bezeichnung

- findet den Wert des Elements des Referenzobjekts unter dem angegebenen Pfad

Parameter

- **DBOBJECT** oBezugsobjekt - das Objekt, dessen Feldwert gesucht wird
- **STRING** &sReturn- Feldwert
- **STRING** sMemberPath - der Pfad zum gesuchten Feld
- **STRING** &sFehlertext - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt

Rückgabewert

- **INT** iRC - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung
 - -1 - Der Membername des Bezugsobjekts existiert nicht; der angegebene Memberpfad existiert nicht

yit_FRM_Get_SubType_By_Brw

Bezeichnung

- öffnet einen Dialog, in dem der Benutzer einen Untertyp auswählen kann, der in der **RetVal**-Variable gespeichert wird

Parameter

- **STRING** sType - `yit_FRM_SubType_` des Element, mit dem nach angebotenen Untertypen gesucht wird (Zuordnung)
- **STRING** &sRetVal- ausgewählter Untertyp

Rückgabewert

- **INT** iRC - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung
 - -1 - kein Zuordnung für den angegebenen Untertyp gefunden

yit_FRM_Handle_Response_

Bezeichnung

- verarbeitet Daten nach dem Ausfüllen des Formulars; erstellt neue Elemente, die durch Klonen erstellt wurden; speichert Werte in Formularelementen und speichert in der Auswertung;

Parameter

- **yit_FRM_Formular_** oForm - das Formular, das der Benutzer gerade ausgefüllt hat
- **HJSON** hResponse - Daten aus dem ausgefüllten Formular im Frontend
- **HJSON** &hOutput - die Antwort, ob das Makro erfolgreich ausgeführt wurde oder nicht
- **STRING** &sFehlertext - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt

Rückgabewert

- **INT** iRC - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung
 - -1 - Formular nicht gefunden oder nicht hinterlegt in der Antwort

yit_FRM_Handle_Response_Coord_

Bezeichnung

- verarbeitet die Daten nach der Positionierung des Formulars

Parameter

- **yit_FRM_Formular_** oForm - das Formular, das der Benutzer gerade positioniert hat
- **HJSON** hResponse - Daten aus dem positionierten Formular im Frontend
- **HJSON** &hOutput - die Antwort, ob das Makro erfolgreich ausgeführt wurde oder nicht
- **STRING** &sFehlertext - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt

Rückgabewert

- **INT** iRC - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung
 - -1 - Formular nicht gefunden oder nicht hinterlegt in Antwort

yit_FRM_Handle_Response_PDF_

Bezeichnung

- verarbeitet Daten nach dem Ausfüllen des Formulars; speichert Werte in Formularelementen und speichert in der Auswertung;

Parameter

- **yit_FRM_Formular_**oForm - das Formular, das mit Benutzerantworten ausgefüllt wird
- **PDF** &oInput - PDF-Formularvorlage
- **HJSON** hResponse - Daten aus dem ausgefüllten Formular
- **HJSON** &hOutput - die Antwort, ob das Makro erfolgreich ausgeführt wurde oder nicht
- **STRING** &sFehlertext - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt

Rückgabewert

- **INT** iRC - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung
 - 2 - Fehler beim Kopieren von Elementen beim Erstellen der Auswertung
 - -1 - Formular nicht gefunden oder nicht hinterlegt in Antwort

yit_FRM_Init_Formular_Build_

Bezeichnung

- erstellt ein JSON-Objekt aus einem DOCUframe-Formularobjekt; kann mit HTML- und PDF-Formularen aufgerufen werden

Parameter

- **yit_FRM_Formular_**oForm - das Formular, aus dem das JSON-Objekt erstellt wird
- **HJSON** &hData - erstelltes JSON-Objekt
- **PDF** &oPDF - PDF-Formularvorlage; spielt nur in PDF-Formularen eine Rolle
- **STRING** &sFehlertext - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt
- **BOOL** bFromProcess - spielt nur bei der Arbeit mit einem HTML-Formular eine Rolle
 - wenn bFromProcess **TRUE** ist, wird das alte Formular überschrieben

- wenn es **FALSE** ist, öffnet sich ein Dialog mit der Frage des Überschreibens des Formulars
- Optionaler Parameter mit dem Standardwert **FALSE**

Rückgabewert

- **INT** iRC - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung
 - -1 - Fehler bei der Erstellung des Formulars

yit_FRM_Init_Formular_Positioning_

Bezeichnung

- erstellt ein JSON-Objekt aus einem DOCUframe-Formularobjekt; dieses JSON-Objekt wird an das Frontend gesendet, um die Elemente zu positionieren

Parameter

- **yit_FRM_Formular_**oForm - das Formular, aus dem das JSON-Objekt erstellt wird
- **HJSON** &hData - erstelltes JSON-Objekt
- **STRING** &sFehlertext - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt

Rückgabewert

- **INT** iRC - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung
 - -1 - Fehler bei der Erstellung des Formulars

yit_FRM_Init_Handle_Response_PDF_

Bezeichnung

- erstellt ein Formular aus einer PDF-Datei

Parameter

- **STRING** sOID - OID der PDF-Datei, aus der das Formular erstellt wird
- **STRING** &sFehlertext - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt

Rückgabewert

- **INT** iRC - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung
 - -1 - Fehler bei der Erstellung des Formulars

yit_FRM_Kategorie_Add_Subkategories

Bezeichnung

- fügt Unterkategorien von einer Kategorie zur anderen hinzu

Parameter

- `yit_FRM_Kategorie_&oSource` - die Kategorie, aus der die Unterkategorien kopiert werden
- `yit_FRM_Kategorie_&oDestination` - Kategorie, zu der Unterkategorien hinzugefügt werden
- `STRING &sFehlertext` - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt

Rückgabewert

- `INT iRC` - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung

yit_FRM_Kategorie_Add_To_Element_

Bezeichnung

- öffnet einen Dialog mit einer Liste aller Kategorien, fügt nach Auswahl die ausgewählte Kategorie zum geöffneten Element hinzu

Parameter

- `yit_FRM_FormElement_&oElement` - ein Element, dem eine Kategorie hinzugefügt wird
- `DBOBJECT oSource` - ein Formular, zu dessen Element wir eine Kategorie hinzufügen
 - bei Unterkategorien wird ein rekursiver Aufruf durchgeführt, wobei die Variable `oSource` ein Kategorieobjekt ist
- `STRING &sFehlertext` - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt

Rückgabewert

- `INT iRC` - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung
 - -1 - keine Kategorien am Formular definiert
 - 1 - der Benutzer hat den Vorgang abgebrochen

yit_FRM_Kategorie_Add_To_List

Bezeichnung

- erstellt eine neue Kategorie oder Kategorievorlage

Parameter

- **DBOBJECT** &oSource - ein Element, dem eine Kategorie hinzugefügt wird
- **STRING** &sFehlertext - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt
- **BOOL** bVorlage
 - wenn bVorlage **TRUE** ist, wird eine neue Kategorievorlage erstellt
 - wenn es **FALSE** ist, wird eine neue Kategorie erstellt

Rückgabewert

- **INT** iRC - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung
 - 1 - Fehler beim Erstellen einer Kategorie oder Kategorievorlage

yit_FRM_Link_Elements_

Bezeichnung

- fügt ein übergeordnetes Formularelement zu einem anderen Formularelement hinzu oder entfernt es daraus

Parameter

- **yit_FRM_FormElement_** &oElement - Element, zu dem das Elternelement hinzugefügt/entfernt wird
- **DBOBJECT** &oSource - ein Formular oder ein Formularelement-objekt
 - wenn es **Formular** ist, öffnet sich ein Dialog zur Auswahl des Elternelements
 - wenn es vom Typ **Element** ist, ist es ein Elternelement
- **STRING** &sFehlertext - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt
- **INT** iMode
 - wenn iMode **1** ist, wird dem Element ein Elternelement hinzugefügt
 - wenn es nicht **1** ist, wird das Elternelement aus dem Element entfernt
 - Optionaler Parameter mit dem Standardwert **1**

Rückgabewert

- **INT** iRC - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung
 - 1 - ungültiges Source-Object
 - 1 - der Benutzer hat den Vorgang abgebrochen

yit_FRM_Open_CSS_MDI

Bezeichnung

- öffnet den CSS-Dateidialog als MDI-Fenster

Parameter

- keine Parameter

Rückgabewert

- kein Rückgabewert

yit_FRM_Parse_Response_

Bezeichnung

- parst HTTP-Antwort

Parameter

- **STRING** sResponse - HTTP-Antwort, die geparkt wird
- **DBSTRINGSET** &ResponseContent- geparkte Antwort;
 - Set mit Elementen der Form "key": "value".
- **STRING** &sResult - HTTP-Statuscode
- **STRING** &sFehlertext - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt

Rückgabewert

- **INT** iRC - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung
 - -1 - konnte Response nicht parsen

yit_FRM_Reset_CSS_File_

Bezeichnung

- setzt das Feld **yit_FRM_Uploaded_** der CSS-Datei auf **FALSE**

Parameter

- **yit_FRM_CSS_File_** &oFile - CSS-Datei, deren Feld geändert wird

Rückgabewert

- **INT** iRC - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung

yit_FRM_SearchInsert_Download_Link_

Bezeichnung

- fügt einen Link vom Formular zum Formularelement hinzu; wenn der Link nicht existiert, kann der Benutzer die Option zum Erstellen auswählen

Parameter

- `yit_FRM_FormElement_ &oElement` - Element, zu dem der Link hinzugefügt wird
- `STRING &sFehlertext` - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt

Rückgabewert

- `INT iRC` - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung
 - -1 - Fehler beim Erstellen des Links

yit_FRM_Set_Basisdaten_Value_

Bezeichnung

- kopiert den Wert aus dem Formularelement in die zugehörigen Basisdaten

Parameter

- `yit_FRM_FormElement_ &oElement` - Formularelement, aus dem wir den Wert kopieren und in dessen Basisdaten wir den Wert speichern
- `STRING &sFehlertext` - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt

Rückgabewert

- `INT iRC` - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung
 - -1 - Bezugsobjekt ist gesperrt

yit_FRM_Set_FormElementValue_From_Basisdaten_

Bezeichnung

- kopiert den Wert aus den Basisdaten im Formularelement

Parameter

- `yit_FRM_FormElement_ &oElement` - Formularelement, in dem wir den Wert seiner Basisdaten speichern
- `STRING &sFehlertext` - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt

Rückgabewert

- **INT** iRC - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung
 - -1 - keine Basisdaten hinterlegt

yit_FRM_Set_Stammdaten_AddOn_

Bezeichnung

- öffnet einen Dialog mit möglichen Stammdaten und nach der Auswahl verbindet es die ausgewählten Stammdaten mit dem offenen Textelement des Formulars

Parameter

- **HDIALOG** Dialog - der Dialog, zu dem dieses Makro hinzugefügt wird
- **yit_FRM_FormElementText_** &Object - ein Textelement des Formulars, dem Stammdaten hinzugefügt werden

Rückgabewert

- **INT** iRC - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung

yit_FRM_Set_Stammdaten_String_

Bezeichnung

- gibt den Feldwert aus dem Formularelement in das Formularbezugsobjekt ein

Parameter

- **yit_FRM_Formular_** oForm - Formular, in dessen Bezugsobjekt der Wert eingetragen wird
- **yit_FRM_FormElement_** oElement - Element, dessen Wert eingegeben wird
- **STRING** &sFehlertext - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt

Rückgabewert

- **INT** iRC - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung
 - 1 - mehr als ein Platzhalter vorhanden

yit_FRM_Set_Standard_Positioning_AddOn_

Bezeichnung

- öffnet 4 Dialoge, in denen der Benutzer die x- und y-Koordinaten, Höhe und Breite für alle Formularelemente eingeben kann, und setzt dann die eingegebenen Werte für die Elemente

Parameter

- `yit_FRM_Formular_`oForm - das entsprechende Formular
- `yit_FRM_FormElement_Set` &SelSet - ein Set aller Formularelemente, denen Breite, Höhe, x- und y-Koordinaten zugewiesen sind
- `STRING` &sFehlertext - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt

Rückgabewert

- `INT` iRC - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung

yit_FRM_Show_Process_

Bezeichnung

- gibt den Namen des Prozesses zurück, der mit den angegebenen Basisdaten verknüpft ist

Parameter

- `yit_FRM_Basisdaten_`oBasisdaten - die Basisdaten, deren Prozess gesucht wird speichern

Rückgabewert

- `STRING` sRet - Prozessname

yit_FRM_Update_FormElement_

Bezeichnung

- kopiert das Formularelement und weist ihm einen bestimmten Wert zu

Parameter

- `yit_FRM_FormElement_` &oSource - das Formularelement, das kopiert wird
- `yit_FRM_FormElement_` &oTarget - das Formularelement, in das es kopiert wird
- `DBSTRINGSET` &ElementValues - der Wert des Formularelements
- `STRING` &sFehlertext - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt

Rückgabewert

- `INT` iRC - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung

yit_FRM_Upload_CSS_File_

Bezeichnung

- lädt eine CSS-Datei hoch, indem die Funktion aufgerufen wird, deren Informationen sich im Dateifeld `yit_FRM_Service_Target_` befinden

Parameter

- `yit_FRM_CSS_File_ &oFile` - CSS-Datei, die hochgeladen wird
- `STRING &sFehlertext` - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt

Rückgabewert

- `INT iRC` - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung

yit_FRM_Valid_Form_

Bezeichnung

- prüft, ob das Formular gültig ist

Parameter

- `yit_FRM_Formular_ &oFile` - das Formular, das geprüft wird
- `STRING &sFehlertext` - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt
- `STRING &sWarning` - String, in dem der Warntext gespeichert wird, wenn eine Warnung auftritt

Rückgabewert

- `INT iRC` - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - das Formular ist gültig
 - -1 - das Formular ist nicht gültig

yit_FRM_Write_Stammdaten_Back_

Bezeichnung

- setzt den Wert des gegebenen Feldes des Bezugsobjekts auf den gegebenen Wert

Parameter

- `DBOBJECT &oBezugsobjekt` - Objekt, auf das wir den Wert des Feldes ändern
- `STRING sMemberPath` - Memberpfad zum Objektfeld
- `STRING sValue` - der in das Feld einzugebende Wert

- **STRING** &sFehlertext - String, in dem der Fehlertext gespeichert wird, wenn ein Fehler auftritt

Rückgabewert

- **INT** iRC - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung
 - -1 - falscher Memberpfad

yit_FRM_Zeilenmakro_Elemente_AddOn_

Bezeichnung

- wenn das Formularelement vom Typ "Header" ist, wird es von Farbe, Hintergrund und Stilisierungstext begleitet

Parameter

- **yit_FRM_FormElement_** Object - ein Element des Formulars, dem der CSS-Stil zugewiesen ist
- **INT** &textColor - Textfarbe
- **INT** &BkColor - Hintergrundfarbe
- **BOOL** &Bold - ob der Titel fett ist
- **BOOL** &Italic - ob der Titel kursiv ist

Rückgabewert

- **INT** iRC - eine ganze Zahl, die angibt, ob das Makro erfolgreich ausgeführt wurde
 - 0 - erfolgreiche Ausführung

Glossar

Application Programming Interface (API)

Eine Anwendungsprogrammierschnittstelle(API), die Interaktionen zwischen mehreren Softwareanwendungen definiert. Sie definiert die Arten von Aufrufen oder Anfragen, die gemacht werden können, wie sie gemacht werden, die Datenformate, die verwendet werden sollten und die zu befolgenden Konventionen.

Cascading Style Sheets (CSS)

Eine Programmiersprache, die es Ihnen ermöglicht, das Design von elektronischen Dokumenten zu bestimmen.

Hypertext Markup Language (HTML)

Eine textbasierte Auszeichnungssprache zur Strukturierung elektronischer Dokumente wie Texte mit Hyperlinks, Bildern und anderen Inhalten. HTML-Dokumente sind die Grundlage des World Wide Web und werden von Webbrowsern dargestellt.

iframe Tag

Das <iframe> Tag gibt einen Inline-Frame an. Ein Inline-Frame wird verwendet, um ein anderes Dokument in das aktuelle HTML-Dokument einzubetten.

JavaScript Object Notation (JSON)

Ein offenes Standard-Dateiformat und Datenaustauschformat, das menschenlesbaren Text verwendet, um Datenobjekte zu speichern und zu übertragen, die aus Attribut-Wert-Paaren und Array-Datentypen(oder anderen serialisierbaren Werten) bestehen.

Parameter

Als Parameter bezeichnet man in der Informatik die Platzhalter für die Eingabewerte einer Methode oder Funktion. Sie stehen in Klammern hinter dem Methodenbezeichner. Mehrere Parameter werden meist durch Kommas voneinander getrennt. Bei einem Methodenaufruf werden die Parameter dann mit konkreten Werten belegt.

Schnittstellenmakro

Eine Funktion in DOCUframe, die von externen Anwendungen aufgerufen werden kann. Nimmt einen String als Input entgegen.