

Genereller Suchhilfe-Exit zur Anzeige von Icons

Die Verwendung von grafischen Elementen im SAP-System ist doch arg eingeschränkt. Icons sind im Grunde die einzige Möglichkeit, etwas Farbe und Abwechslung in eine ALV-Liste zu bekommen. Schön ist es, wenn der Anwender selber einstellen kann, welche Icons zu welchen Informationen angezeigt werden sollen. Eine Status bietet sich perfekt an, um diesen durch ein geeignetes Icon visuell darzustellen.

Für die Definition des zu verwendenden Icons sollte jedoch der Name des Icons verwendet werden (ICON_OKAY, ICON_MESSAGE_INFORMATION etc) anstelle der entsprechenden Icon-ID (@0V@, @19@ usw). Der Vorteil der Icon-ID ist zwar, dass dann das Icon direkt angezeigt wird, wenn das Feld nicht Eingabe bereit ist, allerdings ist die ID überhaupt nicht sprechend und die Pflege wird zum Glücksspiel.

Wenn allerdings der Name des Icons verwendet wird, dann sieht man eben nur den Namen, aber nicht das zugehörige Icon.

Beispiel

Um zu verdeutlichen, wie ich die Icons einsetze, habe ich eine kleine Statustabelle definiert:

STATUS	TEXT	ICON_NAME
E	ohoh...	ICON_LED_RED
W	Achtung	ICON_LED_YELLOW
I	Alles Ok	ICON_LED_GREEN

Der Text sollte natürlich besser in eine separat definierte Texttabelle ausgelagert werden, aber in diesem Fall soll es nur als Beispiel dienen.

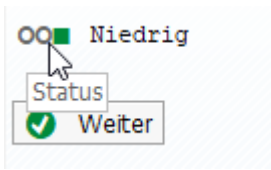
Ich habe also eine Tabelle mit verschiedenen Status. Irgendein Eintrag/ Objekt kann einen dieser Status haben. In der Listenausgabe wird zu einem Status der Name des zugeordneten Icons ermittelt und als Icon ausgegeben.

Darstellung eines Icons

Ein Icon besteht immer mindestens aus dem Icon selbst. Ein Icon muss immer Anfang eines Feldes stehen! Ich kann in jedem Anzeigefeld ein Icon ausgeben, indem ich die Icon-ID der Icons von zwei

Klammeraffen umgeben eintrage: @1A@.

Ein Icon kann aber auch noch einen Text besitzen, der mit angezeigt wird. Ebenso kann ein Icon eine 40 Zeichen lange Quickinfo enthalten, die bei MouseOver angezeigt wird.



Erzeugen eines Icons

Der richtige Weg, um Icons für die Ausgabe in einem Feld vorzubereiten ist die Verwendung des Funktionsbausteins ICON_CREATE:

```
CALL FUNCTION 'ICON_CREATE'
  EXPORTING
    name           = 'ICON_DELETE'
    text           = 'Löschen'
    info           = 'Markiertes Objekt löschen'
    add_stdinf     = ' '
  IMPORTING
    result         = lv_icon
  EXCEPTIONS
    OTHERS        = 3.
```

Dem Funktionsbaustein ist es egal, ob der Name oder die ID des Icons verwendet wird. Es kann also der Text „ICON_DELETE“ übergeben werden oder die Konstante ICON_DELETE. Die Icon-Konstanten sind im Type-Pool ICON definiert und in jedem ABAP-Programm vorhanden.

Tipp:

Erstelle eine globale Klasse mit einer Methode, die dir das fertige Icon direkt zurück gibt. Die Erzeugung von Icons ist dann sehr einfach:

```
DATA(lv_icon) = zcl_icon=>create( icon_okay ).
```

Suchhilfe

Zurück zum Thema. Ich wollte ja eigentlich was ganz anderes...

Eine Suchhilfe zum Status wäre sehr sinnvoll:

1. Der Anwender kann in einer ALV-Gridliste durch betätigen der F4-Hilfe auf dem Statusfeld alle möglichen Status sehen (Legende)
2. Bei der Auswahl eines Status ist eine Suchhilfe sowieso sinnvoll.

Leider wird in diesem Fall ja nur der Name des Icons angezeigt und nicht das Icon selber. Das möchte ich ändern. Allerdings möchte ich nicht für genau diese eine Tabelle einen Suchhilfe-Exit basteln, der genau für diese Tabelle das Statusicon anzeigt. Ich möchte eine Suchhilfe erstellen, die für alle ähnlichen Tabellen ebenfalls verwendet werden kann.

Suchhilfe-Parameter

Zu diesem Zweck verwende ich die wenig bekannte Möglichkeit, in der Suchhilfe einen *Parameter* zu definieren. ein Parameter ist kein Feld der zugrunde liegenden Tabelle, sondern ein separat definierter Eintrag. Ein Parameter wird nicht in der Suchhilfe ausgegeben sondern dient der Übergabe an den Funktionsbausteins eines Suchhilfe-Exits.

Suchhilfe-Exit Funktionsbaustein

Ein Suchhilfe-Exit ist ein Funktionsbaustein mit einer definierten Schnittstelle. Zur Erstellung eines Suchhilfe-Exits wird am besten der Funktionsbaustein F4IF_SHLP_EXIT_EXAMPLE kopiert, der auch in der Hilfe zum Dynprofeld „Suchhilfe-Exit“ erwähnt wird.

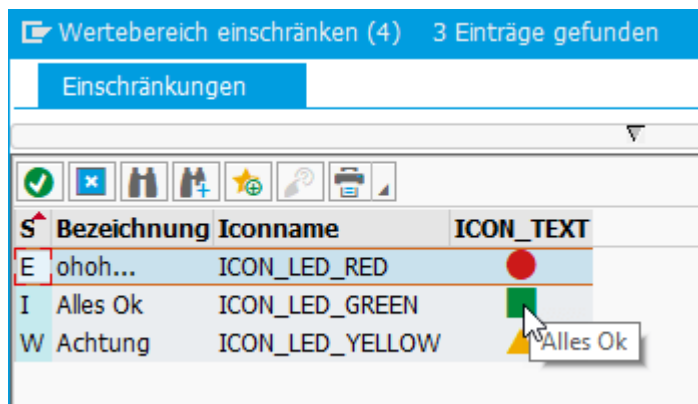
In diesem Funktionsbaustein kann die Suchhilfe zu verschiedenen Zeitpunkten manipuliert werden (Datenselektion, Darstellung etc).




Mit Hilfe des Funktionsbausteins F4UT_PARAMETER_VALUE_GET kann der Wert zu einem Parameter aus der Suchhilfe ermittelt werden.

Vorgehen

Ich möchte einen Suchhilfe-Exit erstellen, der aus der Suchhilfe ermittelt, in welchem Feld der Name des Icons steht und in welchem Feld das Icon selbst angezeigt werden soll.

Das Ergebnis sieht so aus:



	Bezeichnung	Iconname	ICON_TEXT
E	ohoh...	ICON_LED_RED	
I	Alles Ok	ICON_LED_GREEN	
W	Achtung	ICON_LED_YELLOW	

Lösung

Suchhilfe definieren

Als erstes muss ich in der Transaktion SE11 eine Suchhilfe anlegen:

Dictionary: Suchhilfe ändern

← → 🔍 🔄 📄 📁 🗑️ 📌 📎 📧 📧 📧 📧 📧 📧

Elementare Suchhilfe: aktiv

Kurzbeschreibung:

Eigenschaften **Definition**

Datenbeschaffung

Selektionsmethode:

Texttabelle:

Dialogverhalten

Dialogtyp:

Kurzanwahl:

Erweiterte Optionen

Vorschlagssuche auf Eingabefeldern

Spaltenübergreifende Volltextsuche (datenbankabhängig)

Genauigkeitswert für fehlertolerante Volltextsuche:

Suchhilfe-Exit:

Suchhilfe-Exit:

Parameter

Suchhilfeparameter	IMP	EXP	LPos	SPos	SAnz	Datenelement	M...	Defaultwert
STATUS	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1	1	<input type="checkbox"/>	STAT5	<input type="checkbox"/>	
TEXT	<input type="checkbox"/>	<input type="checkbox"/>	2	2	<input type="checkbox"/>	TEXT20		
ICON_NAME	<input type="checkbox"/>	<input type="checkbox"/>	3	3	<input type="checkbox"/>	ICON_NAME		
ICON_TEXT	<input type="checkbox"/>	<input type="checkbox"/>	4		<input type="checkbox"/>	ICON_TEXT		
COLUMN_WITH_ICON_NAME	<input type="checkbox"/>	<input type="checkbox"/>			<input type="checkbox"/>	CHAR30		
COLUMN_WITH_ICON_INFO	<input type="checkbox"/>	<input type="checkbox"/>			<input type="checkbox"/>	CHAR30		
COLUMN_FOR_ICON_DISPLAY	<input type="checkbox"/>	<input type="checkbox"/>			<input type="checkbox"/>	CHAR30		
	<input type="checkbox"/>	<input type="checkbox"/>			<input type="checkbox"/>			
	<input type="checkbox"/>	<input type="checkbox"/>			<input type="checkbox"/>			
	<input type="checkbox"/>	<input type="checkbox"/>			<input type="checkbox"/>			
	<input type="checkbox"/>	<input type="checkbox"/>			<input type="checkbox"/>			

Wertebereich einschränken (4) 3 Einträge gefunden

Einschränkungen

Bezeichnung	Iconname	ICON_TEXT
E ohoh...	ICON_LED_RED	●
I Alles Ok	ICON_LED_GREEN	■
W Achtung	ICON_LED_YELLOW	▲

Suchhilfe-Exit anlegen

Den Suchhilfe-Exit kopiere ich vom Funktionsbaustein F4IF_SHLP_EXIT_EXAMPLE.

Das Coding sieht folgendermaßen aus:

```
FUNCTION z_f4if_shlp_exit_icon.
```

```
*"-----"
```

```
*"*"Lokale Schnittstelle:
```

```
*" TABLES
```

```
*"     SHLP_TAB TYPE SHLP_DESCT
```

```
*"     RECORD_TAB STRUCTURE SEHLPRES
```

```
*" CHANGING
```

```
*"     VALUE(SHLP) TYPE SHLP_DESCR
```

```
*"     VALUE(CALLCONTROL) LIKE DDSHF4CTRL STRUCTURE DDSHF4CTRL
```

*"-----

```
DEFINE get_param.  
  CALL FUNCTION 'F4UT_PARAMETER_VALUE_GET'  
    EXPORTING  
      parameter      = &1  
      fieldname     = '*'  
    IMPORTING  
      value          = lv_parameter_value  
    TABLES  
      shlp_tab      = shlp_tab  
      record_tab    = record_tab  
    CHANGING  
      shlp          = shlp  
      callcontrol   = callcontrol  
    EXCEPTIONS  
      parameter_unknown = 1.  
IF sy-subrc = 0.  
  &2 = lv_parameter_value.  
ENDIF.  
END-OF-DEFINITION.
```

```
DEFINE icon_create.  
  CALL FUNCTION 'ICON_CREATE'  
    EXPORTING  
      name          = &1  
      info          = &2  
      add_stdinf    = ''  
    IMPORTING  
      result        = &3  
    EXCEPTIONS  
      OTHERS        = 3.  
  
END-OF-DEFINITION.
```

- *== Info:
- * Dieser Baustein kann in Suchhilfen als Suchhilfe-Exit verwendet werden
- *== Zweck:
- * Anzeige des Icons zu einem Icon-Namen, der in der Tabelle vorhanden ist
- *== Voraussetzung:
- * Spalte mit dem Namen eines Icons
- *

```
"Tabelle für den Iconnamen, der aus der RESULTTAB ermittelt wird  
DATA lt_col_iconname TYPE STANDARD TABLE OF icon_name.  
"Tabelle für den Inhalt der Info zum Icon  
DATA lt_col_iconinfo TYPE STANDARD TABLE OF icon_text.  
"Tabelle für das Aufbereitete Icon, das an die RESULTTAB übergeben wird  
DATA lt_col_icontext TYPE STANDARD TABLE OF icon_text.  
DATA lv_icontext TYPE icon_text.
```

```

"Generischer Parameter
DATA lv_parameter_value          TYPE  ddshvalue.
"Name der Spalte, die den Iconnamen enthält
DATA lv_column_with_icon_name TYPE  shlpfield.
"Name der Spalte, die den Iconcontext enthält
DATA lv_column_with_icon_info TYPE  shlpfield.
"Name der Spalte, in der das aufbereitete Icon dargestellt werden soll
DATA lv_column_for_display      TYPE  shlpfield.

```

```

* EXIT immediately, if you do not want to handle this step
IF callcontrol-step <> 'SENONE' AND
  callcontrol-step <> 'SELECT' AND
  callcontrol-step <> 'SENONE' AND
  callcontrol-step <> 'PRESEL' AND
  callcontrol-step <> 'SELECT' AND
  callcontrol-step <> 'DISP'.
  EXIT.
ENDIF.

```

```

*"-----
* STEP DISP      (Display values)
*"-----
IF callcontrol-step = 'DISP'.

```

```

  get_param 'COLUMN_WITH_ICON_NAME'      lv_column_with_icon_name.
  get_param 'COLUMN_WITH_ICON_INFO'      lv_column_with_icon_info.
  get_param 'COLUMN_FOR_ICON_DISPLAY'    lv_column_for_display.

```

```

"Ermitteln aller ICON_NAMEN der einzelnen Einträge

```

```

CALL FUNCTION 'F4UT_PARAMETER_VALUE_GET'

```

```

  EXPORTING

```

```

    parameter      = lv_column_with_icon_name
    fieldname      = '*'

```

```

  TABLES

```

```

    shlp_tab       = shlp_tab
    record_tab     = record_tab
    results_tab    = lt_col_iconname

```

```

  CHANGING

```

```

    shlp           = shlp
    callcontrol    = callcontrol

```

```

  EXCEPTIONS

```

```

    parameter_unknown = 1
    OTHERS             = 2.

```

```

IF sy-subrc = 0.

```

```

  "Wenn alles geklappt hat, dann sind in Tabelle LT_COL_ICONNAME die Name

```

```

n

```

```

  "der Icons aus dem Parameter COLUMN_WITH_ICON_NAME

```

```

  "Nun noch die die Texte für die Quickinfo ermitteln

```

```

CALL FUNCTION 'F4UT_PARAMETER_VALUE_GET'
  EXPORTING
    parameter          = lv_column_with_icon_info
    fieldname          = '*'
  TABLES
    shlp_tab           = shlp_tab
    record_tab         = record_tab
    results_tab        = lt_col_iconinfo
  CHANGING
    shlp               = shlp
    callcontrol        = callcontrol
  EXCEPTIONS
    parameter_unknown = 1
    OTHERS             = 2.
if sy-subrc > 0.
  "Macht nichts: Es ist kein Feld definiert/ vorhanden, das zum Icon an
gezeigt werden soll
  "Die Tabelle LT_COL_ICONINFO ist dann halt leer
endif.

LOOP AT lt_col_iconname INTO DATA(lv_iconname).
  "Info zum Icon lesen:
  READ TABLE lt_col_iconinfo INTO DATA(lv_iconinfo) INDEX sy-tabix.
  "Aufbereitung des Icons zur Darstellung
  icon_create lv_iconname lv_iconinfo lv_iconcontext.
  APPEND lv_iconcontext TO lt_col_iconcontext.
ENDLOOP.
ENDIF.

"Alle aufbereiteten Icons an RESULTTAB übergeben
CALL FUNCTION 'F4UT_PARAMETER_RESULTS_PUT'
  EXPORTING
    parameter          = lv_column_for_display
  TABLES
    shlp_tab           = shlp_tab
    record_tab         = record_tab
    source_tab         = lt_col_iconcontext
  CHANGING
    shlp               = shlp
    callcontrol        = callcontrol
  EXCEPTIONS
    parameter_unknown = 1
    OTHERS             = 2.
IF sy-subrc <> 0.
  EXIT.
ENDIF.
EXIT.
ENDIF.

ENDFUNCTION.

```

