

# Dynamisches Anzeigen von Informationen mit jQueryUI Dialogs und Tabs

Dr. Gudrun Pabst

BASEL BERN LAUSANNE ZÜRICH DÜSSELDORF FRANKFURT A.M. FREIBURG I.BR. HAMBURG MÜNCHEN STUTTGART WIEN



2012 © Trivadis

Dynamisches Anzeigen von Informationen mit jQuery UI Dialogs und Tabs

**trivadis**  
makes IT easier. ■ ■ ■

# AGENDA

- Bitte warten
- Auswahlfenster
- Tabs



# Bitte warten

- Langdauernde Verarbeitung:
  - Anzeige soll informieren, dass Verarbeitung noch läuft
  - Benutzer soll keine Buttons oder Links betätigen
- Lösung:
  - Modaler Dialog
  - Einblenden beim Submit



# Bitte warten

- Page 0
- Region „Bitte warten“

- Code:

```
<div style="height: 200px; width: 600px;
        display: table-cell; vertical-align: middle;"
    align="center">
    <h2>Verarbeitung läuft ...</h2>
    
</div>
```

- Static ID: BITTE\_WARTEN



# Bitte warten

## ■ Dynamic Action „initWartenDialog“

- Initialisierung der Region als Dialog
- Page Load
- Code:

```
$("#BITTE_WARTEN").dialog ({  
    autoOpen: false,  
    title:     "Bitte warten",  
    height:   240,  
    width:    650,  
    modal:    true,  
    open:     function(event, ui) {  
                $(".ui-dialog-titlebar-close").hide(); }  
    })
```



# Bitte warten

- Dynamic Action „AnzeigeWartenDialog“
  - Anzeige der Region beim Submit
  - Anzeige bleibt bis zum Aufbau der neuen Seite bestehen
  - Before Page Submit
  - Fire on Page Load : No
  - Code:

```
$( "#BITTE_WARTEN" ).dialog ( "open" );
```



# AGENDA

- Bitte warten
- Auswahlfenster
- Tabs



# Auswahlfenster

- Eingebaute Auswahlfenster:
  - nur je eine Spalte zur Anzeige und zur Rückgabe
- Eigene Auswahlfenster:
  - Auswahlliste basiert auf einer Report-Region
  - mit Suchfeldern – Suche ohne Submit
  - Anzeige der Report-Region als Dialog
  - Rückgabe von mehreren Spalten mittels JavaScript





# Auswahlfenster

- Report Region:
  - Where-Bedingung mit Suchfeldern
  - Page Items to Submit: alle Suchfelder
  - Static ID zur einfacheren Verwaltung mittels Javascript
  - zunächst ausgeblendet

**Source**

Region Source

```
select mit_id ,
       mit_nachname ,
       mit_vorname ,
       mit_email
from dsp_mitarbeiter
where mit_nachname like nvl(:P20_MIT_NACHNAME_SUCHE, '%')
```

Page Items to Submit: P20\_MIT\_NACHNAME\_SUCHE

Use Query-Specific Column Names and Validate Query  
 Use Generic Column Names (parse query at runtime only)

Maximum number of generic report columns:  
60

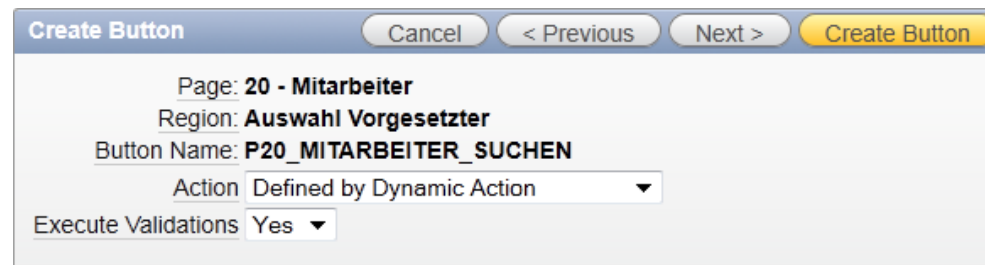
**Attributes**

Static ID: REP\_MIT\_SUCHE

Region Attributes: style="display:none;"

# Auswahlfenster – Suche

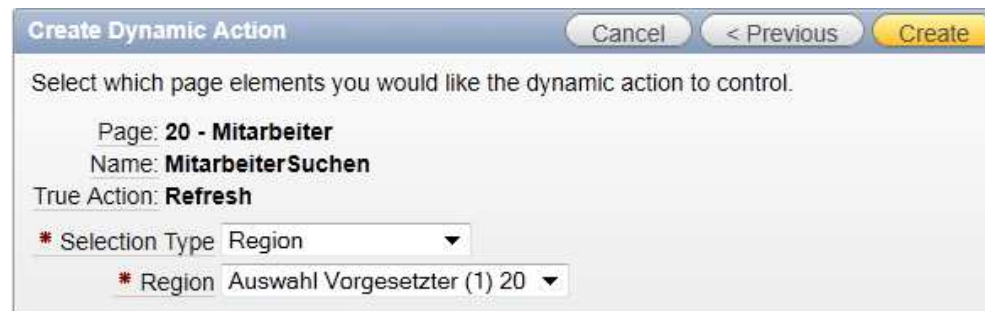
- Button zur Suche:
  - Aktion definiert über Dynamic Action



The screenshot shows the 'Create Button' dialog box. It has a title bar with 'Create Button' and buttons for 'Cancel', '< Previous', 'Next >', and 'Create Button'. The main content area displays the following information:

- Page: **20 - Mitarbeiter**
- Region: **Auswahl Vorgesetzter**
- Button Name: **P20\_MITARBEITER\_SUCHEN**
- Action: **Defined by Dynamic Action** (dropdown menu)
- Execute Validations: **Yes** (dropdown menu)

- Dynamic Action „SucheAusführen“:
  - Refresh der Reports-Region

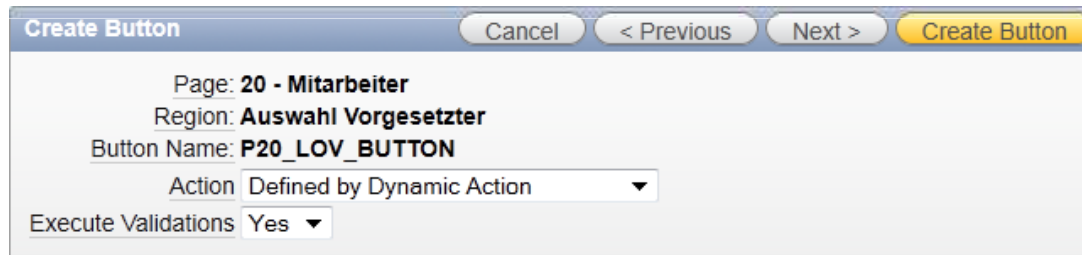


The screenshot shows the 'Create Dynamic Action' dialog box. It has a title bar with 'Create Dynamic Action' and buttons for 'Cancel', '< Previous', and 'Create'. The main content area displays the following information:

- Select which page elements you would like the dynamic action to control.
- Page: **20 - Mitarbeiter**
- Name: **MitarbeiterSuchen**
- True Action: **Refresh**
- \* Selection Type: **Region** (dropdown menu)
- \* Region: **Auswahl Vorgesetzter (1) 20** (dropdown menu)

# Auswahlfenster – Report als Dialog

- Dynamic Action „initSuchDialog“:
  - wie bei „Bitte warten“
- Button und Dynamic Action „AnzeigeSuchDialog“:
  - Button „among this region's items“
  - Aktion definiert über Dynamic Action



Create Button

Cancel < Previous Next > Create Button

Page: **20 - Mitarbeiter**  
Region: **Auswahl Vorgesetzter**  
Button Name: **P20\_LOV\_BUTTON**  
Action: Defined by Dynamic Action  
Execute Validations: Yes

- Dynamic Action – Code:

```
$ (" #REP_MITARBEITER_SUCHE" ).dialog ( "open" );
```

# Auswahlfenster – Rückgabe der Daten

## ■ JavaScript-Funktion „WerteSetzen“:

- Code:

```
function WerteSetzen ( pID , pNachname , pVorname ) {  
    $s ("P20_MIT_MIT_ID", pID);  
    $ ("#P20_VORGESETZTER").text (pNachname+" , "+pVorname);  
    $ ("#REP_MITARBEITER_SUCHE").dialog ("close");  
}
```

- in den Page-Eigenschaften oder in einer JavaScript-Library
- Achtung: Für Hidden-Items muss „Value Protected“ auf „No“ gesetzt werden!

## ■ Link zur Rückgabe der Daten:

- ID im Report als Link auf eine URL
- URL:

```
javascript:WerteSetzen (' #MIT_ID# ', ' #MIT_NACHNAME# ',  
    ' #MIT_VORNAME# ' );
```

# AGENDA

- Bitte warten
- Auswahlfenster
- Tabs

# Tabs

- Tabelle mit mehreren abhängigen Tabellen:
  - Anzeige der Master-Daten und aller Detail-Daten
  - großer Platzbedarf
- Tabs:
  - Anzeige der Master-Daten
  - Anzeige von nur immer einer abhängigen Tabelle mittels Karteikarten

Projekt	Funktion	Prozent	Start	Ende
PR0156-APEX-Entwicklung	Projektleiter	50	15.03.2012	31.12.2012
PR0156-APEX-Entwicklung	Anwendungsdesigner	50	15.03.2012	31.12.2012
PR0157-APEX-Bestellverwaltung	Projektleiter	10	01.01.2012	25.06.2012
PR0157-APEX-Bestellverwaltung	Anwendungsdesigner	40	01.01.2012	25.06.2012
PR0157-APEX-Bestellverwaltung	Programmierer	50	01.01.2012	25.06.2012
PR0160-Konzept APEX-Anwendung	Datenmodellierer	75	17.04.2012	22.09.2012
PR0160-Konzept APEX-Anwendung	Anwendungsdesigner	25	17.04.2012	22.09.2012

1 - 7

# Tabs

## jQuery:

- HTML-Code:

```
<div id="jQueryTabs">
  <ul>
    <li><a href="#tab-1">Erstes Tab</a></li>
    <li><a href="#tab-2">Zweites Tab</a></li>
  </ul>
  <div id="tab-1">
    Code erstes Tab
  </div>
  <div id="tab-2">
    Code zweites Tab
  </div>
</div>
```

- Javascript-Code zum Initialisieren:

```
$( "#jQueryTabs" ).tabs();
```

# Tabs

## APEX:

### ■ Templates

- Einbinden der benötigten js – und css – Dateien in das Page-Template
- Neues Region-Template „jQuery Tab Region“ für Region mit Subregions

### ■ Page

- Anlegen einer HTML-Region „Tabs“ mit dem neuen Region-Template
- Einrichten der zukünftigen Tab-Regions als Subregions der Region „Tabs“

- Beim Page Load:

Initialisieren der Region „Tabs“ als jQuery-Tab



# Tabs - Templates

- Erweitern des Page-Template um:

```
<link rel="stylesheet" href="/i/libraries/jquery-ui/1.8.14/themes/base/jquery.ui.tabs.css" type="text/css" />
```

```
<script src="/i/libraries/jquery-ui/1.8.14/ui/jquery.ui.tabs.js" type="text/javascript"></script>
```

- Region-Template „jQuery Tab Region“

- Einträge:

The screenshot shows a window titled "Definition" with two sections: "Template" and "Sub Regions".

**Template**

```
<div id="jQueryTabs">
  #SUB_REGION_HEADERS#
  #BODY#
</div>
```

**Sub Regions**

**Header Template**

```
<ul>#ENTRIES#</ul>
```

**Header Entry Template**

```
<li><a href="##SUB_REGION_ID#">#SUB_REGION_TITLE#</a></li>
```

**Template**

```
#SUB_REGION#
```

# Tabs - Page

- Neue HTML-Region „Tabs“

User Interface

Template jQuery Tab Region \* Sequence 20

Parent Region - Select a Parent -

Display Point Page Template Body (3. items above region content) Column 1

[Body] [Pos.1] [Pos.2] [Pos.3] [Pos.4]

- Tab-Regions als Subregions

User Interface

Template Reports Region \* Sequence 30

Parent Region Tabs (1) 20

- Dynamic Action beim Page Load

Settings

\* Code `$("#jQueryTabs").tabs();`

VIELEN DANK

BASEL    BERN    LAUSANNE    ZÜRICH    DÜSSELDORF    FRANKFURT A.M.    FREIBURG I.BR.    HAMBURG    MÜNCHEN    STUTTGART    WIEN