# Wizdom Document/LIST CAML Query

# Table of Contents

# 1. Introduction

The Wizdom WSS CAML Connector is a template controlled Web Part used to display data originating from Sharepoint Lists and document archives.

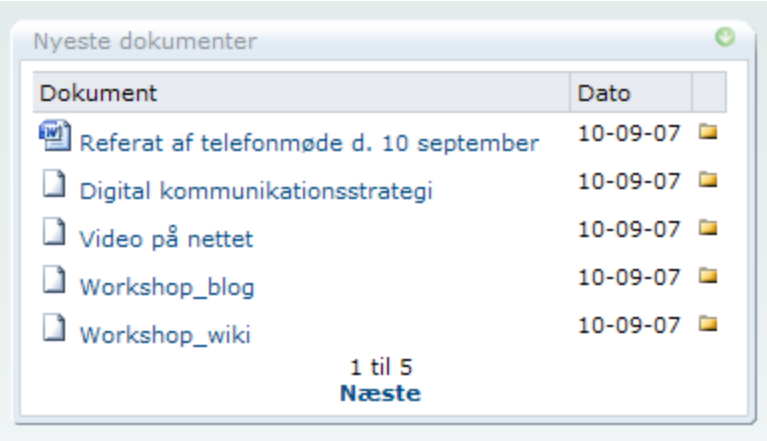
The Web Part is in many ways very similar to the Wizdom SQL-Connector with the main difference being that data originates from Sharepoint Lists instead of SQL Databases. List data are accessed using the Sharepoint object model and this means that only data residing in Sharepoint Application hosted on the same machine is accessible.

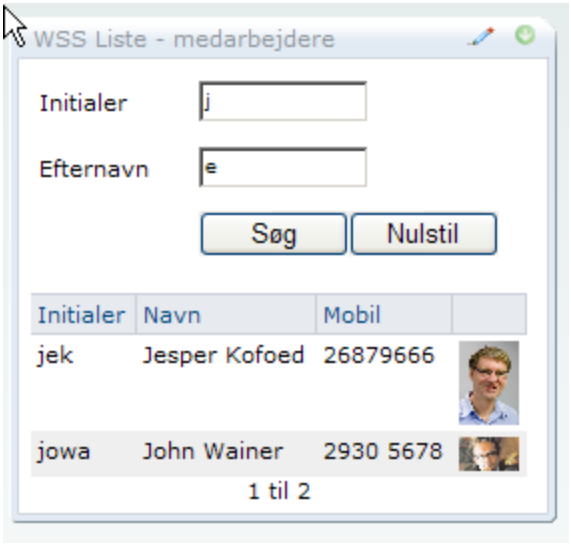
The Web Part is relative complex to configure and is designed to be configured by developers.

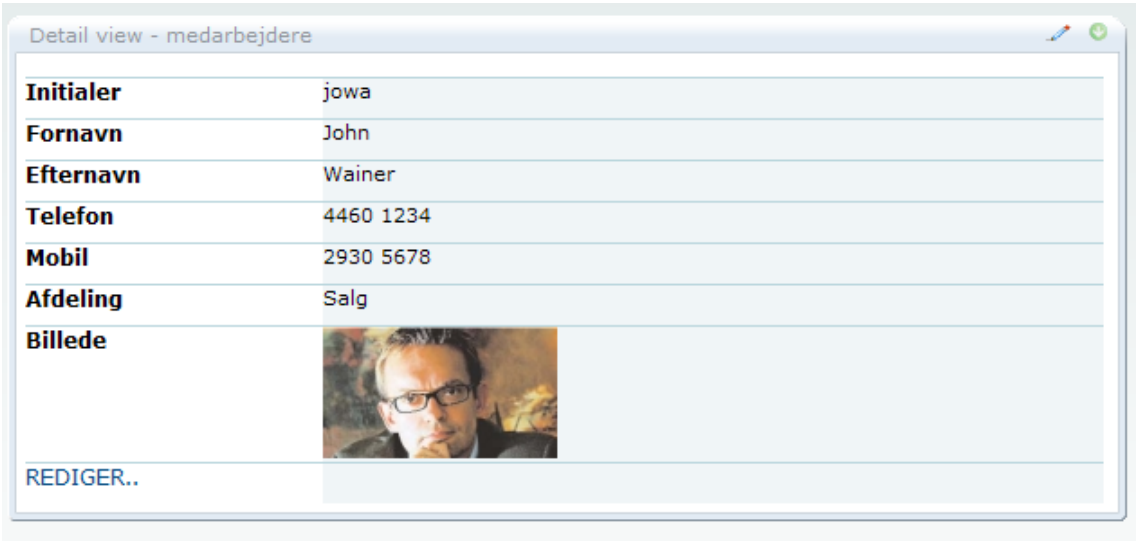
Highlights of the Web Part are:

- Due to the use of properties that outlines HTML templates, the Web Part makes it possible to specify in a detailed way how the component should render the resulting HTML.
- The Web Part uses AJAX technology and thus only the necessary part of the page containing the Web Part will be refreshed when searching, sorting or paging is activated.
- Built-in-script makes it possible to initiate a query by pressing enter instead of using the search link or button.
- It is possible to control the CAML Connector Web Parts from other Web Parts via JavaScript, for instance two CAML Connector Web Parts may be connected in a master/detail view.
- Using the property "Cross site Query", it is possible to specify whether the Sharepoint API `SPSiteDataQuery` or `SPQuery` is used. The advantage of SPSiteDataQuery is that the search scope may be specified to be Site Collection level, Site Level or one or more lists including sub folders. The scope of SPQuery is a single list (and no sub folders), but the query is more efficient.
- Both Document Libraries and Custom Lists are supported as data source.

Four examples of how the CAML-connector may appear, are shown below:



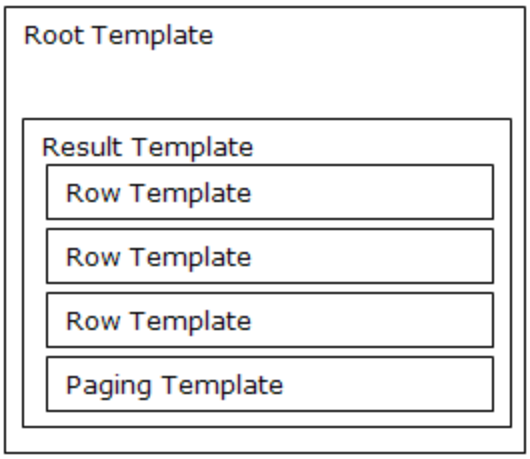

Note: Knowledge about HTML as well as Sharepoint CAML Query's is necessary in order to configure this Web Part.

# 2. HTML Template Properties

Four of the SQL-Connectors 16 properties are used to specify HTML-Templates:

- The Root Template
- The Result Template
- The Row Template
- The Paging Template

The hierarchy of which is shown below:




### THE ROOT TEMPLATE

The root template specifies the HTML that renders the – optional - Query input fields as well as the submit link/button and the reset link/button. The template must also contain the __RESULTSET__ placeholder, which indicates the position in the Root Template HTML where the query result(s) (controlled by the Result Template) should be placed.

The Root Template may also in some cases include links that enables the user to select how to sort the query results, but it is most common to place these links in the Result Template described below.

**Note:** The strange looking string "[$id]" will be replaced with a string unique for the Web Part and thus name collisions can be avoided. If, for instance, a JavaScript function is needed inside the Root Template then the "[$id]" should be included in the function name in order to avoid two or more functions with the same name on the same page.

**Placeholders and built-in script functions:**

- The id of the query input fields should be of the format: <FIELD-NAME>__[$id], e.g.:

```
<input type="text" id="LASTNAME__[$id]" VALUE="" size="14" />
```

- A query is activated via the built in JavaScript function "wt1CAMLSqlSubmit ('wt1formcamlsql[$id]')" e.g.:

```
<button class="button"
 onclick="wt1CAMLSqlSubmit('wt1formcamlsql[$id]');
  return false;">Search</button>
```

- Similarly a query is reset using the built in JavaScript function wt1CAMLSqlReset ('wt1formcamlsql[$id]', 'wt1result[$id]') e.g.:

```
<button class="button"
onclick="wt1CAMLSqlReset('wt1formcamlsql[$id]',
'wt1result[$id]'); return false;">Reset</button>
```

**Example**: The Root Template property for the "most recently changed documents" sample is shown below:

```
<table border="0" cellpadding="5" cellspacing="0" width="100%">
  <tr>
   <td width="70">
    Initialer
   </td>
   <td>
    <input type="text" id="Initialer__[$id]" VALUE="" size="14" />
   </td>
  </tr>
  <tr>
   <td>
    Efternavn
   </td>
   <td>
    <input type="text" id="Efternavn__[$id]" VALUE="" size="14" />
   </td>
  </tr>
```

```
  <tr>
   <td>
     
   </td>
   <td>
    <button class="button" onclick="wt1CAMLSqlSubmit('wt1formcamlsql[$id]'); return
false;">Søg</button>
    <button class="button" onclick="wt1CAMLSqlReset('wt1formcamlsql[$id]',
'wt1result[$id]'); return false;">Nulstil</button>
   </td>
  </tr>
</table>
<br />
__RESULTSET__
```

## THE RESULT TEMPLATE

The Result Template specifies the basic structure of the HTML that defines the lay-out of query results.

The template should contain the __LISTITEMS__ placeholder, which indicates where to place the individual rows (controlled by the Row Template) of the query results.

Optionally the template may contain the __PAGING__ placeholder, which inserts HTML controlled by the Paging Template.

**Placeholders and built-in script functions:**

In many layouts the Result Template will contain links that enables the user to select a sort column. Sorting is activated via the built-in javascript function wt1CAMLSqlorderby ('[$id]', '<column>', '<direction>'). <column> should be the name of a database field and direction should be "ASC" or "DESC" e.g.:

```
<a href="javascript:wt1sqlorderby('[$id]', 'Lastname', 'ASC');">Lastname</a>
```

The direction parameter determines the sort direction for the first click. More than one click will toggle between ASC and DESC

**Example**: The Result Template for the "most recently changed documents" sample is shown below:

```
<table border="0" cellpadding="3" cellspacing="0" align="center" width="99%">
   <tr class="listheadline" valign="top">
   <td class="ListHeader" align="left">
    <a href="javascript:wt1CAMLSqlorderby('[$id]', 'Title', 'ASC');">Initialer</a>
   </td>
   <td class="ListHeader" align="left">
    <a href="javascript:wt1CAMLSqlorderby('[$id]', 'Efternavn', 'ASC');">Navn</a>
   </td>
   <td class="ListHeader" align="left">
```

```
    <a href="javascript:wt1CAMLSqlorderby('[$id]', 'Mobil', 'ASC');">Mobil</a>
   </td>
   <td class="ListHeader" align="left"> </td>
   </tr>
  __LISTITEMS__
</table>
__PAGING__
```

## THE ROW TEMPLATE

The Row Template specifies the HTML that renders the individual rows in the query result.

### Placeholders

The row template supports a number of placeholders. Placeholders should begin with "__[" and end with "]__, e.g. "__[$LISTURL]__".

### Row data

Actual values from the active CAML query are inserted via placeholders of the form "__[<listfieldname>]__", e.g.: "__[Lastname]__".

Values of type "date" may additionally contain a format string. The standard .NET format strings are used. Example: __[OrderDate(dd/MM/yy)]__

### $OddEven

It is possible to render odd and even rows differently via the "OddEven" placeholder. Example:

```
<tr valign="top" class="__[$OddEven(CompTableRowEven1,CompTableRowOdd1)]__">"
```

And the resulting HTML on even rows may be:

```
<tr valign="top" class="CompTableRowEven1">"
```

### $DOCICON

Only valid for Document libraries. This placeholder is replaced with an HTML IMG tag corresponding to the current document.

**$DOCURL**

Only valid for Document libraries. The full URL to the document

**$FOLDERURL**

Only valid for Document libraries. The full URL to the folder containing the current document.

**$ROOTURL**

Only valid for Document libraries. The full URL to root of the Sharepoint Site Collection.

**$LISTURL**

The full URL to the List or document archive.

**$ITEMURL**

The full URL to the item or documents default property form.

**$OPENDOC_SCRIPT**

Only valid for Document libraries. The placeholder is replaced with client script used to open the current document. E.g.:

```
<a href="__[$DOCURL]__" target="_blank"
   onclick="__[$OPENDOC_SCRIPT]__">__[$DOCICON]__ __[$DOCNAME]__</a>
```

It is recommended to include the onclick/$OPENDOC_SCRIPT construct shown above.

**$DOCNAME**

Only valid for Document libraries. The current documents name without extension.

**$DOCNAME_WITH_EXT**

Only valid for Document libraries. The current documents name with extension.

**Built-in script functions**

The built client script functions "wt1CAMLSqlUpdate" may be used to connect to another CAML Web Parts on the same page. See the section "Master Detail Connection of CAML Connectors"

**Example**: The Row Template for the "most recently changed documents" sample is shown below:

```
<tr valign="top" class="__[$OddEven(CompTableRowEven1,CompTableRowOdd1)]__">
  <td align="left">
    <a href="__[$DOCURL]__" target="__blank"
      onclick="__[$OPENDOC_SCRIPT]__">__[$DOCICON]__ __[$DOCNAME]__</a>
  </td>
   <td>
    __[Modified(d-MM-yy)]__
  </td>
  <td>
    <a href="__[$FOLDERURL]__">
      <img height="10" src="__[$ROOTURL]__/_layouts/images/folder.gif"
                                                  border="none" /></a>
   </td>
</tr>
```

## THE PAGING TEMPLATE

The Paging Template specifies the HTML that renders the "Row X of Y [NextPage] [PrevPage] HTML.

**Placeholders:**

__STARTROW__ displays the number of the first row currently displayed; similarly __ENDROW__ displays the number of the last row.

The placeholders __PREV(<link HTML>)__ and __NEXT(<link HTML>)__ is used to specify the location and layout of the "next page" and "previous page" links.

Finally the "__NOROWSFOUND(<Message HTML>)__" placeholder is available. If the query doesn't return any rows then the rest of the Paging Template is ignored and only this placeholder is used.

**Example**: The Paging Template for the "Most recently changed documents" sample is shown below:

```
<center>
__STARTROW__ til __ENDROW__ <br />__PREV(<b>Forrige</b>)__ __NEXT(<b>Næste</b>)__
__NOROWSFOUND(Ingen dokumenter fundet.)__
</center>
```

# 3. Sharepoint CAML related Properties

## SITE COLLECTION URL

This property specifies the full URL to the Sharepoint Site Collection containing the list. Examples:

```
http://Intranet
[default]
.
```

The second example means that the URL should be the one specified in "Wizdom Common Configuration Setting-> WSS Document Libraries". The third example ".", is only valid when the Web Part is hosted by Sharepoint and means the current Site Collection.

## RELATIVE SITE URL

This property specifies the relative URL to the Sharepoint Site containing the list. If the list is placed in the root site of the Site Collection the property should be left empty. Exampel:

```
Documents
```

"[default]" and "." may be used in a similar way as in the Site Collection property.

## LIST/LIB NAME

This property specifies the name of the list or document archive. Exampel:

```
Library1
```

"[default]" may be used in a similar way as in the Site Collection property.

## DOCUMENT LIBRARY

This checkbox should be checked if the specified list is a document library.

## CROSS SITE QUERY

Using this property it is possible to specify whether the Sharepoint API `SPSiteDataQuery` or `SPQuery` is used. The advantage of SPSiteDataQuery is that the search scope may be specified to be at Site Collection level, Site Level or one or more lists including sub folders. The scope of SPQuery is a single list (and no sub folders), but the query may be more efficient.

## CAML PARAMETER ' WEBS'

This (complex) property is only used if "Cross Site Query" is checked. The parameter specifies the Scope for the query and the format is the same as is used for the "Webs" property of the Sharepoint API class SPSiteDateQuery. Typical Values are:

```
<Webs Scope='Recursive' />
```
Search all lists and document archives contained in the specified WSS Site.

```
<Webs Scope='SiteCollection' />
```
Search all lists and document archives contained in the specified WSS Site.

```
<Lists><List ID="129AB4CAE-12EF-9871-DE45-F34A180D3EAB5"/></Lists>
```
Search the specified list. It is possible to specify more than one list. The List ID is accessible as part of the URL "WSS > Site Settings > Site Libraries and Lists > Customize [List name]"

## CAML QUERY STATEMENT OR CONDITIONS

This property specifies the actual CAML query or alternatively a set of conditions that will be transformed to the CAML query.

The full syntax for CAML queries is defined in the Windows Sharepoint Server documentation. Tree examples:

1:

```
<OrderBy>
  <FieldRef Name='Modified' Ascending='FALSE' />
</OrderBy>
```

2:

```
<Where>
  <Eq>
    <FieldRef Name='Quality' />
    <Value Type='String'>medium</Value>
  </Eq>
</Where>
<OrderBy>
  <FieldRef Name='Modified' Ascending='FALSE' />
</OrderBy>
```

3:

```
<Where>
   <And>
      <BeginsWith>
         <FieldRef Name="Efternavn" />
         <Value Type="Text">K</Value>
      </BeginsWith>
      <Gt>
         <FieldRef Name="Modified" />
         <Value Type="DateTime">2007-08-01T00:00:00Z</Value>
      </Gt>
   </And>
</Where>
<OrderBy>
  <FieldRef Name='Modified' Ascending='FALSE' />
</OrderBy>
```

The free tool "U2UCamlCreator" may be used to create CAML XML.

**Dynamic CAML creation**

It is possible to create the CAML dynamically based on HTML input fields, URL parameters etc. This mode is selected if the first character in the Property is set to "#". The property should consist of one or more lines of the form:

<List Column> <CAML Operator> <Dynamic Value>

The resulting query will "and" all conditions where the current "Dynamic Value" is not empty.

The tree parts of a condition are described below.

**<List Column>**

The internal name for the list column in Sharepoint.

**<CAML Operator>**

One of: Eq, Neq, Gt, Lt, Leq, IsNull, IsNotNull, BeginsWith, Contains or DataRangesOverlap.

**<Dynamic Value>**

One of:

FIELD:<FIELDNAME>. FIELDNAME should identify the first part of the ID of an HTML Input tag defined in the Root template property. If the end-user types a value into the specified field, this value will be incorporated into the resulting CAML XML query.

URLPARM:<PARM NAME>. Similar to FIELD, but the value is taken from the querystring.

ITEMCOLLECTION:<ITEM NAME>. Similar to FIELD, but the value is taken from the ASP.NET Context Item Collection. This means that another Web Part on the page may calculate a value that will be incorporated into the query.

LOGONUSER. The logged on users logon id excluding the domain name

LOGONUSERFULL. The logged on users logon id including the domain name

**An Example:**

```
#CREATE FROM CONDITIONS
Editor Contains FIELD:EDITOR
FileRef Contains FIELD:FILENAME
FileRef Contains URLPARM:FILENAME
Modified_x0020_By Eq LOGONUSERFULL
ORDERBY AUTO
```

The resulting CAML XML may be inspected if the property "Show resulting CAML Query" is checked:


## CAML ViewFields

This property corresponds to the Sharepoint SPQuery API "ViewFields" property and should list the List columns used. The "ID" column needs not to be listed and the "FileRef" and "DocIcon" columns are automatically included if the "Document Library" is property is checked. Example:

```
<FieldRef Type="DateTime" Name='Modified' />
<FieldRef Name='Title' />
<FieldRef Name='Body' />
```

Normally the "Type" attribute seems not to be required.

**Hint**: the column names may be obtained from the HTML source generated by the standard WSS List display form "DispForm.aspx" that contains HTML comments showing the "FieldInternalName":

```
..
<!-- FieldName="Efternavn"
         FieldInternalName="Efternavn"
         FieldType="SPFieldText"
-->
..
```

## Show resulting CAML Query

This property makes it possible to view the resulting CAML XML and available columns. This may be helpful during the development phase.

## OTHER PROPERTIES

The remainder of the CAML Connectors properties is described in this section

### Rows per Page

As may be expected this property specifies the maximum number of result rows displayed at a time.

### Show results immediately

If this checkbox is checked then query results are displayed immediately when a page containing the Web Part is rendered. If not, only input fields and submitted links (as defined in the Root Template) are shown initially.

### Unique Client ID

Normally a client ID is assigned automatically, but if the CAML Connector should be controlled from other Web Parts on the page, it is necessary to assign an explicit ID via this property. Two CAML Connectors may for instance be connected in a master detail relationship as described later in this document.

### Cache time in seconds

If the CAML Connector is configured to operate in "static mode", as described later in this document, then the output from the CAML Connector may be cached. Caching is enabled if this property contains a number larger than 0.

Example: A cache time of just 5 seconds may result a very substantial increase in performance if the CAML Connector is used to display news headlines on the start page of a busy Web Site.

### Display nothing if resulting CAML does not contain this string

If a certain parameter (variable specification) in the dynamically created CAML statement is required, then this property may be used to disable output if the parameter is undefined.

Example:

A CAML Connector used to display details of employees may use the following condition:

```
Initials Eq URLPARM:EMP_INITIALS
```

If the property is set to "Initials" then the result of the query is only displayed if the URL parameter "EMP_INITIALS" is defined because the CAML XML will only contain

# 4. Static Mode

It is possible to configure the CAML Connector to execute in *Static Mode*. This mode should be used if no user interaction is needed, e.g. the query is executed when the page is loaded, but there is no need for query input fields, user selected sorting or paging. A typical scenario for the use of static mode is the display of the latest few news headlines on a website.

Static mode is selected if the Root Template property is empty. Additionally the properties *Paging Template, Initial Sort Column, Show results immediately,* are irrelevant in static mode.

The property *Cache time in seconds* - on the other hand - is only used in static mode.

# 5. Master Detail Connection of CAML Connectors

As mentioned in earlier sections it is possible to control the CAML Connector from other web parts on the same page. Often this facility is used to create a CAML Connector that display detailed data for a record selected by the user. E.g. the user click on a row in the result from the master web part and details corresponding to this row are displayed in the detail web part.

The built-in JavaScript function *wt1CAMLSqlUpdate* is used to activate the detail web part. The functions three parameters are: The detail web parts unique client id as specified via the *Unique Client ID* property, the name of a (normally hidden) input field defined in the detail view web parts "Root Template property" and the value to be assigned to this field before the web part is updated. An example taken from an "employee list sample" Row Template is shown below.

An example:

```
wt1CAMLSqlUpdate('EmpDetails','WSSEMPID', '__[Title]__')">
```

The CAML Connector's use of AJAX technology ensures a nice user experience in this setup.

The interesting properties for the sample detail web part is shown below

**CAML Query:**

```
#CREATE FROM CONDITIONS
Title Eq FIELD:WSSEMPID
```

**Root Template**:

```
<input type="hidden" id="WSSEMPID__[$id]" VALUE="" />
__RESULTSET__
```

**Result Template**:

```
__LISTITEMS__
```

**Row Template:**

(The HTML is a slightly modified version of the HTML generated by Sharepoint when the list row is displayed)

```
<table class="ms-formtable" style="margin-top: 8px;" border="0" cellpadding="0" cell-
spacing="0" width="100%">      <tr>
         <td nowrap="true" valign="top" width="165px" class="ms-formlabel"
style="BORDER-TOP: #bad6dd 1px solid;">
            <h3 class="ms-standardheader" style="margin-bottom:8px">
               <a name="SPBookmark_Title"></a>Initialer</h3>
         </td>
         <td valign="top" class="ms-formbody" width="450px" id="SPFieldText"
style="BACKGROUND: #f0f5f7;BORDER-TOP: #bad6dd 1px solid;">
            <!-- FieldName="Initialer"
                        FieldInternalName="Title"
                        FieldType="SPFieldText"
                  -->
            __[Initialer]__ 
         </td>
      </tr>
      <tr>
         <td nowrap="true" valign="top" width="165px" class="ms-formlabel"
style="BORDER-TOP: #bad6dd 1px solid;">
            <h3 class="ms-standardheader" style="margin-bottom:8px">
               <a name="SPBookmark_Fornavn"></a>Fornavn</h3>
         </td>
         <td valign="top" class="ms-formbody" width="450px" id="TD1" style="BACKGROUND:
#f0f5f7;BORDER-TOP: #bad6dd 1px solid;">
            <!-- FieldName="Fornavn"
                        FieldInternalName="Fornavn"
                        FieldType="SPFieldText"
                  -->
            __[Fornavn]__ 
         </td>
      </tr>
      <tr>
         <td nowrap="true" valign="top" width="165px" class="ms-formlabel"
style="BORDER-TOP: #bad6dd 1px solid;">
            <h3 class="ms-standardheader" style="margin-bottom:8px">
               <a name="SPBookmark_Efternavn"></a>Efternavn</h3>
         </td>
         <td valign="top" class="ms-formbody" width="450px" id="TD2" style="BACKGROUND:
#f0f5f7;BORDER-TOP: #bad6dd 1px solid;">
            <!-- FieldName="Efternavn"
                        FieldInternalName="Efternavn"
                        FieldType="SPFieldText"
                  -->
            __[Efternavn]__ 
```

```
         </td>
      </tr>
      <tr>
         <td nowrap="true" valign="top" width="165px" class="ms-formlabel"
style="BORDER-TOP: #bad6dd 1px solid;">
            <h3 class="ms-standardheader" style="margin-bottom:8px">
               <a name="SPBookmark_Telefon"></a>Telefon</h3>
         </td>
         <td valign="top" class="ms-formbody" width="450px" id="TD3" style="BACKGROUND:
#f0f5f7;BORDER-TOP: #bad6dd 1px solid;">
            <!-- FieldName=""
                        FieldInternalName="Telefon"
                        FieldType="SPFieldText"
                  -->
            __[Telefon]__ 
         </td>
      </tr>
      <tr>
         <td nowrap="true" valign="top" width="165px" class="ms-formlabel"
style="BORDER-TOP: #bad6dd 1px solid;">
            <h3 class="ms-standardheader" style="margin-bottom:8px">
               <a name="SPBookmark_Mobil"></a>Mobil</h3>
         </td>
         <td valign="top" class="ms-formbody" width="450px" id="TD4" style="BACKGROUND:
#f0f5f7;BORDER-TOP: #bad6dd 1px solid;">
            <!-- FieldName="Mobil"
                        FieldInternalName="Mobil"
                        FieldType="SPFieldText"
                  -->
            __[Mobil]__ 
         </td>
      </tr>
      <tr>
         <td nowrap="true" valign="top" width="165px" style="BORDER-TOP: #bad6dd 1px
solid;">
            <h3 class="ms-standardheader" style="margin-bottom:8px">
               <a name="SPBookmark_Afdeling"></a>Afdeling</h3>
         </td>
         <td valign="top" class="ms-formbody" width="450px" style="BACKGROUND:
#f0f5f7;BORDER-TOP: #bad6dd 1px solid;">
            <!-- FieldName="Afdeling"
                        FieldInternalName="Afdeling"
                        FieldType="SPFieldChoice"
                  -->
            __[Afdeling]__ 
         </td>
      </tr>
      <tr>
         <td nowrap="true" valign="top" width="165px" class="ms-formlabel"
style="BORDER-TOP: #bad6dd 1px solid;">
            <h3 class="ms-standardheader" style="margin-bottom:8px">
               <a name="SPBookmark_Billede"></a>Billede</h3>
         </td>
         <td valign="top" class="ms-formbody" width="450px" id="SPFieldURL"
style="BACKGROUND: #f0f5f7;BORDER-TOP: #bad6dd 1px solid;">
            <!-- FieldName="Billede"
                        FieldInternalName="Billede"
                        FieldType="SPFieldURL"
                  -->
            <img src="http://webtopintranet/k2/Pictures/Foto/__[Initialer]__.jpg"
alt=""> 
         </td>
      </tr>
      <tr>
         <td nowrap="true" valign="top" width="165px" class="ms-formlabel"
style="BORDER-TOP: #bad6dd 1px solid;">
            <h3 class="ms-standardheader" style="margin-bottom:8px">
               <a href="__[$ITEMURL]__" target="_blank">REDIGER..</a></h3>
         </td>
         <td valign="top" class="ms-formbody" width="450px" id="SPFieldURL"
style="BACKGROUND: #f0f5f7;BORDER-TOP: #bad6dd 1px solid;">
             
         </td>
      </tr>
</table>
```

**Unique Client ID:**

```
EmpDetails
```