

Standaard Lijst- en Detailschermen

Applicatiebouw

Portal beheerportaal-Nieuw. Tegel *Tabellen standaardAPI*.

Screenidentifiers (beheertegel *Schermkolomdefinitie tabellen standaard-API*)

- MDLC_getVwFrmSysStandardtableList.xml
- MDDC_getVwFrmSysStandardtableDetail.xml
- MDFC_getVwFrmSysStandardtableList.xml

Tabellen:

- tbSysStandardTable/VwfrmSysstandardTable
- tbSysstandardButton

Zie ook: [Uitgewerkt voorbeeld](#)

Omschrijving

Met behulp van de tabel tbSysStandardTable kan een applicatiebeheerder zelf nieuwe schermen op bestaande OpenWave tabellen/views definiëren.

De kaarten in de tabel tbSysStandardTable hebben een unieke codering. Elke kaart is een representatie van hoe een bepaalde view of tabel uit de OpenWave database moet worden getoond met die restrictie dat altijd gerenderd wordt conform de opmaak van een OpenWave-lijstschermben en OpenWave-detailschermben. Naast het definiëren van de tabel/view die getoond moet worden, en met welke parent, worden ook de verwijzing(en) naar de bijbehorende schermkolomdefinitie(s) vastgelegd in de kaart. In de dochtertabel tbSysstandardButton kunnen tenslotte de knoppen linksonder op de te leveren lijst-of detailpagina worden gedefinieerd met bijbehorende action. Eventueel kan ook een filter op het lijstschermben worden getoond. Zie hiervoor onderaan deze pagina *Voorbeeld filterdefinitie*.

Aanroep

- Aanroep voor het tonen van een **lijstschermben** op basis van de informatie van een kaart in tbSysStandardTable (dus bijvoorbeeld een action op een tegel):
getFlexList(SysStandardList,nil,nil,G,DitIsDeUniekeCodering)
- OF aanroep van een blok in een detailschermben, wanneer in dat blok een lijstschermben getoond moet worden: Dan moet de tag type van dat blok in de xml-definitie van dat detailschermben gevuld worden met bijv. *getFlexList(SysStandardList,,%keypointer%,,beheer_kopcompngem)*

Waarbij:

- de methode getFlexList() aangeeft dat het resultaat van de aanroep een lijstschermben is
- de eerste parameter *SysStandardList* aangeeft dat alle informatie verder op te halen is in de tabel tbSysStandardTable op basis van dvcode = param 5 (dus in de voorbeeldaanroep

beheer_kopcompngem of *DitIsDeUniekeCodering*)

- de tweede parameter wordt genegeerd. Kan dus leeg zijn
- de derde parameter (hier alleen gevuld in tweede voorbeeld) wordt alleen gevuld met de waarde van de primary key van de parenttable indien van toepassing (welke parenttable: dat staat in de betrokken kaart van *tbSysStandardTable*)
- de vierde parameter (hier in het eerste voorbeeld de waarde 'G') leeg mag zijn. De vulling met een 'A' of een 'G' betekent dat de lijst op moet starten met alle kaarten of alleen de geldige kaarten, indien gebruik gemaakt wordt van box vervallen kaarten onzichtbaar in de *tbSysStandardTable*-kaart
- de vijfde parameter, met hier de waarde *DitIsDeUniekeCodering*, geeft aan in welke kaart van *tbSysStandardTable* de gewenste informatie staat: *tbSysStandardTable.dvcode = DitIsDeUniekeCodering* of *beheer_kopcompngem*
- optioneel kan nog aangevuld worden tot de negende parameter: *paramtype*. Dit is alleen van toepassing als er bij het lijstscherf een filter xml gedefinieerd is EN gebruik wordt gemaakt van een query afweging bij een van de tags *<visible>* in het filter xml. De waarde van parameter 9 mag leeg zijn indien *<visible>* true is of gevuld met waarde '1' indien *<visible>* waarde false moet krijgen.
- Aanroep voor het tonen van een **detailscherf** op basis van de informatie van een kaart in *tbSysStandardTable* (dus bijvoorbeeld een action op een tegel of knop):
getFlexdetail(SysStandardDetail,1234,DitIsDeUniekeCodering) waarbij:
- de methode *getFlexDetail()* aangeeft dat het resultaat van de aanroep een detailscherf is
- de eerste parameter *SysStandardDetail* aangeeft dat alle informatie verder op te halen is in tabel *tbSysStandardTable* op basis van *dvcode = param 3*
- de tweede parameter de waarde van de primary key bevat van de tabel waarvan het detailscherf wordt opgevraagd
- de derde parameter, met hier de waarde *DitIsDeUniekeCodering*, geeft aan in welke kaart van *tbSysStandardTable* de gewenste informatie staat: *tbSysStandardTable.dvcode = DitIsDeUniekeCodering*.

De kolommen van de tabel *tbSysStandardTable*

- **Identifier** (*dnkey*). Primary key van de tabel.
- **Unieke codering voor tabel** (*dvcode*). De unieke codering van de kaart die bij de action-aanroep *getFlexList(sysStandardList,,,code)* als vijfde parameter moet worden meegegeven en bij de aanroep *getFlexDetail(sysStandardDetail,,code)* als derde parameter.
- **Systeem-categorie** (*dnkeysysstandardcat*). Foreign key naar de tabel *tbSysstandaardcategorie* om de standaardapi's beter in te delen. Niet verplicht.
- **Systeemkaart** (*dlsystem*). Niet muteerbaar. T of F. Indien T dan is de betreffende kaart voor de OpenWave-applicatie onmisbaar en mag de kaart niet verwijderd worden.
- **Hoofdtabel- of viewnaam** (*dvmaintablename*). De naam van de view of tabel waarvan een of meer kaarten in lijst of detail getoond moeten worden.
- **Kolomnaam van de primary key** (*dvmainprimkeyname*). De kolom naam van de primary key van de hoofdtabel/view.
- **Tabelnaam waarop hoofdtabel/view is gebaseerd** (*dvmainbasetablename*). Indien de hoofdtabel een view is dan dient hier de naam van de onderliggende hoofdtabel van die view genoteerd te worden. Indien de hoofdtabel een tabel is, dan staat hier hetzelfde als in de kolom *Hoofdtabel- of viewnaam*.
- **Kolomnaam foreign key uit deze achterliggende tabel** (*dvmainbaseforeignkeyname*). Deze kolom alleen invullen indien ook de kolom *parenttabelnaam* wordt gevuld. Het gaat in dat

geval om het weergeven van een lijst op basis van *Hoofdtabel-* of *viewnaam* waarbij deze lijst gelimiteerd is door een foreign key naar de parenttabel (bijv. de medewerkers van een rechtengroep). Hier dus de kolomnaam van de foreign key invullen uit de achterliggende tabel (dvmainbasetablename).

- **Parenttabelnaam** (dvparenttablename). De naam van de tabel (of view) die als parent fungeert voor de hoofdtabel/view.
- **Kolomnaam foreign key (uit hoofdtabel/view)** ((dvmainforeignkeyname). Ook deze kolom alleen invullen indien ook de kolom *parenttabelnaam* wordt gevuld. Het gaat in dat geval om het weergeven van een lijst op basis van *Hoofdtabel-* of *viewnaam* waarbij deze lijst gelimiteerd is door een foreign key naar de parenttabel (bijv. de medewerkers van een rechtengroep). Hier dus de kolomnaam van de foreign key invullen uit de hoofdtabel/view (dvmaintablename).
- **Kolomnaam blokkering uit parenttabel**(dvparentblokkeringfieldname). Betekent dat indien de achterliggende waarde van deze kolom gevuld is - en parenttable is van toepassing- , dat dan geen wijzigingen en inserts en deletes bij deze tabel kunnen plaatsvinden.
- **Kolomnaam blokkering (uit hoofdtabel/view)**(dvblokkeringfieldname). Betekent dat indien de achterliggende waarde van deze kolom gevuld is, dat dan geen wijzigingen en verwijderacties bij deze kaart kunnen plaatsvinden.
- **Schermidentificer voor lijst** (dvlistscreenfilename). De unieke naam met de schermkolominformatie die verwijst naar tbscreencolumns.dvscreenfilename (beheertegel *Schermkolomdefinitie tabellen standaard-api*). **LET OP:** de conventie in OpenWave voor lijsten is MDLC_getXXXXXXXXList.xml. Met de knop achter deze kolomnaam wordt naar de betreffende schermkaart in tbscreencolumns genavigeerd. Indien er nog geen kaart bestaat wordt deze automatisch aangemaakt.
- **Schermidentificer voor detail** (dvdetailscreenfilename). De unieke naam met de schermkolominformatie die verwijst naar tbscreencolumns.dvscreenfilename (beheertegel *Schermkolomdefinitie tabellen standaard-api*). **LET OP:** de conventie in OpenWave voor lijsten is MDDC_getXXXXXXXXDetail.xml. Met de knop achter deze kolomnaam wordt naar de betreffende schermkaart in tbscreencolumns genavigeerd. Indien er nog geen kaart bestaat wordt deze automatisch aangemaakt.
- **Schermidentificer voor filter** (dvfilterscreenfilename). De unieke naam met de schermkolominformatie die verwijst naar tbscreencolumns.dvscreenfilename (beheertegel *Schermkolomdefinitie tabellen standaard-api*). **LET OP:** de conventie in OpenWave voor filterschermen is MDFC_getXXXXXXXXList.xml. Met de knop achter deze kolomnaam wordt naar de betreffende schermkaart in tbscreencolumns genavigeerd. Indien er nog geen kaart bestaat wordt deze automatisch aangemaakt.
- **Kijkrechtenkolom (bijv. tbomgrechten.dlcomgadvvsvb)** (dvauthvisiblefield). Een verwijzing naar een rechtenkolom waarvan de waarde T moet zijn voor de inlogger om de gevraagde lijst of detailpagina te bekijken. Indien ingevuld gaat deze kolom voor op de kolom *tbqueries.dvcode kijkrechten*.
- **Muteerrechtenkolom (bijv. tbomgrechten.dlcomgadvedt)** (dvautheditfield). Een verwijzing naar een rechtenkolom waarvan de waarde T moet zijn voor de inlogger om de gevraagde lijst of detailpagina te muteren. Indien ingevuld gaat deze kolom voor op de kolom *tbqueries.dvcode wijzigrechten*.
- **tbqueries.dvcode kijkrechten (result = true)** (dvauthvisiblequerycode). Een verwijzing naar tbqueries.dvcode alwaar de SQL-statement een true of een false moet geven, hetgeen aangeeft of de inlogger de gevraagde lijst of detailpagina mag bekijken. Indien echter de kolom *Kijkrechtenkolom (bijv. tbomgrechten.dlcomgadvvsvb)* is gevuld dan wordt deze querykolom genegeerd.
- **tbqueries.dvcode wijzigrechten (result = true)** (dvautheditquerycode). Een verwijzing naar tbqueries.dvcode alwaar de SQL-statement een true of een false moet geven, hetgeen aangeeft of de inlogger de gevraagde lijst of detailpagina in beginsel mag muteren (in de

schermkolomdefinitie wordt daar per cel/kolom geautoriseerd). Indien echter de kolom *Muteerrechtenkolom* (bijv. *tbomgrechten.dlcomgadvedt*) is gevuld dan wordt deze querykolom genegeerd.

- **Module/schermgroepcode** (dvmodulescreengroup). Vrij te gebruiken. Alleen indien een vervolgaction van een standaardapi-lijst of detail een interne OW-API aanroept, met een verplichte parameter dvmodule, dan is een waarde in deze kolom ook verplicht (dvmodulescreengroup is dan B,C,E,H,I,O,V of W). Dit is bijvoorbeeld het geval bij een wijziging op een kolom (met de interne API-aanroep setcolumnvalue) op een detail of lijstscherf van een (dochter)tabel van tbomgvergunning (W), tbhandhavingen (H), tbinfoaanvragen (I), tbovvergunningen (O) en tbmilinrichtingen (V) en tbmilvergunningen (E) en tbhorecavergunningen (C) en tbbouwvergunningen (B). Bij beheertabellen kan deze kolom dus leeg blijven.
- **Datumkolomnaam box vervallen** (dvvervallenboxfieldname). Indien gevuld met een datumkolomnaam van de hoofdtabelview dan zal onderaan in de lijstweergave van die tabel/view een aanvinkbox *vervallen kaarten onzichtbaar* zijn. Indien onzichtbaar aangevinkt dan zal de lijst gefilterd worden op deze kolom is null.
- **Zoekbox?** (dlzoekbox). Indien aangevinkt dan zal een zoekbox onderaan de lijstweergave zichtbaar zijn.
- **Action;bij dubbel klik; op lijstregel** (dvactionselectlineinlist). De action die wordt aangeroepen wanneer de inlogger in de lijstweergave dubbelklikt op een regel. De phrase *{id}* zal daarbij door OpenWave automatisch vervangen worden door de waarde van de primary key (kolom dvmainprimkeyname) uit de hoofdtabel/view. Voorbeeld:
getFlexdetail(SysStandardDetail,{id},beheer_rechtsvorm) zal door OpenWave doorgezet worden als bijv. *getFlexdetail(SysStandardDetail,1234,beheer_rechtsvorm)*.
- **Detailvenster openen na insert met sysStandardRow?** (dlldetailopenennaininsert). Indien 'T' dan wordt na een insert met insertSysStandardRow (zie hieronder bij sysstandardbuttons) de action *getFlexdetail(SysStandardDetail,{id},dvcode)* uitgevoerd, waarbij dvcode wordt vervangen met de waarde van dvcode van de betreffende sysstandardtablekaart en {id} met de nieuwe dnkey aangemaakt met de insert.
- **Where clause;bij lijst** (dvwhere). Indien het gaat om een lijst dan kan hier een extra where clause worden opgegeven waaraan de hoofdtabel/view moet voldoen. Bijvoorbeeld bij een medewerkerslijst kan hier staan: *where dvgeslacht = 'M'*. De phrases:
 - *:keyaccount* zal door OpenWave automatisch worden vervangen door *tbmedewerkers.dvcode* van de inlogger
 - *%keyaccount%* idem
 - *%inlogger%* idem
 - *%keyparent%* met de primary key van de bovenliggende tabel indien de standaardlijst wordt aangeroepen in een blok vanuit een detailscherf.

Zie verder over het gebruik en mogelijkheden van actions: [Actions](#).

Voorbeeld gebruik rechtenkolom op basis van query

Het gaat hier om het vullen van de kolom *tbqueries.dvcode wijzigrechten* (*dvautheditquerycode*) of *tbqueries.dvcode kijkrechten* (*dvauthvisiblequerycode*) waarbij de uitkomst true of false bepalend is voor het kijk- of wijzigrecht van het scherm. Ook bij de knoppen (zie hieronder) wordt van deze systematiek uitgegaan.

Voorbeeld

wanneer het inzien of wijzigen van een scherm voorbehouden moet zijn aan een functioneel beheerder is er een verwijzing naar de systeemquery met dvcode = *sysstandaard_isbeheerder* het makkelijkst. Deze standaard uitgeleverde query retourneert true indien de medewerker beheerniveau 99 heeft.(*tbmedewerkers.dnbeheerniveau*).

Wanneer het inzien of wijzigen van een scherm gekoppeld moet zijn aan de autorisaties van een medewerker rekening houdend met compartiment is het volgende voorbeeld met gebruik functies *fn_rechtenkolom* en *fn_iscompartimentok* van toepassing.

In de kolom *dvauthvisiblequerycode* wordt naar een query uit *tbqueries* verwezen bijv. met de code *omgeving_milalertmuteren*

De query met dvcode = *omgeving_milalertmuteren* kan dan als volgt gedefinieerd zijn:

```
select case when (fn_iscompartimentok(:keyaccount, 'W',{id}) = 1)
           and (fn_rechtenkolom('tbomgrechten.dlbomgmoedt',:keyaccount) =
'T' )
       then 'true'
       else 'false'
       end
```

De query maakt gebruik van twee OpenWave functies:

fn_iscompartimentok kijkt op grond van de inlogger, de module en de dnkey van de kaart of de compartmentsrechten in de weg zitten. De string *:keyaccount* wordt onder water altijd vervangen door de dvcode van de inlogger. De string (*id*) altijd door de dnkey van de kaart waar de gebruiker op staat in de hoofdtabel (*dvmaintablename*).

fn_rechtenkolom kijkt op grond van de aangegeven rechtenkolom (rechtentabel gevolgd door een punt gevolgd door de kolomnaam) of de inlogger (*:keyaccount*) wijzigrechten heeft volgens die rechtenkolom .

Zie ook [Database functies](#)

Controle op valide schermverwijzingen

In het servicecentrum portaal onder de kolom Notificaties is een tegel *Ontbrekende sysstandardschermen in AAR* gedefinieerd. Met deze tegel wordt een lijst gegenereerd van schermoproepen (lijstschermen of detailschermen of filterschermen of insertstandardrowschermen) in *tbsysstandardtable* en/of *tbsysstandardbutton* (dus in de tabel achter de tegel *tabellen standaardapi* van het nieuwe beheerportaal onder de kolom Scherm- en tegelbeheer) die niet zijn opgenomen in de AAR.

Dit zijn schermen die niet met implementatie en updates van OpenWave zijn aangeleverd. Dat kan

zijn omdat de schermen door een functioneel beheerder zelf zijn gedefinieerd: in de tabel *tbscreencolumns* (tegel *Schermkolomdefinitie tabellen standaard-api*) is de kolom *dvscreenxml* in dat geval gevuld met de eigen opmaak. In bovengenoemde controlelijst is dat zichtbaar indien de kolom **Afwijkend scherm** aangevinkt is. Er gaat dus pas iets mis indien een regel in deze lijst is opgenomen zonder dat de kolom *Afwijkend scherm* is gevuld. Een reden is vaak dat de verwijzing en benaming van de feitelijke opmaakxml-file in de AAR van elkaar verschillen in kameleannotatie.

Filterdefinitie bij lijstschermb

Zie: [Definitie filterblokken op lijstschermb](#). Indien er gewenst is dat het lijstschermb gefilterd kan worden zal er een filter xml moeten worden gedefinieerd. De naam van de xml moet beginnen met 'MDFC_' en de rest van de naam moet gelijk zijn als de xml-naam van het lijstschermb (zonder de prefix MDDL_).

Ga naar gewenste kaart in *tbsysstandardtable* (beheertegel: *Tabellen standaardAPI*):

- Zet bij **Schermidentificer voor filter** (bijv.: *MDFC_getTest_MWPerRechtengroepList.xml* want in voorbeeld is de lijst.xml *MDLC_getTest_MWPerRechtengroepList.xml*)
- Klik op de knop 'Ga naar schermdefinitie'
- In de kolom **Kolominformatie Toggle F11** moet vervolgens de layout van het filterschermb gedefinieerd worden in xml-formaat met een of meer kolommen uit het lijstschermb. In het voorbeeld worden twee filters gemaakt.

Knoppen op lijst- en detailschermb

Knoppen die binnen een detailschermb dat door *tbsysstandardtable* wordt gedefinieerd- bijv. achter een specifieke kolom - moeten verschijnen, worden in de xml van dat detailschermb gedefinieerd inclusief de actions die aan die knoppen verbonden moeten zijn: dus - in bovenstaand voorbeeld - in de *MDFC_getTest_MWPerRechtengroepList.xml*. Zie [Scherm informatie voor detailschermb](#).

De knoppen die linksonder op het gedefinieerde lijst- of detailschermb moeten komen, kunnen binnen de detailkaart van *tbSysStandardTable* in het blok *Knoppen* worden gedefinieerd. Deze informatie wordt in de tabel *tbSysStandardButton* opgeslagen: een dochtertabel van *tbsysstandardtable*.

Per knop zijn de volgende kolommen beschikbaar:

- Blok Identifier:
 - **Systeemkaart** (dlsystem). Indien aangevinkt dan is de kaart bij een update door REM aangemaakt en onontbeerlijk voor goede werking van het programma. Niet aankomen dus.
- Blok Knop:
 - **Hint**. Deze tekst verschijnt als hint bij de knop, of als omschrijving van de knop indien onderdeel van itemlist.
 - **Lijst of Detail**. Een L of een D. Indien L dan is de knop zichtbaar op het gedefinieerde lijstschermb. Bij D dus alleen op het detailschermb.
 - **Linksonder of Itemlijst**. Een L of een I. Indien L dan verschijnt de knop met een icoon linksonder aan de pagina. Indien I dan verschijnt de knop als item met als omschrijving de

- Hint in een itemlijst (Opties-menu) rechtsboven aan de pagina.
- **Icoonnummer.** Alleen van toepassing indien (L)inksonder. Hier moet een nummer komen uit de lijst: [Iconenlijst](#).
 - **Volgorde.** Met deze numerieke waarde kan de volgorde van de knoppen van links naar rechts of - indien itemlist- van boven naar beneden bepaald worden.
 - **Refresh.** Indien aangevinkt dan zal het scherm na het uitvoeren van de action bij de knop opnieuw worden uitgeschreven.
 - **Blok Rechten:**
 - **Action execute-rechtenkolom** (dvauthexecuteveld). Een verwijzing naar een rechtenkolom waarvan de waarde T moet zijn voor de inlogger om de de action die aan de knop vastzit te mogen uitvoeren. Indien ingevuld gaat deze kolom voor op de kolom *tbqueries.dvcode actionrechten*.
 - **tbqueries.dvcode action execute (result = true)** (dvauthexecutequerycode). Een verwijzing naar *tbqueries.dvcode* alwaar de SQL-statement een true of een false moet geven, hetgeen aangeeft of de inlogger de action bij de knop mag uitvoeren. Indien echter de kolom *execute-rechtenkolom* (bijv. *tbomgrechten.dlcomgadvvsvb*) is gevuld dan wordt deze querykolom genegeerd.
 - **Blok Action en parameters.** De eerste kolom is de naam van de aan te roepen methode. De volgende kolommen worden gevuld met één of meer vereiste parameters. Voor alle action/parameters geldt dat 'on the fly' OpenWave de variabelen:
 - %:keyaccount% zal vervangen met de waarde van *tbmedewerkers.dvcode* van de inlogger
 - %inlogger% met de waarde van *tbmedewerkers.dvcode* van de inlogger
 - %keypointer% met de waarde van de primary key van de kaart waar de gebruiker op dat moment op staat (alleen bij knoppen op een detailscherm)
 - %keyparent% met de waarde van de primary key van de parenttabelkaart
 - %query(querynaam)% wordt vervangen door resultaat van de query met naam *querynaam* (alleen bij knoppen op een detailscherm)
 - %query(querynaam,%keypointer%)% wordt vervangen door resultaat van de query met naam *querynaam*, waarbij de string {id} in de query eerst wordt vervangen met de waarde van de primary key van de kaart waar de gebruiker op dat moment op staat (alleen bij knoppen op een detailscherm)
 - { + kolomnaam uit hoofdvew/tabel + } wordt vervangen door de achterliggende waarde van die kolomnaam voor de actieve rij (zowel detail als lijstschermb). Dus stel dat de hoofdvew/tabel = *tbomgvergunning* en een van de parameters heeft de waarde {dvzaakcode} dan zal die parameter vervangen worden met de achterliggende waarde van *dvzaakcode* voor de actieve rij.

Knop Standaard insertscherm

Om een knop te maken met als doel een standaard insert op een tabel moet bij de knopdefinitie de action startWizard aangeroepen worden met:

- als eerste parameter de tekst *insertSysStandardRow*
- als tweede parameter de schermnaam van een xml waarin het insertscherm is gedefinieerd. Bijv. *MDWC_insertTbKopCompGem.xml*. Deze naam moet beginnen met 'MDWC_' en eindigen op '.xml'. De xml met daarin de scherm informatie moet opgenomen worden in de tabel *tbscreencolumns*. OpenWave maakt zelf een kaart aan in deze tabel als deze niet bestaat. Indien (*insertSysStandardRow*) zal het programma op het detailscherm van de knop achter de tweede parameter een verwijsknop naar deze *screencolumns* kaart plaatsen

- de derde parameter is LEEG indien de tabel waarop een insert plaatsvindt GEEN parenttabel heeft. Indien deze tabel echter wel een parenttabel heeft dan moet deze parameter gevuld worden met de tekst %keyparent%. De tekst %keyparent% wordt door OpenWave 'on the fly' vervangen met de primary key van de parenttabel
- als vierde parameter een verwijzing naar de unieke codering van de kaart uit tbsysstandardtabel waar deze knopdefinitie bij hoort.

Voorbeeld

Een voorbeeld van de aanroep van insertSysStandardRow in de basis uitlevering van OpenWave is de insert-knop op het lijstscherf van medewerkers verbonden aan een team.

Zie voor de definitie hiervan: beheerportaal, onder de kolom *Scherf-en tegelbeheer* de tegel *Tabellen Standaardapi* en vervolgens de rij in tbsysstandardtable met dvcode = *beheer_vwfrmmwteams*. Het gaat om de knop *Voeg medewerker toe*.

Het resultaat is te gebruiken in het beheerportaal onder de kolom *Gebruikers* de tegel *Teams*. Op het detailscherf van een team staat de lijst met medewerkers bij dat team. Aan de insert-knop is dus de InsertSysStandardRow-functie verbonden.

Voor de opmaak van standaard insertscherf (de xml) zie: [Schermf informatie voor standaard insert- en kopieer](#).

Knop Standaard kopieerscherf

Om een knop te maken met als doel een standaard insert op een tabel te doen als kopie van de kaart waar de gebruiker op staat moet bij de knopdefinitie de action startWizard aangeroepen worden met:

- als eerste parameter de tekst *kopieerSysStandardRow*
- als tweede parameter de schermnaam van een xml waarin het insertscherf is gedefinieerd. Bijv. MDWC_insertTbKopCompGem.xml. Deze naam moet beginnen met 'MDWC_' en eindigen op '.xml'. De xml met daarin de schermf informatie moet opgenomen worden in de tabel tbscreencolumns. OpenWave maakt zelf een kaart aan in deze tabel als deze niet bestaat. Indien (kopieerSysStandardRow) zal het programma op het detailscherf van de knop achter de tweede parameter een verwijfsknop naar deze screencolumns kaart plaatsen
- de derde parameter is bevat de primary key van de kaart waarop de gebruiker staat. Hier in te geven met de tekst {id}. Die tekst id wordt door OpenWave 'on the fly' vervangen met de primary key van de te kopiëren kaart
- als vierde parameter een verwijfsknop naar de unieke codering van de kaart uit tbsysstandardtabel waar deze knopdefinitie bij hoort.

Voorbeeld

Een voorbeeld van de aanroep van `kopieerSysStandardRow` in de basis uitlevering van OpenWave is de kopieer-knop op het lijstscherf van de zaaktypes.

Zie voor de definitie hiervan: beheerportaal, onder de kolom *Scherf-en tegelbeheer* de tegel *Tabellen Standaardapi* en vervolgens de rij in `tbsysstandardtable` met `dvcode = beheer_vwfrmsoortomgvergs`. Het gaat om de knop *Kopieer zaaktype*.

Het resultaat is te gebruiken in het portaal Zaakbeheer onder de kolom *Zaaktypes* de tegel *Zaaktypes*. Op het lijstscherf van de omgevingzaaktypes staat de kopieer-knop waar dus de `kopieerSysStandardRow`-functie aan verbonden is.

Voor de opmaak van standaard insertscherf (de xml) zie: [Scherm informatie voor standaard insert- en kopieer](#).

Knop Standaard verwijderen van een kaart

Om een knop te maken met als doel een standaard verwijderactie op een kaart van een tabel moet bij de knopdefinitie de action `startWizard` aangeroepen worden met:

- als eerste parameter de tekst `deleteSysStandardRow`
- als tweede parameter de tabelnaam gevolgd door een punt gevolgd door `{id}` die `{id}` wordt on the fly' door OpenWave vervangen met primary key van de kaart die verwijderd moet worden
- als derde parameter een kolomnaam uit de view of tabel die aan de lijst ten grondslag ligt, waarvan de achterliggende waarde gebruikt wordt voor de *weet u zeker* tekst
- als vierde parameter de code uit `tbsysstandardtable` die verwijst naar de kaart waar de betreffende standaardlijst in is gedefinieerd.

Voorbeeld

Een voorbeeld van de aanroep van `deleteSysStandardRow` in de basis uitlevering van OpenWave is de min-knop op het lijstscherf van de adviesinstanties:

Zie voor de definitie hiervan: beheerportaal, onder de kolom *Scherf-en tegelbeheer* de tegel *Tabellen Standaardapi* en vervolgens de rij in `tbsysstandardtable` met `dvcode = beheer_tbadviesinstanties`. Het gaat om de knop *Verwijder adviesinstantie*.

Het resultaat is te gebruiken in het beheerportaal onder de kolom *Gebruikers* de tegel *Adviesinstanties*. Op het lijstscherf is de min-knop hier dus verbonden aan de `deletesysstandardrow`-functie.

Deze methode `deleteSysStandardRow` kijkt naar de voorwaarden gedefinieerd in de kaart uit `tbsysstandardtable` met `dvcode =` de vierde parameter.

In deze kaart kan de *Kolomnaam blokkering uit parenttabel* gevuld zijn, hetgeen betekent dat indien de achterliggende waarde van deze kolom gevuld is - en `parenttable` is van toepassing- , dat dan de verwijderactie niet door kan gaan. In deze kaart kan de *Kolomnaam blokkering uit hoofdtabel/view*

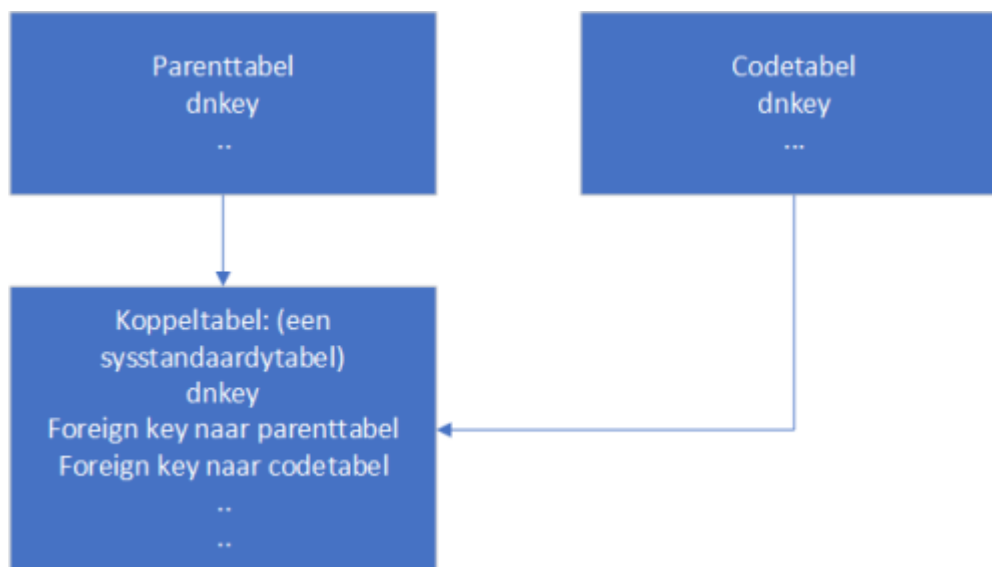
gevuld zijn, hetgeen betekent dat indien de achterliggende waarde van deze kolom gevuld is, dat dan de verwijderactie niet door kan gaan.

In de dochtertabel tbsysstandardbutton is bij de betreffende deletesysstandardrowkaart gedefinieerd naar welke rechten het programma dient te kijken.

Voor verwijderacties op de hoofdtabellen houdt OpenWave rekening met compartiment.

OpenWave waarschuwt ook met naam en toenaam dat een verwijderactie niet plaats kan vinden indien er een foreign key in de weg zit.

Knop insert of synchroniseer vanuit een derde codetabel



Toepassingen:

- Inserten in een sysstandaardtabel van meerdere kaarten op basis van een selectie uit een derde codetabel
- Of synchroniseren van de rijen in een sysstandaardtabel op basis van een selectie uit een derde codetabel

In beide gevallen is de sysstandaardtabel in kwestie een dochtertabel van een parent.

Voorbeeld

Een voorbeeld van synchronisatiegebruik in de basis uitlevering van OpenWave is de koppeling van documentsjablonen aan een of meer groepen:

Zie voor de definitie hiervan: beheerportaal, onder de kolom *Scherf-en tegelbeheer* de tegel

Tabellen Standaardapi en vervolgens de rij in tbsysstandardtable met dvcode =

beheer_vwfrmkopdocaandocsoort. Het gaat om de knop *Sjabloon koppelen aan sjabloongroepen*.

Het resultaat is te gebruiken in het beheerportaal onder de kolom *Werkbeheer* de tegel

Documentsjablonen. Op het detailscherm van een willekeurig sjabloon is een lijst opgenomen van de groepen waaraan dat sjabloon is verbonden. De plusknop is hier dus verbonden aan de koppelsysstandardrow-functie.

- **startwizard(koppelSysStandardRow,param2,param3,param4)**

- param1: koppelSysStandardRow
- param2: Bestaat uit vier delen gescheiden door een puntkomma:
 - de eerste deelparameter is de waarde van de parentkey naar de parenttabel (in te geven als %keyparent% en wordt on the fly vervangen met de echte waarde)
 - de twee deelparameter is een veldnaam uit de parenttabel waarvan de waarde gebruikt wordt in de koptekst
 - de derde deelparameter is optioneel en bevat - indien gevuld - een veldnaam uit de parenttabel die een moduleletter bevat. Mag dus leeg zijn
 - de vierde deelparameter is leeg of 0 of 1. Indien 0 of leeg dan worden de rijen in de standaardtabel (de koppeltabel) gelijkgemaakt (gesynchroniseerd) met de aangevinkte rijen van de codetabel. Sommige worden dus verwijderd, sommige worden toegevoegd. Indien de waarde 1 dan worden alle aangevinkte rijen altijd toegevoegd aan de standaardtabel.
- param3: Bestaat uit acht delen gescheiden door een puntkomma:
 - de eerste deelparameter is de tabelnaam van de codetabel
 - de tweede deelparameter is de veldnaam van de foreign-key kolom uit de koppeltabel naar de codetabel
 - de derde deelparameter is de veldnaam van de primary key van de codetabel
 - de vierde deelparameter is een veldnaam uit de codetabel die getoond kan worden in het aanvinkscherm. Mag leeg zijn
 - de vijfde deelparameter is een veldnaam uit de codetabel die getoond moet worden in het aanvinkscherm. Mag NIET leeg zijn
 - de zesde deelparameter is optioneel en bevat - indien gevuld - een veldnaam uit de codetabel die een moduleletter bevat. Mag dus leeg zijn.
 - de zevende deelparameter is optioneel en bevat - indien gevuld - een veldnaam uit de codetabel die gevuld kan zijn met een vervaldatum. Indien gevuld dan zijn de vervallen items niet zichtbaar in het keuzelijstscherf
 - de achtste deelparameter is optioneel en bevat een extra where clause die aan de query die ten grondslag aan de keuzelijst wordt toegevoegd. Bijvoorbeeld *substr(dvcode,1,1) = 'R'* of *dnminwaarde > 40*
- param4: de code uit tbsysstandardtable die verwijst naar de kaart waar de betreffende tabel in is gedefinieerd.

In beide toepassingsgevallen kan de gebruiker één of meer rijen aan- of uitvinken van de codetabel in een wizardscherf.

De selectie van rijen op die codetabel wordt beïnvloed door:

- Indien de zevende deelparameter van param3 is gevuld met de kolomnaam van een datumveld dan gaat OpenWave er van uit dat het hier gaat om een vervaldatum en worden alleen die kaarten uit de codetabel getoond die niet vervallen zijn.
- De moduleletter (de derde deelparameter van param2 die verwijst naar kolomnaam van de parenttabel, en de zesde deelparameter van param3 die verwijst naar kolomnaam van de codetabel) wordt - indien beide deelparameters gevuld - om de rijen van de codetabel te filteren op de moduleletter van de parenttabel. Mogen dus leeg zijn. Let op de inhoud van de modulekolom van de parenttabel moet in dat geval gevuld zijn met één letter namelijk: W,A,O,I,H of E).
- De rijen van de codetabel kunnen aan nog een voorwaarde moeten voldoen indien de achtste deelparameter van param3 gevuld is met een statement dat gebruikt kan worden in de where clause van de SQL die ten grondslag ligt aan het oproepen van de codetabelrijen.

De vierde deelparameter van param2 (met de waarde 0, leeg of 1) is bepalend voor wat OpenWave met de selectie van rijen uit de codetabel gaat doen: Indien

- leeg of de waarde 0 dan zal OpenWave synchroniseren. Dat betekent dat op basis van de NIET aangevinkte rijen OpenWave kijkt of er rijen met foreign keys bestaan in de sysstandaardtabel (de koppeltabel) met een verwijzing naar zo'n niet aangevinkte rij. Zo ja dan worden deze verwijderd uit de koppeltabel. Vervolgens wordt gekeken of er aangevinkte rijen zijn in de codetabel die nog geen verwijzing hebben vanuit de standaardtabel. Zo ja worden op basis daarvan nieuwe kaarten aangemaakt in de syststandaardtabel (de koppeltabel) met gevulde foreign key naar de parenttabel en gevulde foreign key naar de codetabel.
- de waarde 1 dan zal OpenWave op basis van elk van de aangevinkte rijen een kaart aanmaken in de sysstandaardtabel (de koppeltabel) met gevulde foreign key naar de parenttabel en gevulde foreign key naar de codetabel.

From:
<https://acc-doc.open-wave.nl/> - Documentatie

Permanent link:
https://acc-doc.open-wave.nl/doku.php/openwave/1.30/applicatiebeheer/instellen_inrichten/standardlist_standarddetail

Last update: **2025/05/26 12:28**