

Actions

Applicatiebouw

Wat is een action

Een action is een aanroep naar een methode van OpenWave met de juiste parameters. Die action wordt gekoppeld aan een trigger (zoals een knop, of een tegel of dubbelklikken op een lijstregel). OpenWave roept de methode aan en het gevolg is dat een lijst- of detailscherm wordt geopend, of een nieuw tabblad met een URL, of een wizard die wordt opgestart, of een kaartje dat getoond wordt, een memo.... et cetera.

Waar worden actions gedefinieerd

Op vele plekken in OpenWave kan een action aan een trigger worden gekoppeld. Bijvoorbeeld:

- in de xml van een schermkolomdefinitie van een detailscherm, waarbij een action wordt gedefinieerd bij een knop die binnen het scherm wordt opgenomen
- in de xml van een schermkolomdefinitie van een lijstschermbij een knop die binnen de lijst op elke regel wordt opgenomen
- in de tegeldefinitie (beheertegel *Portal*) waarbij de action wordt aangeroepen indien de gebruiker op de tegel drukt
- in een rapportdefinitie (beheertegel *Rapportagedefinitie*) waarbij de action gekoppeld wordt aan het dubbelklikken op de rapportageregel
- in de definitie van een proces-termijnstap (beheertegel *Processen*) waarbij de action gekoppeld wordt aan een knop bij een processtap
- in de definitie van lijst- en detailschermen via de beheertegel *Tabellen Standaardapi*, waarbij de actions kunnen worden gekoppeld aan:
 - het dubbelklikken op regel van lijstschermbij
 - de knoppen van het lijstschermbij en die van het detailschermbij.

Syntax en substitutie van variabelen

In de meeste gevallen moet de action (methode + parameters) in één string worden gedefinieerd zoals: `geefZaakMemo(12234,W)` of `getFlexList(SysStandardList,nil,nil,G,beheer_tbdocumenttype)`. Alleen de actions gekoppeld aan knoppen van een standaardlijst of standaarddetail (dus via de beheertegel *Tabellen standaard API*) zijn opgesplitst in een kolom action, waarin dit keer alleen de methodenaam moet komen, en kolommen voor de parameters: elke parameter een eigen kolom.

OpenWave kan bij de aanroep van een action binnen die action-definitie de volgende substrings tegenkomen:

- `%query(querynaam)%` of `%query(querynaam,%keypointer%)%`. In dat geval zal het SQL-statement van de betreffende query (beheertegel *Queries*) eerst worden geëvalueerd en het resultaat wordt gesubstitueerd in de action-definitie en daarna wordt de action uitgevoerd. Toepassing: schermdefinitie van de omgevingsdetailkaart met de knop *Ga naar OLO-loket* met

de volgende action-definitie:

openTabPage(<https://www.omgevingsloket.nl/BevoegdGezag/bevoegdgezag/AanvraagTab/Aanvraag/%query>(omgeving_olnummer,%keypointer%)/AanvraagGegevens)

- de substring %keypointer% kan alleen worden gesubstitueerd (altijd met de primary key van het record dat actief is) wanneer de action wordt aangeroepen vanuit een detailscherm
- de substring %keyparent% kan alleen worden gesubstitueerd (altijd met de primary key van het moederrecord dat actief is) wanneer de action wordt aangeroepen vanuit een lijst- of detailscherm dat als dochter is gedefinieerd
- %keyaccount% zal vervangen worden met de waarde van tbmedewerkers.dvcode van de inlogger
- %inlogger% met de waarde van tbmedewerkers.dvcode van de inlogger
- {id} kan alleen worden gebruikt bij een action die wordt uitgevoerd vanuit een lijstscherf (bijv. dubbelklikken op regel). De substring {id} wordt vervangen met de regel-id uit de lijst waar de gebruiker op dat moment op staat, die - meestal- de primary key bevat van de tabel waarop de lijst is gebaseerd. Toepassing: Zie het record in standaardapi-tabel met code *beheer_queries*.
- {kolomnaam_uit_onderliggendeview/tabel}. Werkt hetzelfde als {id} bij een action die wordt uitgevoerd vanuit een lijstscherf, bijv. {dnkeyopslag} waarbij deze string zal worden vervangen door de waarde van de kolom dnkeyopslag van de betreffende rij.

Welke OpenWave methodes kunnen worden gebruikt?

Voor de onderstreepte items geldt dat de substring *{id}* of *{kolomnaam}* in de action - indien aangeroepen van een lijstscherf - vooraf vervangen zal worden door de betreffende regel id (de waarde van de primary key van de regel) c.q. de waarde van de genoemde kolomnaam.

openTabPage

- Wat doet het?: Opent een nieuw tabblad met een kaart, externe website of een intern OpenWave portaal
- aanroep: openTabPage(param1)
 - Indien:
 - param1 begint met de substring (http) dan wordt geacht dat param1 een URL bevat bijvoorbeeld <https://www.open-wave.nl/>. Let op dat een ampersand-teken (&) in de URL geschreven moet worden als & (5 karakters)
 - param1 bedoeld is een specifiek portaal binnen het domein van OpenWave te openen moet de portaalnaam en de dnkey worden doorgegeven bijv. openTabPage(#omgevingdetail/x) waarbij de x staat voor een specifieke dnkey van de hoofdtabel van het portaal. Naast omgevingdetail kunnen hier de portaalnamen handhavingdetail, apvoverigdetail, infodetail, horecadetail en inrichtingdetail en milieugebruikdetail worden gebruikt.
 - param1 de waarde #kaartoverzicht heeft, dan zal OpenWave de interne kaart openen met de gedefinieerde kaartlagen uit de beheertabel *GEO kaartlagen*. Zie o.a. voor centrering [Kaart](#)
 - param1 opgebouwd is als URI-aanroep voor openen of bewerken van een fileserver-document via een Microsoft-office pakket, dan zal OpenWave deze aanroep doorzetten in de URL-balk (met de juiste slashes) b.v. ms-word:ofe| u

|file:/oxygen/users/pdeboer/LocalFileLinksTest.docx.

Waarschuwing

Indien openTabPage vanuit een tegelaction wordt aangeroepen kunnen de parameters niet via een query aanroep worden gesubstitueerd. Dat moet in dat geval iets ingewikkelder via een flexaction: bijvoorbeeld de action op de tegel is *getflexaction(omgeving_complex_oorsprong,{id})* waarbij de aangeroepen query (in dit voorbeeld *omgeving_complex_oorsprong*) een openTabPage-aanroep dient te construeren zonder te substitueren variabelen zoals: *select 'openTabPage(#omgevingdetail' | | dnkeyparentverg | | ')' from tbomgvergunning where dnkey = {id}*

getFlexList

- Wat doet het?: In een modal venster wordt een standaard lijstscherf over een view of tabel getoond. Zie: [Standaard Lijst- en Detailschermen](#)
- aanroep: *geefFlexList(param1, param2,param3,param4,param5)*:
 - param1: De eerste parameter moet de waarde *SysStandardList* bevatten. Er zijn andere mogelijkheden maar die worden hier niet besproken
 - param2: De tweede parameter kan leeg blijven
 - param3: Indien de lijst een dochtertabel is van een andere tabel EN alleen die kaarten die horen bij een specifieke moederkaart getoond moeten worden, dan moet hier de primary key waarde komen van die moederkaart. Anders, de lijst wordt zelfstandig in een modal getoond zonder moeder, dan kan deze parameter leeg blijven
 - param4: Heeft de waarde leeg, G of A. De lijst wordt met G opgestart met alle niet vervallen kaarten. Met A wordt de lijst gestart met alle kaarten. Het programma kijkt daarbij naar de *vervallen-box kolomnaam* in de bijbehorende definitie van standaardapi-tabel
 - param5: De vijfde parameter moet verwijzen naar een dvcode uit de standaardapi-tabel (beheertegel *Tabellen Standaardapi*). Aldaar haalt OpenWave alle informatie op over de lijst (inhoud, lay-out, autorisatie en triggers)
- autorisatie: Wordt in de aangewezen kaart van de standaardapi-tabel geregeld
- voorbeeld aanroep lijst binnen een detailscherm van moeder:
getFlexList(SysStandardList,,%keypointer%,beheer_kopcompgem)
- voorbeeld aanroep vanuit action op tegel zelfstandig lijstscherf
getFlexList(SysStandardList,,,beheer_compartiment).

getFlexDetail

- Wat doet het?: In een modal venster wordt een standaard detailscherm over een view of tabel getoond. Zie: [Standaard Lijst- en Detailschermen](#)
- aanroep: *geefFlexDetail(param1, param2,param3)*:
 - param1: De eerste parameter moet de waarde *SysStandardDetail* bevatten. Er zijn andere mogelijkheden maar die worden hier niet besproken
 - param2: De tweede parameter bevat de dnkey van de kaart waarvan het detailscherm wordt opgeroepen
 - param3: De derde parameter moet verwijzen naar een dvcode uit de standaardapi-tabel (beheertegel *Tabellen Standaardapi*). Aldaar haalt OpenWave alle informatie op over het detailscherm (inhoud, lay-out, autorisatie en triggers)
- autorisatie: Wordt in de aangewezen kaart van de standaardapi-tabel geregeld

- voorbeeld aanroep bij trigger dubbelklikken op rij in een lijst:
`getFlexdetail(SysStandardDetail,{id},beheer_compartiment).`

getFlexAction

- Wat doet het?: Handig bij knoppen op lijsten, waarbij de action pas gedefinieerd kan worden nadat de gebruiker een regel actief heeft gemaakt. De uiteindelijk uit te voeren action wordt met deze methode eerst opgehaald uit een gedefinieerde query, waarbij vooraf de substring {id} van die query wordt gesubstitueerd door de waarde van param2. Indien param2 de waarde {id} heeft dan zal OpenWave deze vervangen door de identifier van de betreffende regel: meestal de dnkey van de tabel (de methode moet dan vanaf bijv. een schermknop op een lijst worden aangeroepen)
- aanroep: `getFlexAction(param1, param2):`
 - param1: De eerste parameter moet een bestaande dvcode uit de tabel tbqueries bevatten
 - param2: De tweede parameter is {id} of bevat een waarde waarmee de substring {id} van die query wordt gesubstitueerd
- autorisatie: Wordt geregeld in de autorisatiekolommen van tbqueries en anderzijds - indien mogelijk - door de API die door de uiteindelijke action wordt aangeroepen

Waarschuwing

Het is raadzaam dat wanneer de action startwizard(showtekst) wordt gebruikt, waarbij deze een tekst ophaalt uit een kolom waarbij harde returns en linefeeds kunnen worden gebruikt (bij voorbeeld een memo), deze aanroep te laten lopen via `getFlexAction`.

Voorbeeld getflexaction en showtekst

Een voorbeeld van het gebruik van een `getFlexAction` in de definitie van het lijstschermbij de processtappen (termijnbewakingstappen) bij een zaak, zoals die default wordt uitgeleverd: De kolom `void` in `MDLC_GeefProcessenoverzicht` toont een klikbaar vraagtekenicoon indien de kolom `dvprocitemtoelichting` op de regel is gevuld (de query achter `celvisible`) EN indien de instelling *Sectie: Termijnbewaking Item: ToelichtingZichtbaar* is aangevinkt (de query achter `visible`).

```
<column tagnaam="void">
  <label></label>
  <index>80</index>
  <length>60</length>
  <wavetype>schermmknop</wavetype>
  <icoon>28</icoon>
  <showhint>>false</showhint>
  <action>getFlexAction(termijnbewaking_toelichting,{id})</action>
  <visible>%query(termijnbewaking_toelichtingzichtbaar)%</visible>
<celvisible>%query(termijnbewaking_celtoelichtingzichtbaar,{id})%</celvisibl
e>
</column>
```

Met het aanklikken van het vraagtekenicoontje wordt de query *termijnbewaking_toelichting* geëvalueerd die aangeroepen wordt met de *getFlexAction* met de {id} ofwel de dnkey van *tbtermijnbewstappen-rij* als parameter. Die query:

```
select startWizard(showtekst,beginarg(' || coalesce(dvprocitemtoelichting
,'Sorry Geen toelichting aanwezig') || ')endarg,
  Toelichting op processtap,300)' from tbtermijnbewstappen where dnkey =
{id}
```

Het evalueren van de query levert een aanroep naar een action *startwizard(showtekst)* op , die vervolgens uitgevoerd wordt:

```
startWizard(showtekst,beginarg(ja ff documentje starten snap je
wel?)endarg,Toelichting op processtap,300)
```

Het gebruik van *beginarg(* en *)endarg* voorkomt dat komma's in de inhoud van *dvprocitemtoelichting* gezien worden als parameterscheiding.

Voorbeeld getflexaction en hoovertekst

Een tweede voorbeeld is de aanroep bij het hooverten over het groepicoontje op de lijst alle zaken (openingsportaal). Die kolom in de *MDLC_AlleZakenList.xml* is als volgt gedefinieerd.

```
<column tagnaam="dnkeygroepvergunning">
  <label>Groep</label>
  <index>10</index>
  <length>40</length>
  <wavetype>icoon</wavetype>
  <icoon>12</icoon>
  <showhint>false</showhint>
<hooverhint>getFlexAction(groep_inhoudtekst,{dnkeygroepvergunning})</hooverhint>
</column>
```

De tag *hooverhint* moet bij afspraak gevuld worden met de aanroep *getFlexAction(querynaam,{id-naam})* waarbij de query een string teruggeeft. Deze string wordt getoond bij het hooverten over de kolom in de lijst, waarbij puntkomma's in de string worden geïnterpreteerd als een harde return. Voorbeeld: het groepvergunningicoontje in de lijst alle zaken. De *hooverhint* is hier gedefinieerd als *getFlexAction(groep_inhoudtekst,{dnkeygroepvergunning})*. Dus de query met de naam *groep_inhoudtekst* wordt aangeroepen met de parameter *dnkeygroepvergunning* (uit *vwfrmalleaanvragen*) van de rij waar de gebruiker over het icoontje hoovert. De query retourneert startdatum, betreft en wavezaakcode van de leden van de groep, gescheiden met een puntkomma. Door het gebruik van *getFlexAction* wordt de query pas aangeroepen wanneer daar om gevraagd wordt.

Voorbeeld één getflexaction voor zowel getflexballoon als showtekst

Dit voorbeeld is dat achter een geflexballoon-knop en een startwizard(showtekst)-knop verwezen kan worden naar eenzelfde query: de actie achter de twee knoppen tonen dezelfde inhoud, alleen in een ander jasje. action bij knop showtekst: `getFlexAction(TestPaul_3_HaalMemoOp,S%keypointer%)` action bij knop flexballoon: `getFlexAction(TestPaul_3_HaalMemoOp,F%keypointer%)` De query: `TestPaul_3_HaalMemoOp` ontleedt de substitutie van {id} , dus begint met letter **S** of **F** en construeert de gewenste vervolg actieaanroep.

```
select
  case when substr('{id}',1,1) = 'F' then
    'getFlexBalloon(beginarg(' ||
coalesce(replace(dvmemo,chr(13)||chr(10),chr(59)),'Sorry. Geen memotekst
aanwezig')) || ')endarg,P)'
  else
    'startWizard(showtekst,beginarg(' || coalesce(dvmemo,'Sorry. Geen
memotekst aanwezig')) || ')endarg,Memo bij adres,300)'
  end
from tbomgvergunning where dnkey = substr('{id}',2)::integer
```

Voor getFlexBaloon worden de linefeeds en returns vervangen worden door een puntkomma.

getFlexMap

- Wat doet het?: In een venster wordt een kaart getoond op basis van gedefinieerde xml's met daarin informatie die de kaart moet tonen. Zie [Flexmap \(technische\) Beschrijving](#)
 - aanroep: `getFlexMap(param1, param2,param3,param4)`:
 - param1: De eerste parameter moet de een primary key zijn van een tabel die hoort bij de tweede parameter (tabelnaam). Dus het vertrekpunt van de kaart en ook het record waarin een evntuele wijziging wordt opgeslagen. Indien de kaart vanuit een detailpagina of lijst wordt ogeroepen kan verwezen worden naar de betreffende kolom in de onderliggende view of tabel door gebruik van accolades: {kolomnaam} bijvoorbeeld {dnkeymilinrichtingen} .
 - param2: De tweede parameter is de tabelnaam die hoort bij de param1.
 - param3 is de xmlnaam met daarin de instructie over wat op de kaart te zien is, en hoe de kaart gecentreerd moet worden bijv. `MDMC_tbmilinrichtingen.xml`. Strandaard levert OpenWave een groot aantal MDMC_xxx.xml's uit. Zie bovengenoemde link. Een functioneel beheerder kan een kaart aanpassen via de tabel tbscreencolumns of zelf kaarten definiëren.
 - param 4 = indien
 - leeg dan wordt de kaart gestart in kijkmodus.
 - Indien Polygon dan kan de gebruiker een polygoon tekenen op de kaart.
 - Indien Point dan kan de gebruiker een punt aanwijzen op de kaart
 - Indien LineString dan kan de gebruiker een lijn trekken op de kaart
- Voorbeeld:
`geefFlexMap({dnkeymilinrichtingen},tbmilinrichtingen,MDMC_tbmilinrichtingen.xml,Polygon)`

geefGeoVanLokatie DEPRECATED!!!!

- Wat doet het?: In een modal venster wordt een kaart getoond op basis van de coördinaatgegevens van een detailkaart. Zie [Kaart](#)
- aanroep: `geefGeoVanLokatie(param1, param2)`:
 - param1: De eerste parameter moet de een primary key zijn van een tabel die hoort bij de tweede parameter. Met uitzondering indien param2 = *AlgemeneKaart*. In dat laatste geval kan param1 een lege waarde hebben
 - param2: De tweede parameter is OF een tabelnaam OF de waarde *AlgemeneKaart*. De tabelnaam moet één van volgende zijn: *tbperceeladressen*, *tbmilinrichtingen*, *tbmildiversen*, *tbmilemlucht*, *tbmilemwater*, *tbmilopslag*, *tbharontheffingen*, *tbhandhavingen*, *tbovergunningen*, *tbomgvergunning*, *tbmilasbest*, *tbmilvergunningen*, *tbinfoaanvragen*, *tbhorecavergunningen*, *tbmilstal* of *tbzaakkadperc* of *tbmilafvalstoffen*
 - autorisatie: Niet van toepassing
- Voorbeeld: `geefGeoVanLokatie(%keypointer%,tbperceeladressen)` aangeroepen vanuit een knop op detailscherm.

getFlexMemo

- Wat doet het?: In een modal venster wordt een memo (de kolom *dvmemo*) getoond
- aanroep: `getFlexMemo(param1, param2)`:
 - param1: De eerste parameter moet de primary key zijn van de tabel die hoort bij de tweede parameter
 - param2: De tweede parameter is de letter C (*tbhorecavergunningen*), E (*tbmilvergunningen*), H (*tbhandhavingen*), I (*tbinfoaanvragen*) O (*tbovergunningen*) of W (*tbomgvergunning*) of V (*tbmilinrichtingen*)
- autorisatie: De gebruiker moet memokijkrechten hebben voor de betreffende module en de compartimentsrechten moeten kloppen
- Voorbeeld: `getFlexMemo(%keypointer%,W)` aangeroepen vanuit een knop op detailscherm.

getFlexBalloon

- Wat doet het?: In een hint-venster (ballontekst) behorende bij de knop waarmee deze methode wordt aangeroepen wordt een tekst getoond. **De tag refresh bij de knop moet de waarde false hebben (of leeg zijn)**
- aanroep: `getFlexBalloon(param1, param2)`:
 - param1: Een gecrypte tekst of een niet gecrypte tekst of een evalueerbare query. Afhankelijk van param2
 - param2:
 - D dan wordt de tekst in param1 in twee regels getoond in het ballonnetje. Eerste regel is param1 voorafgegaan door 'encrypt:'. De tweede regel is de gedecrypte versie van param1 voorafgegaan door 'decrypt'. (Zie [2-way encryptie van externe wachtwoorden](#))
 - P dan wordt param1 ongewijzigd in de tekstballoon getoond. Een semicolon (;) wordt daarbij geïnterpreteerd als harde return
 - QD dan bevat param1 een evalueerbare query die één regel en één kolom als resultaat teruggeeft, bijvoorbeeld: `select dvpas from tabelA where dnkey = %keypointer%`. De query moet beginnen met select en mag geen semicolons bevatten. De variabele `%keypointer%` wordt hierbij vervangen door de ID (dnkey) van de tabelkaart waar de gebruiker op staat. Nadat de query is geëvalueerd wordt het resultaat getoond als bij

- param1 = 'D'
 - QP dan bevat param1 een evalueerbare query die één regel en één kolom als resultaat teruggeeft, bijvoorbeeld: `select 'Let OP:' || dvnaam from tabelA where dnkey = %keypointer%`. De query moet beginnen met select en mag geen semicolons bevatten. De variabele %keypointer% wordt hierbij vervangen door de ID (dnkey) van de tabelkaart waar de gebruiker op staat. Nadat de query is geëvalueerd wordt het resultaat getoond als bij param1 = 'P'
- autorisatie: Niet van toepassing
- Voorbeeld: `getFlexBalloon(Hier alleen voorletters; zonder punt en spaties,P)`.

Voorbeeld ophalen memo-kolom in flexballoon

Indien een query wordt aangeroepen, waarbij de waarde van een kolom wordt opgehaald waarin linefeeds en harde returns kunnen zitten (memo), is het belangrijk dat de query deze linefeeds en returns vervangt door een puntkomma: aanroep:

`getFlexBalloon(%query(TestPaul_2_HaalMemoOp,%keypointer%)%,P)` de query `TestPaul_2_HaalMemoOp` kan dan bijvoorbeeld de volgende select bevatten:

```
select 'beginarg(' || coalesce(
replace(dvmemo,chr(13)||chr(10),chr(59)),'Sorry. Geen memotekst aanwezig')
|| ')endarg'
from tbomgvergunning where dnkey = {id}
```

Het gebruik van `beginarg(` en `)endarg` voorkomt dat komma's in de inhoud van dvmemo gezien worden als parameterscheiding.

noAction

- Wat doet het?: Niets
- aanroep: `noAction()`

refreshActiveDialog

- Wat doet het?: Het scherm waarvandaan deze action wordt aangeroepen wordt opnieuw uitgeschreven
- aanroep: `refreshActiveDialog()`
- autorisatie: OpenWave kijkt opnieuw naar de rechten van het te overschrijven scherm.

refreshActiveDialog(parameterlist)

- Wat doet het?: Overschrijft het flexDetail- of flexListscherm waar je op staat, maar dan op basis van parameterlist. Bijvoorbeeld: vanuit het flexdetailscherm van de tabel `tblegesregels` is een knop gedefinieerd met de action: `refreshActiveDialog(tblegesregels,13,W)`. De browser vraagt dan de detailgegevens op met `getFlexDetail(tblegesregels,13,W)` en overschrijft het bestaande detailscherm daarmee.

setcolumnvalue

- wat doet het? : met deze aanroep kan een waarde opgeslagen worden in een bepaalde kolom van een tabel in OpenWave, mits de gebruiker voldoende rechten heeft.
- Voorbeeld: `setcolumnvalue(tbmilinrichtingen,333,dnaantalbezoekers,100,V)`
- Parameterlist:
 - param1 is de tabelnaam waarin het gegeven moet worden opgeslagen
 - param2 is een verwijzing naar de primary key van die tabel om de juiste rij te vinden
 - param3 is de naam van de kolom waarin het gegeven moet worden opgeslagen
 - param4 is de waarde van het gegeven dat moet worden opgeslagen.
 - param5 is de moduleletter voor de rechtenafweging, indien:
 - leeg, dan moet:
 - OF de ingelogde medewerker beheerder zijn (`tbmedewerker.dnbeheerniveau > 98`)
 - OF - indien het gaat om `tbperceeladressen` - het wijzigrecht op `tbperceeladressen` aangevinkt zijn (`tbrechten.dldpcledt`)
 - OF de zesde parameter gevuld zijn
 - V dan moet de medewerker wijzigrechten hebben op de inrichtingen (`tbmilrechten.dlbmilinredt`)
 - W dan moet de medewerker wijzigrechten hebben op de omgevingzaken (`tbomgrechten.dlbomgedt`)
 - O dan moet de medewerker wijzigrechten hebben op de APV/overige zaken (`tbovrechten.dlbovvedt`)
 - H dan moet de medewerker wijzigrechten hebben op de handhavingszaken (`tbhhrechten.dlbhahedt`)
 - I dan moet de medewerker wijzigrechten hebben op de infozaken (`tbinforechten.dlbinfoedt`)
 - param6 mag leeg zijn. Indien gevuld gaat deze voor op de vijfde parameter en indien
 - de waarde begint met *tbrechten.* of *tbomgrechten.* of *tbovrechten.* of *tbinforechten.* of *tbhorrechten.* of *tbbestrechten.* of *tbmilvergrechten.* of *tbmilrechten.* gevolgd door een passende rechtenkolomnaam dan bepaald de evaluatie van die rechtenkolom of de gebruiker voldoende rechten heeft.
 - anders wordt de waarde opgezocht in `tbquery` (op kolom `tbquery.dvcode`) en dan bepaald de evaluatie van die query of de gebruiker voldoende rechten heeft

Waarschuwing

De OpenWave API `setcolumnvalue()` kan bij een aantal kolommen - naast de database constraints - nog extra controles uitvoeren op een bepaalde invoer EN kan zelf vervolgacties doen: zoals een (REV-) eindeGeldigheid op inrichtingsniveau doorvoeren naar onderliggende REV-objecten, of op grond van een bepaalde wijziging een updateZaakbericht (StUF Zaak/DMS) verzenden.

startWizard

- **startwizard(deleteContactadres,param2)**
 - Voorbeeld: `startwizard(deleteContactadres,333)`
 - param1: `deleteContactadres`
 - param2: De dnkeywaarde van de contactadreskaart die verwijderd moet worden. Indien param2 de waarde {id} bevat: de API wordt aangeroepen vanuit een lijst, dan wordt deze

string {id} on the fly door OpenWave vervangen met deze primary key-waarde van de actieve kaart uit een lijst.

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

- Voorbeeld:
startwizard(deleteSysStandardRow,tbadressoort.{id},dvomschrijving,beheer_tbadressoort)
- Aanroep van een standaard verwijderactie van een kaart van een tabel die gedefinieerd is in tbsysstandardtable (beheertegel *Tabellen Standaardapi*). Deze action kan bijv. aan een verwijderknop onder aan een lijst gekoppeld worden. De functie houdt rekening met de in de tbsysstandardbutton gedefinieerde rechten bij die knop en met het al of niet gevuld zijn van de in de tbsysstandardtable gedefinieerde blokkeringsvelden. De wizard geeft een waarschuwing indien de te verwijderen kaart komt uit tbomgvergunning of tbmilinrichtingen of tbhandhavingen of tbhorecavergunningen of tbovergunningen of tbinfoaanvragen of tbmilvergunningen m.b.t. fysieke documenten die niet mee verwijderd worden. De wavezaakcode van een verwijderde kaart uit een van deze tabellen kan opnieuw worden gebruikt
- param1: deleteSysStandardRow
- param2: De tabelnaam waaruit een kaart verwijderd moet worden gevolgd door een punt gevolgd door {id}. Die {id} wordt on the fly door OpenWave vervangen met primary key-waarde van de kaart die verwijderd moet worden (bijv. de actieve kaart uit een lijst)
- param3: een kolomnaam uit de view of tabel die aan de lijst ten grondslag ligt, waarvan de achterliggende waarde gebruikt wordt voor de *weet u het zeker* tekst
- param4: de code uit tbsysstandardtable die verwijst naar de kaart waar de betreffende standaardlijst in is gedefinieerd.

- **startwizard(insertContactadres)**

- Aanroep vanuit een situatie dat een adres moet worden aangemaakt zonder deze via een rol te koppelen aan een inrichting of zaak. De wizard vraagt om elementaire gegevens waarmee een nieuwe kaart wordt aangemaakt waarna vervolgens automatisch het detailscherm van de contactadreskaart wordt aangeroepen
getFlexDetail(tbcontactadressen,denieuweaangemaaktednkey).

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

- Voorbeeld:
startwizard(insertSysStandardRow,MDWC_insertTbMwTeams.xml,%keyparent%,beheer_tbmwteams)
- Aanroep van een standaard insertactie van een kaart van een tabel die gedefinieerd is in tbsysstandardtable (beheertegel *Tabellen Standaardapi*). Deze action kan bijv. aan een insertknop onder aan een lijst gekoppeld worden. De functie houdt rekening met de in de tbsysstandardbutton gedefinieerde rechten bij die knop en met het al of niet gevuld zijn van de in de tbsysstandardtable gedefinieerde blokkeringsvelden
- param1: insertSysStandardRow
- param2: De naam van de screen.xml waarin de opmaak van het insertscherm is geregeld. De naam moet beginnen 'MDWC_'. De xml moet aan een aantal voorwaarden voldoen. Zie: [Scherm informatie voor standaard insert- en kopieer](#)
- param3: Alleen gevuld indien de tabel waarop een insert wordt uitgevoerd een parenttabel heeft. Hier wordt de betreffende keywaarde van die parenttabel ingevuld. De waarde %keyparent% betekent dat OpenWave dit zelf onder water regelt
- param4: de code uit tbsysstandardtable die verwijst naar de kaart waar de betreffende tabel in is gedefinieerd.

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

- Voorbeeld:
startwizard(kopieerSysStandardRow,MDWC_insertTbMwTeams.xml,{id},beheer_tbmwteams)
- Aanroep van een standaard insertactie van een kaart van een tabel die gedefinieerd is in tbsysstandardtable (beheertegel *Tabellen Standaardapi*). Deze action kan bijv. aan een insertknop onder aan een lijst gekoppeld worden. De functie houdt rekening met de in de tbsysstandardbutton gedefinieerde rechten bij die knop en met het al of niet gevuld zijn van de in de tbsysstandardtable gedefinieerde blokkeringsvelden
- param1: kopieerSysStandardRow
- param2: De naam van de screen.xml waarin de opmaak van het insertscherm is geregeld. De naam moet beginnen 'MDWC_'. De xml moet aan een aantal voorwaarden voldoen. Zie: [Scherm informatie voor standaard insert- en kopieer](#). Kan in veel gevallen dus gelijk zijn aan het scherm dat hoort bij de insertStandardRow
- param3: Wordt gevuld met de dnkey van de rij waar je op staat. Indien de kopieerknop onderaan een lijst staat kan {id} worden gebruikt, en anders, op een detailscherm, %keypointer%
- param4: de code uit tbsysstandardtable die verwijst naar de kaart waar de betreffende tabel in is gedefinieerd.
- **startwizard(koppelSysStandardRow,param2,param3,param4)**
 - Voorbeeld: startwizard(koppelSysStandardRow,%keyparent%;dvomschrijving;;0,tbdocumentsoorten;dnkeydocsoort;dnkey;;dvomschrijving;;ddvervaldatum;1=1,beheer_vwfrmkopdocaandocsoort)
 - Aanroep van een meerdere inserts tegelijk op een koppeltabel die gedefinieerd is in tbsysstandardtable (beheertegel *Tabellen Standaardapi*) als dochtertabel (dus voorzien van een parentkey). Deze action kan bijv. aan een insertknop onder aan een lijst gekoppeld worden. De gebruiker kan uit een derde tabel rijen aanvinken op grond waarvan nieuwe kaarten worden toegevoegd en/of verwijderd in de standaardtabel. De functie houdt rekening met de in de tbsysstandardbutton gedefinieerde rechten bij die knop en met het al of niet gevuld zijn van de in de tbsysstandardtable gedefinieerde blokkeringsvelden. Zie ook knoppen bij [Standaard Lijst- en Detailschermen](#)
 - param1: koppelSysStandardRow
 - param2: Bestaat uit vier delen gescheiden door een puntkomma.
 - 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).
 - 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 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 keuzelijstscherm.
- 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.
- **startWizard(selecteerTaak,param2)** waarbij een wizard wordt gestart om een selectie te maken van openstaande taken op medewerker(s), modules en taaksoorten. De waarde 1 bij param2 geeft aan dat de zoekwizard blijft staan.
- **startWizard(maakDocument,param2,param3,param4)** waarbij een wizard wordt gestart teneinde een documentsjabloon aan te wijzen. Zie uitgewerkte voorbeelden onder kopje action bij [Termijnstappen](#).
- **startWizard(maakEmail,param2,param3,param4)** Idem als documentsjabloon, maar dan voor e-mails.
- **startWizard(maaknieuwelnrichting,param2)** waarbij een wizard wordt gestart teneinde een nieuwe inrichting te definiëren. Param2 kan leeg zijn. In dat geval wordt de wizard geopend, waarbij de gebruiker eerst gemeente, woonplaats en straat moet kiezen. Als param2 gevuld is verwacht OpenWave dat dit een dnkey uit de tabel TbOpenBareRuimte is. De inlogger zal dan alleen het adres binnen die straat moeten kiezen.
- **startWizard(maaknieuwproces,param2,param3)** waarbij een wizard wordt gestart teneinde een vervolgproces te kiezen vanuit de procesbewaking. Zie uitgewerkte voorbeelden onder kopje action bij [Termijnstappen](#).
- **startWizard(maaknieuwezaak,param2,param3,param4)** waarbij een wizard wordt gestart teneinde een nieuwe hoofdzaak te definiëren. Zie uitgewerkte voorbeelden onder kopje action bij [Termijnstappen](#) EN bij lemma [Aanmaken van nieuwe zaak](#).
- **startwizard(showTekst,param2,param3,param4)** waarbij een wizard wordt gestart van één scherm met alleen een sluitknop die de tekst uit param2 toont. Indien de tekst wordt opgehaald uit een kolom die ook linefeeds en harde returns kan bevatten (zoals een memo) moet de aanroep via getflexaction verlopen: zie voorbeeld hierboven bij getFlexAction.
 - Voorbeeld: `startWizard(showTekst, dit is een tekst,dit is de koptekst,400)`
 - param1: showTekst
 - param2: de tekst die getoond wordt in het wizardscherm. Mag een lange tekst zijn
 - param3: de koptekst. Mag leeg zijn
 - param4: hoogte van tekstvak in pixels. Indien leeg dan is de default 120.
- **startwizard(sluitZaak,param2,param3,param4)** waarbij een wizard wordt gestart teneinde een hoofdzaak af te sluiten. Zie uitgewerkte voorbeelden onder kopje action bij [Termijnstappen](#) en het lemma [Sluiten van zaak](#).
- **startwizard(startreport,param2,param3)** waarbij een bepaald rapport wordt gestart (`tbrapporten.dnkey = param2`). Param3 mag een lege waarde hebben, maar indien gevuld dan moet het rapport aangeroepen worden vanuit een zaakportaal, waarbij param3 de id is van die hoofdzaak (dus bijv. een dnkey uit tbomgvergunning). Zie voorbeeld voor het gebruik van deze param3 identifier van zaakportal (`nportalid`) in [Rapportages](#).
- **startWizard(zoekInrichtingopNaam,param2)** waarbij een wizard wordt gestart teneinde een inrichting te zoeken. De waarde 1 bij param2 geeft aan dat de zoekwizard blijft staan. Bij 0

wordt de wizard gesloten.

- **startWizard(ZoekZaakViaZaaknummer,param2)** waarbij een wizard wordt gestart teneinde een zaak te zoeken op zaakcodering. De waarde 1 bij param2 geeft aan dat de zoekwizard blijft staan.
- **startWizard(ZoekZaakViaAdres,param2)** waarbij een wizard wordt gestart teneinde een zaak te zoeken op adres. De waarde 1 bij param2 geeft aan dat de zoekwizard blijft staan.
- **startWizard(ZoekZaakViaBetreftDatum,param2)** waarbij een wizard wordt gestart teneinde een zaak te zoeken op omschrijving of datums. De waarde 1 bij param2 geeft aan dat de zoekwizard blijft staan.
- **startWizard(ZoekZaakViaContact,param2)** waarbij een wizard wordt gestart om een zaak te zoeken op contactpersoon. De waarde 1 bij param2 geeft aan dat de zoekwizard blijft staan.
- **startWizard(ZoekInspectieViaZaaknummer,param2)** waarbij een zoekwizard wordt gestart om een inspectiezaak te zoeken op een zaakcodering. De waarde 1 bij param2 geeft aan dat de zoekwizard blijft staan.
- **startWizard(StuurDSOOntvangstbevestiging,param2,param3,param4)** waarbij een wizard wordt gestart om een DSO ontvangstbevestigingsmail te versturen.
 - param1: StuurDSOOntvangstbevestiging
 - param2: Primary key van tabel genoemd bij param3. Dnkey van tbomgvergunning of tbomgdsoaanvullintrek
 - param3: Naam van de tabel waarvoor na genereren de verstuurdatum gevuld moet worden. Moet gevuld zijn en of waarde *tbomgvergunning* (voor DSO initieel) of *tbomgdsoaanvullintrek* (voor DSO aanvulling) hebben
 - param4: optioneel, indien gevuld dan dnkey van processtap (tbtermijnbewstappen) die moet worden afgesloten. Let op als param4 gevuld dan mag param3 alleen waarde *tbomgvergunning* hebben: DSO ontvangstbevestiging Aanvulling versturen vanaf processtap kan niet. Er kunnen immers meer dan 1 aanvullingen zijn, dnkey is niet bekend bij termijnbewakingsstappen. Zie uitgewerkte voorbeelden onder kopje action bij [Termijnstappen](#).
- **startWizard(insertInspectieTraject,param2,param3,param4)** waarbij een wizard wordt gestart die een inspectietraject aanmaakt. Deze wizard mag ook aangeroepen worden vanuit een termijnstap.
 - param1: insertInspectieTraject
 - param2: Moet de dnkey van de bovenliggende zaak zijn (bijvoorbeeld %keyparent%).
 - param3: Moet gevuld zijn met de module letter, deze mag niet leeg zijn.
 - param4: Dit mag de dnkey van een termijnstap zijn(%keypointer%), deze mag leeg zijn.
- **startWizard(Insertinbehandelingbij,param2,param3,param4)** waarbij een wizard wordt gestart die een behandelaar bij de aangegeven zaak aanmaakt. Deze wizard mag ook aangeroepen worden vanuit een termijnstap.
 - param1: Insertinbehandelingbij
 - param2: Moet de dnkey van de bovenliggende zaak zijn (bijvoorbeeld %keyparent%).
 - param3: Moet gevuld zijn met de module letter, deze mag niet leeg zijn.
 - param4: Dit mag de dnkey van een termijnstap zijn (bijvoorbeeld %keypointer%), deze parameter mag leeg zijn.
- **startWizard(InsertLegesRegel,param2,param3,param4)** waarbij een wizard wordt gestart die een nieuwe legesregel bij de aangegeven zaak aanmaakt. Deze wizard mag ook aangeroepen worden vanuit een termijnstap.
 - param1: InsertLegesRegel
 - param2: Moet de dnkey van de bovenliggende zaak zijn (bijvoorbeeld %keyparent%).
 - param3: Moet gevuld zijn met de module letter, deze mag niet leeg zijn EN moet de letter **W**(abo/omgeving), **O**(overig/diversen/APV), **C**(horeca) of **I**(nfo-aanvraag) zijn.
 - param4: Dit mag de dnkey van een termijnstap zijn (bijvoorbeeld %keypointer%), deze

parameter mag leeg zijn.

- **startWizard(insertZaakProduct,param2,param3,param4)** waarbij een wizard wordt gestart die een product bij de aangegeven zaak aanmaakt. Deze wizard mag ook aangeroepen worden vanuit een termijnstap.
 - param1: insertZaakProduct
 - param2: Moet de dnkey van de bovenliggende zaak zijn (bijvoorbeeld %keyparent%).
 - param3: Moet gevuld zijn met de module letter, deze mag niet leeg zijn.
 - param4: Dit mag de dnkey van een termijnstap zijn (bijvoorbeeld %keypointer%), deze parameter mag leeg zijn.
- **startWizard(insertSWFruimte,param2,W,param4)** waarbij een wizard wordt gestart die een nieuwe Samenwerkingsruimte (SWF) bij de aangegeven zaak aanmaakt. Deze wizard mag ook aangeroepen worden vanuit een termijnstap.
 - param1: insertSWFruimte
 - param2: Moet de dnkey van de bovenliggende zaak zijn (bijvoorbeeld %keyparent%).
 - param3: Deze parameter moet altijd gevuld zijn met de moduleletter van de omgevingsmodule (**W**).
 - param4: Dit mag de dnkey van een termijnstap zijn (bijvoorbeeld %keypointer%), deze parameter mag leeg zijn.
- **startWizard(wijzigBevoegdGezagDSO,param2,param3)** waarbij een wizard wordt gestart die het bevoegdgezag of de behandeldienst bij de aangegeven zaak wijzigt. Deze wizard mag ook aangeroepen worden vanuit een termijnstap.
 - param1: wijzigBevoegdGezagDSO
 - param2: Moet de dnkey van de bovenliggende zaak zijn (bijvoorbeeld %keyparent%).
 - param3: Dit mag de dnkey van een termijnstap zijn (bijvoorbeeld %keypointer%), deze parameter mag leeg zijn.
- **startWizard(insertToestemming,param2,param3,param4)** waarbij een wizard wordt gestart die een nieuw onderdeel/activiteit bij de aangegeven zaak toevoegt. Deze wizard mag ook aangeroepen worden vanuit een termijnstap.
 - param1: insertToestemming
 - param2: Moet de dnkey van de bovenliggende zaak zijn (bijvoorbeeld %keyparent%).
 - param3: Deze parameter moet altijd gevuld zijn met een moduleletter.
 - param4: Dit mag de dnkey van een termijnstap zijn (bijvoorbeeld %keypointer%), deze parameter mag leeg zijn.
- **startWizard(koppelzaakaaninrichting,param2,param3,param4)** waarbij een wizard wordt gestart die de bovenliggende zaak koppelt aan een inrichting. Deze wizard mag ook aangeroepen worden vanuit een termijnstap.
 - param1: koppelzaakaaninrichting
 - param2: Moet de dnkey van de bovenliggende zaak zijn (bijvoorbeeld %keyparent%).
 - param3: Deze parameter moet altijd gevuld zijn met een moduleletter.
 - param4: Dit mag de dnkey van een termijnstap zijn (bijvoorbeeld %keypointer%), deze parameter mag leeg zijn.
- **startWizard(StuurMailAanBAGBeheerder,param2,param3a,param3b,param4)** waarbij een wizard wordt gestart die de bovenliggende zaak koppelt aan een inrichting. Deze wizard mag ook aangeroepen worden vanuit een termijnstap.
 - param1: StuurMailAanBAGBeheerder
 - param2: Moet de dnkey van de bovenliggende zaak zijn (bijvoorbeeld %keyparent%).
 - param3a: Deze deel-parameter moet altijd gevuld zijn met een tabelnaam van de gewenste module-basistabel(bijvoorbeeld *tbomgvergunning*).
 - param3b: Deze deel-parameter mag leeg zijn of moet een punt-komma(;) volgen en

- gevuld zijn met de dnkey van een termijnstap (bijvoorbeeld %keypointer%).
 - param4: Deze parameter moet verwijzen naar het veld in de tabel dat verwijst naar de reden voor het sturen van de mail, bijvoorbeeld *ddinitbag* bij een omgevingsvergunning, zie het lemma [Email naar BAG beheerder](#) voor de overige mogelijkheden.
- **startWizard(PubliceerZaken,param2,param3,param4)** waarbij een wizard wordt gestart die OF de bovenliggende zaak OF alle zaken die klaar zijn gezet publiceert. Deze wizard mag ook aangeroepen worden vanuit een termijnstap.
 - param1: PubliceerZaken
 - param2: Mag leeg zijn (dan worden alle nog niet gepubliceerde zaken in *vwfrmzakentepubliceren* gepubliceert) maar indien gevuld dan moet het overeenkomen met de inhoud van een veld in de kolom *dvid* van de view *vwfrmzakentepubliceren*. Het veld *dvid* is een samenstelling van een moduleletter en een dnkey (bijvoorbeeld *W31334*).
 - param3: Deze parameter wordt momenteel (nog) niet gebruikt en moet leeg zijn.
 - param4: Dit mag de dnkey van een termijnstap zijn (bijvoorbeeld %keypointer%), deze parameter mag leeg zijn.
- **startwizard(InsertRolContact,param2,param3a;param3b)** waarbij een wizard wordt gestart die een contactpersoon, en diens rol, toevoegd aan de bovenliggende zaak. Deze wizard mag ook aangeroepen worden vanuit een termijnstap.
 - param1: InsertRolContact, dit is de naam van de wizard
 - param2: Deze parameter moet gevuld zijn met de dnkey van de hoofdtabel die hoort bij de in param3 aangegeven moduleletter (bijvoorbeeld %keyparent%).
 - param3a: Deze deel-parameter moet tenminste gevuld zijn met de module letter, deze mag niet leeg zijn EN moet de letter **W**(abo/omgeving), **O**(overig/diversen/APV), **C**(horeca), **V** (inrichtingen), **E**(milieuvergunning) of **I**(nfo-aanvraag) zijn.
 - param3b: Deze deel-parameter mag leeg zijn of moet een punt-komma(;) volgen met de dnkey van de termijnstap (bijvoorbeeld %keypointer%).
- **startwizard(haalGeregistreerdDocument,param2,param3,param4)** waarbij een geregistreerd document wordt opgehaald en geopend
 - param1: *haalGeregistreerdDocument*, dit is de naam van de wizard (een wizard zonder schermen)
 - param2: een dnkey van tbcorrespondentie (dus van een geregistreerd document)
 - param3: de module letter van de hoofdzaak of inrichtingen waar het document toe behoort: dus V, B, C, E, I, O, H, of W
 - param4 is een samengestelde parameter bestaande uit drie delen gescheiden door puntkomma's waarbij het eerste deel de 4-cijferige gemeentecode is van de hoofdzaak of inrichting waarbij het document is opgeslagen, het tweede deel een keyverwijzing naar een compartiment (mag 0 zijn of leeg) en het derde deel is *false* of *true* en geeft aan of het document geblokkeerd is tegen schrijven. bijv. *0187;0:false*

Action column:kolomnaam

Deze actie heeft dezelfde functie als getFlexAction. Alleen wordt de uiteindelijke uit te voeren actie NIET opgehaald uit een query, maar uit een kolom van de view die aan de lijst ten grondslag ligt.

De tag <action> bij de knop wordt zonder verdere parameters gevuld wordt door de vaste tekst 'column:' die direct gevolgd wordt door een kolomnaam uit de betreffende lijst. De waarde van die kolomnaam bevat de action die uitgevoerd wordt bij het indrukken van de bewuste knop. Het gaat dan om de waarde uit de actieve regel van die lijst. In zo'n kolom staat dus bijv. als waarde: *geefGeovanLokatie(1234,tbperceeladressen)* en in diezelfde lijst bij een volgende regel

geefGeovanLokatie(5678,tbperceeladressen).

Toepassing: zie beheertegel *Tabellen Standaardapi* en zoek de kaart met dvcode = *opening_vwfrmtevolgenzaken*.

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

Permanent link:
https://acc-doc.open-wave.nl/dev/applicatiebeheer/instellen_inrichten/actions?rev=1784103605

Last update: **2026/07/15 11:07**

