

# FORM SCRIPTING

## USER GUIDE

### Document Version Control

Version	Date	Authors	Description / Impact to Document
1.2	16/06/2015	Peter Davis	New document
2.0	08/05/2018	Peter Burns	Simplify installation
2.1	14/05/2018	Peter Burns	Add suppression of validation on table fill
2.2	12/10/2018	Peter Burns	Remove Master file scripting.
2.3	17/10/18	Peter Burns	Extend examples
2.4	07/12/18	Peter Burns	Developer help
2.5	07/01/19	Peter Burns	Validate script, Safe SQL, section actions, debug changes, ONLOADEDVAL event.
2.6	13/02/19	Peter Burns	Use of * in FILL-TABLE target field.
2.7	02/07/19	Peter Burns	Enable turn off validation on FILL-COLUMN. Update Radio Button example and calculated field examples.
700.6	11/07/2019	Peter Burns	Add Expressions
700.6.1	11/07/2019	Peter Burns	Button example using Expressions
700.6.8	13/12/2020	Peter Burns	Review examples and make validation flag consistent across all scripting.

## TABLE OF CONTENTS

<b>1. INTRODUCTION</b>	<b>4</b>
<b>2. Flexi-Field Group Configuration for script</b>	<b>5</b>
2.1. FILLCOLUMNS	5
2.2. FILLLISTS	6
2.3. FILLPARAMS	6
2.4. FILLEXPRESS	7
2.5. Link Flexi Field Group to an Attribute Master file	8
<b>3. Forms Auto-Population Usage</b>	<b>9</b>
3.1. Fill Fields	9
3.2. Fill Lists	11
3.3. Fill Parameters	13
3.4. Fill Expressions	14
3.5. Example Form	16
3.5.1. Example: FILL-LIST ONLOADED	17
3.5.2. Example: FILL-LIST ONVALIDATED	18
3.5.3. Example: FILL-COLUMN ONLOADEDVAL	19
3.5.4. Example: FILL_LIST in Table section	20
3.5.5. Example: FILL-LIST on text field (drop-down list)	21
3.5.6. Example: READ-ONLY	21
3.5.7. Example: READ-ONLY in table row	22
3.5.8. Example: HIDE field	22
3.5.9. Example: SECTION ACTIONS - HIDE & READ-ONLY	23
3.5.10. Example: FILL-TABLE	24
3.5.11. Example: Calculated Fields	25
3.5.12. Example: Radio Buttons with Expressions	26
3.5.13. Example: Aggregate Expression	27
3.6. Flexi-group setup for examples	28
<b>4. Enhanced DIA</b>	<b>29</b>
4.1. Disable this custom functionality	30
<b>5. Form UserID Update</b>	<b>31</b>
5.1. Disable this custom functionality	32
<b>6. Validating script definition</b>	<b>33</b>
6.1. Disable Validation of script when entering in the web.	33
<b>7. SAFE SQL</b>	<b>34</b>
7.1. Disable safe SQL	34

## 8. Debugging

35

## 1. INTRODUCTION

This is a custom scripting tool for Flexi-groups.

There is a lot of functionality built into flexi-groups, such as validation and data control. From Milestone 7 there is some functionality to dynamically control forms, default data and update objects in workflow. ***Standard functionality should be used in preference to this custom scripting tool.***

Standard functionality and custom scripting tools will work together but take care! The standard functionality is a little restrictive, but this makes sure that you can't do any damage. The scripting tool is more flexible, but it means that it is more complex to use and does give you access to areas where you could break the usability of a form.

The configuration data for form scripting is stored in four Flexi field groups. These groups are attached to a Master file attribute to provide a data entry mechanism. The Master file attribute and the attribute values used or not important, scripting is driven by the content of the four Flexi-groups when they reference a form attribute ID.

In a form, each flexi-field group becomes a form section. In this document the word, section, is used to refer to a flexi-field group as it appears on a form.

## 2. FLEXI-FIELD GROUP CONFIGURATION FOR SCRIPT

The required flexi-groups are distributed in the installation package as XML files. Those files can be imported via the Desktop client system admin screen; **Settings ► System Administration ► System setup ► Flexi-field group definition ► Flexi-field group definition**

Select **'File' > 'Open...'**, then, in turn, each XML file included in the '02 Flexi Field' directory of the installation package. Then **Save** the screen (tabbing through the fields may cause validation errors and should be avoided). Because of the old column names (no '\_fx') entering them manually is not going to be possible.

### 2.1. FILLCOLUMNS

Import file: fillcolumns.xml

Definition:

Flow-field group	FILLCOLUMNS	Description	Fill Field	
Type	Flexi-field group	Table name	afillcolumns	
View format	Table	Status	Active	
<input type="checkbox"/> Add date interval with validation	Search criteria	Show no search criteria		

	Title	Multilingual title	Column name	Data type	Attribute	Related attribute	Display format	Field width	Data length	Mand	Hide	Sum	Read	Val	Def	Value lookup	ID
1	Form attribute id	<input type="checkbox"/>	fill_attribute_id	Attribute	ATTNAME		Code	12	25	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		11
2	Trigger event	<input type="checkbox"/>	trigger_event	Text			Text any case	20	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0
3	Trigger group	<input type="checkbox"/>	trigger_flexi_gr_id	Attribute	FLEXIGRP		Code	25	12	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		11
4	Trigger field	<input type="checkbox"/>	trigger_field	Text			Text lower case	20	40	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0
5	Action	<input type="checkbox"/>	action_fx	Text			Text any case	20	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0
6	Target group	<input type="checkbox"/>	fill_flexi_gr_id	Attribute	FLEXIGRP		Code	12	12	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		11
7	Target field	<input type="checkbox"/>	fill_col_name	Text			Text lower case	20	30	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0
8	SQL	<input type="checkbox"/>	fill_command	Text			Text any case	1000	1000	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0
9	#	<input type="checkbox"/>	fill_seq_no	Integer			Small	3	3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0
10	Keep	<input type="checkbox"/>	keep_existing_value	Check box				1	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0
11	Append	<input type="checkbox"/>	append_to_table	Check box				1	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0
12	Validate target	<input type="checkbox"/>	call_validate	Check box				1	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0
13	ASQL	<input type="checkbox"/>	asql_flag	Check box				1	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0
14	Comment	<input type="checkbox"/>	comment_fx	Text			Text any case	50	1000	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0

Validations:

Field name: **Action**      Data type: **Text**

Range restrictions

Operator:       Numeric:

Period type:       Date:

Period:

User-defined values

ONLOADED  
ONVALIDATED  
ONLOADVAL

Attribute value restrictions

Attribute:

Possible values

Field name: **Action**      Data type: **Text**

Range restrictions

Operator:       Numeric:

Period type:       Date:

Period:

User-defined values

FILL-COLUMN  
FILL-TABLE  
HIDE  
READ-ONLY

Attribute value restrictions

Attribute:

Possible values

Chosen values

## 2.2. FILLLISTS

Import file: filllists.xml

Definition:

Flex-field group: FILLLISTS Description:

Type: Flex-field group Table name:

View format: Table Status: Active

☐ Click to minimise ☐ With validation Search criteria:

	Title	Multilingual	Column name	Data	Attribute	Related	Display	Field	Data	Mand	Hide	Sum	Read	Val	Def	Value	lookup	ID
1	Form attribute ID	<input type="checkbox"/>	fill_attribute_id	Attribute	ATTNAME		Code	12	25	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	11
2	Trigger event	<input type="checkbox"/>	trigger_event	Text			Text any case	12	12	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0
3	Trigger group	<input type="checkbox"/>	trigger_flex_gr_id	Attribute	FLEXIGRP		Code	25	12	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	11
4	Trigger field	<input type="checkbox"/>	trigger_field	Text			Text lower case	20	40	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0
5	Fill group	<input type="checkbox"/>	fill_flex_gr_id	Attribute	FLEXIGRP		Code	12	12	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	11
6	Fill field	<input type="checkbox"/>	fill_col_name	Text			Text lower case	20	30	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0
7	Fill query	<input type="checkbox"/>	fill_command	Text			Text any case	1000	1000	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0
8	#	<input type="checkbox"/>	fill_seq_no	Integer			Small	3	3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0
9	Type ahead	<input type="checkbox"/>	type_ahead	Check box				1	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0
10	Default first	<input type="checkbox"/>	def_first_val	Check box				1	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0
11	Default single	<input type="checkbox"/>	def_single_val	Check box				1	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0
12	ASQL	<input type="checkbox"/>	asql_flag	Check box				1	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0
13	Comment	<input type="checkbox"/>	comment_fx	Text			Text any case	1000	1000	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0
14	Not Used	<input type="checkbox"/>	debug	Check box				1	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0

## 2.3. FILLPARAMS

Import file: fillparams.xml

Definition:

Flex-field group: FILLPARAMS Description:

Type: Flex-field group Table name:

View format: Table Status: Active

☐ Add date interval with validation Search criteria:

	Title	Multilingual	Column name	Data	Attribute	Related	Display	Field	Data	Mand	Hide	Sum	Read	Val	Def	Value	lookup	ID
1	Form attribute ID	<input type="checkbox"/>	fill_attribute_id	Attribute	ATTNAME		Code	12	25	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	11
2	Parameter	<input type="checkbox"/>	fill_parameter_id	Text			Text lower case	12	30	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0
3	Source group	<input type="checkbox"/>	from_flex_gr_id	Attribute	FLEXIGRP		Code	12	12	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	11
4	Source field	<input type="checkbox"/>	from_col_name	Text			Text lower case	12	30	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0
5	Target group	<input type="checkbox"/>	fill_flex_gr_id	Attribute	FLEXIGRP		Code	12	12	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	11
6	Target field	<input type="checkbox"/>	fill_col_name	Text			Text lower case	12	30	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0
7	not used	<input type="checkbox"/>	from_line_no	Integer			Small	5	5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0

## 2.4. FILLEXPRESS

Import file: fillexpress.xml

Definition:

Flex-field group: **FILLEXPRESS** Description: **Fill Field with local script**

Type: **Flex-field group** Table name: **afvfillexpress**

View format: **Table** Status: **Active**

☐ Add date interval with validation Search criteria: **Show no search criteria**

	Title	Multilingual title	Column name	Data type	Attribute	Related attribute	Display format	Field width	Data length	Mand	Hide	Sum	Read	Vai	Def	Value lookup	ID
1	Form attribute id		fill_attribute_id	Attribute	ATTNAME		Code	12	25	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		11
2	Trigger event		trigger_event	Text			Text any case	20	20	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		0
3	Trigger group		trigger_flexi_gr_id	Attribute	FLEXGRP		Code	25	12	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		11
4	Trigger field		trigger_field	Text			Text lower case	20	40	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0
5	Action		action_fx	Text			Text any case	20	20	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		0
6	Target group		fill_flexi_gr_id	Attribute	FLEXGRP		Code	12	12	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		11
7	Target field		fill_col_name	Text			Text lower case	20	30	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0
8	Source group		source_group_fx	Attribute	FLEXGRP		Code	25	12	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		11
9	Expression		fill_command	Text			Text any case	1000	1000	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0
10	#		fill_seq_no	Integer			Small	3	3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0
11	Keep		keep_existing_value	Check box				1	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0
12	Append		append_to_table	Check box				1	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0
13	Validate target		call_validate	Check box				1	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0
14	Comment		comment_fx	Text			Text any case	50	1000	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0

Default value

Field name: **Action**

Data type: **Text**

Mask: ☐

Default value: **EXPRESSION**

## 2.5. Link Flexi Field Group to an Attribute Master file

The above flexi field groups should be linked to a new **attribute master file**. It does not matter what ID you use for the attribute master file as the flexi tables created above will be queried directly, however an example setup is shown below:

Attribute <input type="text" value="#FAUTO"/> auto fill												
Table name <input type="text" value="actitemvalue"/>												
	Flexi-field group	Description	Data table	Tab	Tab name	Multilingual title	Section	Section name	Multilingual title	Mandatory	Connect to rule	S
1	FILLCOLUMNS	Fill field	actfillcolumns	1	SQL Fill fields	<input type="checkbox"/>		1 Fill fields SQL	<input type="checkbox"/>	<input type="checkbox"/>		N
2	FILLISTS	Fill lists	actfilllists	2	SQL Fill lists	<input type="checkbox"/>		1 Fill lists SQL	<input type="checkbox"/>	<input type="checkbox"/>		N
3	FILLPARAMS	Fill parameters	actfillparams	3	SQL Fill parameters	<input type="checkbox"/>		1 Fill parameters SQL	<input type="checkbox"/>	<input type="checkbox"/>		N
4	FILLEXPRESS	Fill field with local ratfillsexpress		4	Fill Fields (Expressions)	<input type="checkbox"/>		1 Fill fields expressions	<input type="checkbox"/>	<input type="checkbox"/>		N

When creating new attribute values for this master file it is possible to hold all your configuration against a single attribute value or, to make things easier to maintain with large configurations, you can hold individual form configurations against different attribute values.

You can spread your forms scripting across as many, or as few, different attribute values as you like.

**Script on Parked or Closed attribute values will not be run**, so splitting up your script in some way (by section perhaps) can be advantageous.

EN Attribute master file x

New link Organise links

Text Show workflow map Company

1 - Attribute 2 - Address 3 - Relation 4 - Memo 5 - SQL Fill fields 6 - SQL Fill lists 7 - SQL Fill parameters 8 - Fill Fields (Expressions)

Attribute FFAUTO Attr.value NEW

Description

Period from 0 Period to 209999

Owner Value

Status

Updated User



### 3. FORMS AUTO-POPULATION USAGE

The form population part of this customisation allows you to automatically populate fields within a form based on the values in other fields and SQL look-ups against the database. You should be able to configure the software to populate any field and make changes in the future without the need for any changes to the software.

#### 3.1. Fill Fields

Form attribute id	Trigger event	Trigger group	Trigger field	Action	Target group	Target field	SQL	#	Keep	Append	Validate target	ASQL	Comment
1								0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
2									<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

This table specifies the fields and table sections to be auto-populated and the SQL lookup that is used to look up the value(s).

#### Form attribute ID

The attribute ID of the form that the flexi group above is attached to.

#### Trigger event

The event that will trigger the action.

- ONLOADED means the script action will be run when the form loads, after the Form Id has been entered.
- ONVALIDATED means that the script action will be run when the **trigger field** has been changed and validated.
- ONLOADVAL the script will be run both ONLOADED and ONVALIDATED.

#### Trigger group

The Flexi-group ID that contains the **trigger field**.

#### Trigger field

The field which triggers an ONVALIDATED script action.

#### Action

This is the scripting action that will be used, it should be set to.

- 'FILL-COLUMN' to populate a single field. ('AUTO-FILL' is a deprecated name for the same action.)
- 'FILL-TABLE' to populate a table section.
- 'HIDE' to show/hide a field. This will work on table sections removing a whole column!
- 'READ-ONLY' to enable/disable data entry to a field.

## Target group

The flexi-field group to be populated, or that contains the field to be acted on.

## Target field

The column name to be acted on.

This column does not need to be populated when the 'FILL-TABLE' action has been chosen.

This column can be set to an asterisk (\*) if the action is for a whole section.

## SQL

This is the SQL command that is used to look-up a value to populate the 'Fill Column Name' or table section specified above and will be run against the ABW database. When populating a single field, the SQL must return a single row with one field and when populating a table section, the SQL can return any number of rows and columns, however, only columns with names matching a column name in the table section will be populated, if you have any columns in the table section that do not get returned by the SQL these will be set to their default values. You can add parameters within the SQL statement, these should start with a \$ or @ symbol. These parameters are used in conjunction with the Fill Parameters flexi group that is detailed below. The current client can be specified by using \$client, which is treated as a reserved parameter, so client should not be specified in the Fill Parameters flexi group.

For the HIDE and READ-ONLY actions the query should return a row if the action is true and no row for false. For example, the query `SELECT 1 WHERE @value > 100` could hide a field if the field passed in as @value was over 100. See the examples below for more detail.

## #

Should usually be 0, however it also provides a way to chain together script actions by creating multiple lines against the same 'Trigger Field', each with a different sequence number, these will then be run in the order specified.

## Keep

If this field is checked then the 'Fill Column Name' specified above will only be populated if it is currently empty.

When populating a table section, if this is checked, then any rows already in the table will be kept, if it is un-checked the table section will be cleared first before new rows are added.

## Append

Only applicable for 'FILL-TABLE' actions. If this is checked and the table you are populating already contains rows then any rows returned by the SQL will be appended to the table. No duplicate checking is performed.

## Validate target

This applies to the FILL-TABLE and FILL-COLUMN action.

When this box is checked, the value of every column, on each row added, will be fully validated against the flexi-group definition. Any ONVALIDATED script triggered by the target column will also be run. This has the potential to seriously slow down filling a table.

Leaving this blank will stop validation of added data as it is inserted by the FILL-TABLE and FILL-COLUMN actions.

## ASQL

Should be checked if the SQL you are providing used ASQL (Business world SQL) syntax.

Leave it unchecked if you are using native (Microsoft T-SQL or Oracle) syntax.

## Comment (optional)

For documentation. This column may be hidden to give more space to other columns on screen.

## 3.2. Fill Lists

This table specifies the fields that you wish to turn into drop-down/type-ahead lists containing specific values. Any flexi-field that you want this functionality to run against should be set to be a standard 'Text' field via the flexi-field group definition screen, it will then be changed to a dropdown list at run-time.

Form attribute ID	Trigger event	Trigger group	Trigger field	Fill group	Fill field	Fill query	#	Type ahead	Default first	Default single	ASQL
1							0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2								<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Explanation of fields:

### Form attribute ID

The attribute ID of the form that the flexi group above is attached to.

### Trigger event

The event that will trigger the creation of a list of values, this can be set to either ONLOADED or ONVALIDATED.

ONLOADED means the SQL will be run and a list will be created **only** when the form loads.

ONVALIDATED means the SQL will be run and a list will be created when the form loads **and** when the trigger field is amended.

### Trigger group

The Flexi-group ID that contains the **trigger field**.

### Trigger field

The field which triggers an ONVALIDATED script action.

### Fill group

The flexi group to be populated.

### Fill field

The target column name in the flexi group that we are creating a list for.

### Fill query

This is the SQL command that is used to look-up the values to populate the drop-down list. This SQL should return two columns.

1. the first will be used as the 'value' for the lines in the drop-down list,
2. the second will be used as the 'description'.

If any other columns are returned these will be ignored.

No ordering of values will be carried out by the customisation, if you require the drop-down list to be ordered in a specific way you should include an 'order by' clause in your SQL statement.

You can add parameters within the SQL statement, and these should start with a \$ or @ symbol. These parameters are used in conjunction with the Fill Parameters flexi group that is detailed below. The current client can be specified by using \$client which is treated as a reserved parameter so client should not be specified in the Fill Parameters flexi group.

### Type-Ahead

In the web client, if checked, the field will appear as a type-ahead rather than a drop-down list.

This is **important for table sections** where drop down lists cannot be refreshed once initially loaded on screen start, type-ahead removes this limitation, so returned results can consider values in other fields on the current row.

**Note:** The field **must be a defined as an attribute** field for this to work.

### Default First

If this is checked then the customisation will default the first value in the list to be the currently selected value.

### Default Single

If this is checked should the SQL return only a single line, then this value will be set to the currently selected value.

### ASQL

Should be checked if the SQL you are providing includes ASQL commands.

### Comment (optional)

For documentation. This column may be hidden to give more space to other columns on screen.

## 3.3. Fill Parameters

Parameters in the fill SQL allow us to pick up values from anywhere in the current form, as the form is filled in.

The Fill parameters tab maps a parameter to a flexi field in the current form.

Parameters are identified by a prefix of '\$' or '@'. (Use of these characters in the SQL must be restricted to identifying parameters!)

If the Source group is a table, the value will be from the active row.

	Form attribute ID	Parameter	Source group	Source field	Target group	Target field
1	FF1	good_fx	HRSTATUS	good_fx	HRSTATUS	good_fx
2	FF1	average_fx	HRSTATUS	average_fx	HRSTATUS	average_fx
3	FF1	poor_fx	HRSTATUS	poor_fx	HRSTATUS	poor_fx
Σ						

Explanation of fields:

### Form Attribute ID

The attribute ID of the form that the flexi group above is attached to.

### Parameter

The parameter name that is being used in the fill query.

In the query, parameter names are prefixed with the '\$' or '@' character. You don't need to use the prefix in the parameter definition.

You do not need to set up the following system supplied parameters; \$client, \$user\_id or \$resource\_id. These are 'macro' values that are taken from the active session.

### Target group

The flexi group to be populated.

### Target field

The column name on the flexi group that is being populated.

This column should be set to an asterisk (\*) if the action is for a whole section, such as a 'FILL-TABLE'.

### Source Group

The flexi group on the form that contains the field we will be using to populate the parameter value.

### Source field

The field name that contains the value to populate the parameter value.

### 3.4. Fill Expressions

#### Form attribute ID

The attribute ID of the form that the flexi group above is attached to.

#### Trigger event

The event that will trigger the action.

- ONLOADED means the script action will be run when the form loads, e.g. after the Form Id has been entered.
- ONVALIDATED means that the script action will be run when the **trigger field** has been changed and validated.
- ONLOADVAL the script will be run both ONLOADED and ONVALIDATED.

#### Trigger group

The Flexi-group ID that contains the **trigger field**.

#### Trigger field

The field which triggers an ONVALIDATED script action.

#### Target group

The flexi-field group to be populated, or that contains the field to be acted on.

#### Target field

The column name to be set.

#### Expression

This is a .NET DataColumn expression to calculate the value of a column or create an **aggregate** column.

Any columns referenced must be in the Trigger flexi-group.

A new (hidden) field will be created on the trigger flexi-group for each expression.

Refer to <https://docs.microsoft.com/en-us/dotnet/api/system.data.datacolumn.expression> for full documentation.

Expressions can be more efficient than SQL if all the data you need is contained locally in the form.

## #

Should usually be 0, however it also provides a way to chain together script actions by creating multiple lines against the same 'Trigger Field', each with a different sequence number, these will then be run in the order specified.

## Keep

If this field is checked then the 'Fill Column Name' specified above will only be populated if it is currently empty.

When populating a table section, if this is checked, then any rows already in the table will be kept, if it is un-checked the table section will be cleared first before new rows are added.

## Append

Only applicable for 'FILL-TABLE' actions. If this is checked, and the table you are populating already contains rows, then any rows returned by the SQL will be appended to the table. No duplicate checking is performed.

## Validate target

When this box is checked, the value of every target column updated will be fully validated against the flexi-group definition.

Any ONVALIDATED script triggered by the target column will also be run.

Leaving this blank will stop validation of the target column.

### 3.5. Example Form

What follows is an example of using the various auto-population options described above to fill a sample HR form.

Details on the setup of this example form (FF1) can be found below.

The screenshot shows an HR form with the following sections:

- Employee**
  - Employee ID\*: 90020102
  - Chris Smiley
  - Position id\*: 0010
  - 2 Consultant
- Position detail**
  - Line manager: 112233
  - Leif Nielsen
  - Division:
  - Cost centre:
  - Rating: GOOD
- Absence**

Type	Start	End	Project	Work order	Interview outcome
LEAVE1	12/1/2018	12/6/2018	ABW1	ABW1-1	
SICK	1/3/2018	1/5/2018	ABW2	ABW2-1	

Buttons: Add, Delete

In this example we need to apply the following functionality.

- Any Employee ID can be entered.  
We only want the current employee and any staff that reports to them to be available. We will assume that this is not practical through data control.
- Any position can be entered.  
We only want the positions for the selected employee to be available.
- Line manager, Division and Cost centre are blank.  
We want them to default to the selected position.
- A work order may not be valid for the work order. We need the list to be restrained by the current work order.
- We want to be able to amend the cost centre if the form is for an employee who reports to us, else it should be read-only
- If the position is '0001' don't show the Division.
- The absence list is blank.  
We want to fill it in with the current employee absence history.
- Only allow entry of an interview outcome where the absence is due to sickness.



## 3.5.1. Example: FILL-LIST ONLOADED

To control the Employee ID list the following entry is made in the **Fill lists** tab. **ONLOADED** can be used because the resource ID is populated by a flexi-field macro.

In this example the employee list never changes once the form is loaded.  
It will let the User choose themselves or someone who reports to them.

No additional user defined parameters are required.

The SQL is.

```
SELECT d.dim_value,
       d.description
FROM
(
  SELECT client,
         resource_id
  FROM ahsrelvalue
  WHERE client = $client
        AND rel_value = $resource_id
  UNION
  SELECT $client, $resource_id
) r
INNER JOIN agldescription d
  ON d.client = r.client
  AND r.resource_id = d.dim_value
  AND d.attribute_id = 'C0';
```

1 - Attribute 2 - Address 3 - Relation 4 - Memo 5 - Fill Field 6 - Fill Lists 7 - Parameters 8 - Fill Expressions 9 - Fill Tool Tip

Attribute  Attr value

Description

Lists

	Form attribute ID	Trigger event	Trigger group	Trigger field	Fill group	Fill field	Fill query	#	Type ahead	Default first	Default single	ASQL
1	FF2	ONLOADED	HRPOSITION	*	HRPOSITION	employee_id_fx	SELECT d.dim_value,d.description FROM (SELECT client,resource_id FROM ahsrelvalue WHERE client=\$client AND rel_value=\$resource_id UNION SELECT \$client,\$resource_id) r INNER JOIN agldescription d ON d.client = r.client AND r.resource_id = d.dim_value AND d.attribute_id = 'C0'	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

### 3.5.2. Example: FILL-LIST ONVALIDATED

To control the Position ID list the following entry is made in the **Fill lists** tab.

**ONVALIDATED** is used as we need to get the list after an employee ID has been selected.

**Note:** FILL-LIST ONVALIDATED script will always be run when the form is loaded *and* when you amend a field. So, this script does not need to be replicated for ONLOADED, and we don't need a separate ONLOADEDVAL trigger event.

This time we must declare a parameter to pick up the employee ID from the form.

The fill query which must return a list of codes and descriptions is;

```
SELECT p.post_id, p.description FROM aprresourcepost r
INNER JOIN aprpost p ON p.client=r.client AND p.post_id=r.post_id
WHERE r.client=$client AND r.resource_id=@employee_id
AND GETDATE() BETWEEN r.date_from AND r.date_to AND r.status='N'
```

	Form attribute ID	Trigger event	Trigger group	Trigger field	Fill group	Fill field	Fill query	#	Type ahead	Default first	Default single	ASQL
1	FF2	ONVALIDATED	HRPOSITION	employee_id_fx	HRPOSITION	position_id_fx	SELECT p.post_id, p.description FROM aprresourcepost r INNER JOIN aprpost p ON p.client=r.client AND p.post_id=r.post_id WHERE r.client=\$client AND r.resource_id=@employee_id AND GETDATE() BETWEEN r.date_from AND r.date_to AND r.status='N'	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Map the form parameter in the **Fill Parameters** tab;

Parameters						
	Form attribute ID	Parameter	Source group	Source field	Target group	Target field
1	FF2	employee_id	HRPOSITION	employee_id_fx	HRPOSITION	position_id_fx

## 3.5.3. Example: FILL-COLUMN ONLOADEDVAL

To default in the Line Manager the following entry is made in the **Fill fields** tab

We are using **ONLOADVAL** because we need to default in the current line manager when the form is opened but reset the line manager if the employee changes. Apart from the Trigger event the rows are the same. Setting Division would be similar, just a different relation attribute.

The fill query is;

```
SELECT rel_value
FROM ahsrelvalue
WHERE client = $client
AND resource_id = $employee_id
AND getdate() BETWEEN date_from AND date_to
AND status = 'N'
AND rel_attr_id = 'C0'
```

1 - Attribute 2 - Actions 3 - Relation 4 - Memo 5 - Fill Field 6 - Fill Lists 7 - Parameters 8 - Fill Expressions 9 - Fill Tool Tip														
Attribute	FF001		Attribute								FF0003			
Description	Line Manager													
Fields														
	Form attribute id	Trigger event	Trigger group	Trigger field	Action	Target group	Target field	SQL	#	Keep	Append	Validate target	ASQL	Comment
1	FF2	ONLOADEDVAL	HRPOSITION	employee_id_bx	FILL-COLUMN	HRPOSREL	line_manager_fx	SELECT rel_value FROM ahsrelvalue WHERE client = \$client AND resource_id = \$employee_id	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Σ														

In the **Fill parameter** tab, the parameter used in the query is mapped to the correct field on the form.

Parameters						
	Form attribute ID	Parameter	Source group	Source field	Target group	Target field
1	FF2	employee_id	HRPOSITION	employee_id_fx	HRPOSREL	line_manager_fx

## 3.5.4. Example: FILL\_LIST in Table section

This works much the same as the FILL-LIST examples already given. But when a table section field is mapped as a parameter the current row value is used. The list values will be reset as you click from line to line.

The query used is;

```
SELECT work_order, description FROM atworkorder
WHERE client = $client AND project = $project
```

An absence line has a project and a work order. The work order values must be valid for the project on the same row. We only need to define ONVALIDATED as type-ahead lists will be refreshed as text is entered.

Form attribute ID	Trigger event	Trigger group	Trigger field	Fill group	Fill field	Fill query	#	Type ahead	Default first	Default single	ASQL
FF1	ONVALIDATED	HRABS	project.fx	HRABS	work_order.fx	SELECT work_order, description FROM atworkorder WHERE client = \$client AND project = \$project	0	0	0	0	0

In the **Fill parameter tab**, the parameter used in the query is mapped to the correct field on the form.

Form attribute ID	Parameter	Trigger group	Trigger field	Target group	Target field
FF1	project	HRABS	project.fx	HRABS	work_order.fx

## 3.5.5. Example: FILL-LIST on text field (drop-down list)

This example shows how a drop-down list can be created from a query to populate a single text field.

The query in this case is;

```
select top 100 attribute_id, description from agldimvalue where client = @client
```

	Form attribute ID	Trigger event	Trigger group	Trigger field	Fill group	Fill field	Fill query	#	Type ahead	Default first	Default single	ASQL
1	FF1	ONVALIDATED	HRPOSITION	employee_id_fx	HRPOSREL	rating_fx	select top 100 attribute_id, description from agldimvalue where client = @client	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

There are no parameters to set up.

The field, rating\_fx, will be set to a drop-down list with 100 values. Large drop-down lists can cause the form to become unusable and are better set to type-ahead.

This example could have been done using ONLOADED, but as mentioned before ONVALIDATED fill-lists are done ONLOADED too.

## 3.5.6. Example: READ-ONLY

Cost Centre is read only if the User is entering the form on behalf of themselves. So, a line manager would be able to amend the cost centre when submitting a form on behalf of a direct report. But cannot amend their own cost centre.

We will use the **READ-ONLY** action in the **Fill fields** tab. If the Fill query returns a row, then cost centre is read-only. The query used is.

```
SELECT 1 WHERE @employee_id_fx = $resource_id
```

The read only status is set when opening the form and or changing the employee ID when linked to the **ONLOADEDVAL** event ff

	Form attribute ID	Trigger event	Trigger group	Trigger field	Action	Target group	Target field	SQL	#	Keep	Append	Validate target	ASQL
1	FF2	ONLOADEDVAL	HRPOSITION	employee_id_fx	READ-ONLY	HRPOSREL	cost_centre_fx	SELECT 1 WHERE @employee_id_fx = \$resource_id	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

In the **Fill parameter** tab, the parameter used in the query is mapped to the correct field on the form.

	Form attribute ID	Parameter	Source group	Source field	Target group	Target field
1	FF2	employee_id_fx	HRPOSITION	employee_id_fx	HRPOSREL	cost_centre_fx

Note: Script can still populate read-only fields.

## 3.5.7. Example: READ-ONLY in table row

Unlike the HIDE action, READ-ONLY works well in table sections row by row.

In this example we only accept data entry in the interview column when the absence type is SICK on the current row.

Description: READ ONLY table cell													
Form attribute id	Trigger event	Trigger group	Trigger field	Action	Target group	Target field	SQL	#	Keep	Append	Validate target	ASQL	
1 FF2	ONVALIDATED	HRABS	type_fx	READ-ONLY	HRABS	interview_fx	SELECT 1 WHERE @type_fx != 'SICK'	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

In the **Fill parameter tab**, the parameter used in the query is mapped to the correct field on the form.

Description: READ ONLY table cell						
Form attribute ID	Parameter	Source group	Source field	Target group	Target field	
1 FF2	type_fx	HRABS	type_fx	HRABS	interview_fx	

When the table is loaded by a FILL-TABLE, the **Validate target** flag should be set on to trigger the READ-ONLY.

## 3.5.8. Example: HIDE field

In this example we hide the division field if the position code is '0001'.

We need to re-check whenever the position value is amended, so we use **ONLOADEDVAL**. If the query returns at least one row with at least one column then the target field will be hidden. If the query returns no rows the field will be visible.

In the Fill Field tab, we will use the following T-SQL to check the position code.

SELECT 1 WHERE @position\_id\_fx = '0001'

Description: HIDE field													
Form attribute id	Trigger event	Trigger group	Trigger field	Action	Target group	Target field	SQL	#	Keep	Append	Validate target	ASQL	
1 FF2	ONLOADEDVAL	HRPOSITION	position_id_fx	HIDE	HRPOSREL	division_fx	SELECT 1 WHERE @position_id_fx = '0001'	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

In the **Fill parameter tab**, the parameter used in the query is mapped to the correct field on the form.

Description: HIDE field						
Form attribute ID	Parameter	Source group	Source field	Target group	Target field	
1 FF2	position_id_fx	HRPOSITION	position_id_fx	HRPOSREL	division_fx	

Note: Script can still populate hidden fields.

### 3.5.9. Example: SECTION ACTIONS - HIDE & READ-ONLY

It is possible to make whole sections read-only or hidden through script.

In our sample form we only want to hide the comments section if there are no absences for the current resource.

If there are absences, we only want the resource who was absent to be able to enter comments. So, the section should be read only if the form is not for the current resource.

Description: HIDE/READ-ONLY section											
	Form attribute ID	Trigger event	Trigger group	Trigger field	Action	Target group	Target field	SQL	#	Keep	Append
1	FF2	ONLOADED/VAL	HRPOSITION	employee_id_fx	READ-ONLY	HRCOMMENT	*	SELECT 1 WHERE @employee_id_fx = \$resource_id	0	<input type="checkbox"/>	<input type="checkbox"/>
2	FF2	ONLOADED/VAL	HRPOSITION	employee_id_fx	HIDE	HRCOMMENT	*	SELECT 1 WHERE NOT EXISTS (SELECT 1 FROM apsabsence WHERE client = \$client AND resource_id = @employee_id_fx)	0	<input type="checkbox"/>	<input type="checkbox"/>

The SQL used to make the comments section READ-ONLY is.

*SELECT 1 WHERE @employee\_id\_fx = \$resource\_id*

The SQL used to HIDE the comments section is.

*SELECT 1 WHERE NOT EXISTS (SELECT 1 FROM apsabsence WHERE client = \$client AND resource\_id = @employee\_id\_fx)*

The employee\_id\_fx parameter, shared by the two SQLs, is mapped as below.

Description: HIDE/READ-ONLY section					
	Form attribute ID	Parameter	Source group	Source field	Target group
1	FF2	employee_id_fx	HRPOSITION	employee_id_fx	HRCOMMENT

In this example the order in which these actions are executed is not important. The section will be read-only if the following SQL returns one or more rows. But that will be irrelevant if it is subsequently hidden!

**NOTE:** If you make a tab read-only by turning off update permission in the menu access, no row will be saved to that flexi-group table. Even if you default values into fields on a section it will not be saved if the user does not have update permission. If you need 'read-only' data saved, then make each field in the section read-only but make sure that the tab/section has update permissions.

## 3.5.10. Example: FILL-TABLE

Absence							
	Type	Start	End	days	Project	Work order	Interview outcome
<input type="checkbox"/>	LEAVE1 Absence to	01/12/2018	08/12/2018	7	ABW1 Project AB	ABW1-1 wo abw1-1	
<input type="checkbox"/>	SICK	03/01/2018	05/01/2018	2	ABW2	ABW2-1	

To load the absence history for the selected employee we will use a **FILL-TABLE** query in the **Fill fields** tab.

The query used is;

```
SELECT absence_code type_fx,date_from start_fx,date_to end_fx,dim_2 project_fx,dim_4
work_order_fx,DATEDIFF(dd, date_from, date_to)+1 days_fx FROM apsabsence WHERE client=$client
AND resource_id=$employee_id
```

Note that the column names of the query match the flexi-fields we want to populate.

We don't need to specify all of them. Columns that don't match a field name are ignored, with no warnings given.

We will set the target field to \* as we are filling a table.

As we haven't checked **Keep** or **Append**, old data will always be cleared when we change the employee.

**No validate** has been checked.

1 - Attribute 2 - Address 3 - Relation 4 - Memo 5 - Fill Field 6 - Fill Lists 7 - Parameters 8 - Fill Expressions 9 - Fill Tool Tip

Attribute:  Attribute value:

Description:

Fields

	Form attribute id	Trigger event	Trigger group	Trigger field	Action	Target group	Target field	SQL	#	Keep	Append	Validate target	ASQL	Comment
1	FF2	ONLOADDEFAULT	HRPOSITION	employee_id_fx	FILL-TABLE	HRABS	*	SELECT absence_code type_fx,date_from start_fx,date_to end_fx,dim_2 project_fx,dim_4 work_order_fx,DATEDIFF(dd, date_from, date_to)+1 days_fx FROM apsabsence WHERE client=\$client AND resource_id=\$employee_id	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

In the **Fill parameter** tab, the parameter used in the query is mapped to the correct field on the form.

Parameters						
	Form attribute ID	Parameter	Source group	Source field	Target group	Target field
1	FF2	employee_id	HRPOSITION	employee_id_fx	HRABS	*

There is a calculated field in this query. The number of days absent is calculated from the start and end date.

As we can amend the dates in this table, or add new rows, we will need to add some more script to calculate the days when the data is amended (below).



## 3.5.11. Example: Calculated Fields

Calculated fields are configured in the same way for tables and single fields.

When in a table the current active row values are used.

Absence							
	Type	Start	End	days	Project	Work order	Interview outcome
<input type="checkbox"/>	LEAVE1 Absence to	01/12/2018	08/12/2018	7	ABW1 Project AB	ABW1-1 wo abw1-1	
<input type="checkbox"/>	SICK	03/01/2018	05/01/2018	2	ABW2	ABW2-1	

In this example (as above) a read-only field, **days**, must be set to the difference between two other fields, **start\_fx** and **end\_fx**. The value must be calculated and set when either of these two fields is amended.

As we have two triggering fields we need to set up the following script.

1 - Attribute   2 - Address   3 - Relation   4 - Memo   5 - Fill Field   6 - Fill Lists   7 - Parameters   8 - Fill Expressions   9 - Fill Tool Tip														
Attribute	FF001		Attribute value											
Description	DateDiff													
Fields														
	Form attribute id	Trigger event	Trigger group	Trigger field	Action	Target group	Target field	SQL	#	Keep	Append	Validate target	ASQL	Co
1	FF2	ONVALIDATED	HRABS	start_fx	FILL-COLUMN	HRABS	days_fx	select datediff(dd,@start_fx,@end_fx)+1	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
2	FF2	ONVALIDATED	HRABS	end_fx	FILL-COLUMN	HRABS	days_fx	select datediff(dd,@start_fx,@end_fx)+1	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

The two parameters are defined as follows;

Parameters						
	Form attribute ID	Parameter	Source group	Source field	Target group	Target field
1	FF2	start_fx	HRABS	start_fx	HRABS	days_fx
2	FF2	end_fx	HRABS	end_fx	HRABS	days_fx

DataColumn Expressions do not have the same rich list of functions as SQL, so this example could not easily be done using Expressions. So, we are using the following TSQL to do the calculation.

```
SELECT DateDiff(dd,@start_fx,@end_fx)+1
```

We have duplicated the calculation in the initial table load as it is more efficient than validating every field.

### 3.5.12. Example: Radio Buttons with Expressions

In this example a flexi-field group has three check boxes to grade performance. Only one check box of the three should be set at any one time.

Every time a box is checked we need to clear the other two boxes. That's two fields to set for each of three triggers, making six updates to script.

Note that a checked box has a value of 1; an unchecked box has a value of zero.

Note that we set the no validate flags. Without this the box you just checked can get un-checked again!

Although this could be done with SQL, this example uses expressions as they are more efficient.

Note: Expressions do not need parameters!

1 - Attribute 2 - Address 3 - Relation 4 - Memo 5 - Fill Field 6 - Fill Lists 7 - Parameters 8 - Fill Expressions 9 - Fill Tool Tip												
Attribute	FF2		Attr value	FF2								
Description	Radio buttons (expressions)											
Expressions												
	Form attribute id	Trigger event	Trigger group	Trigger field	Target group	Target field	Source group	Expression	#	Keep	Append	Validate target
1	FF2	ONVALIDATED	HRCOMMENT	poor_fx	HRCOMMENT	good_fx	HRCOMMENT	poor_fx=0 AND average_fx=0	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	FF2	ONVALIDATED	HRCOMMENT	poor_fx	HRCOMMENT	average_fx	HRCOMMENT	poor_fx=0 AND good_fx=0	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3	FF2	ONVALIDATED	HRCOMMENT	average_fx	HRCOMMENT	good_fx	HRCOMMENT	average_fx=0 AND poor_fx=0	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4	FF2	ONVALIDATED	HRCOMMENT	average_fx	HRCOMMENT	poor_fx	HRCOMMENT	good_fx=0 AND average_fx=0	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5	FF2	ONVALIDATED	HRCOMMENT	good_fx	HRCOMMENT	average_fx	HRCOMMENT	poor_fx=0 AND good_fx=0	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6	FF2	ONVALIDATED	HRCOMMENT	good_fx	HRCOMMENT	poor_fx	HRCOMMENT	average_fx=0 AND good_fx=0	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

It is important that we **do not call validation** for the target field in this example. If we do the process will get caught in an endless loop and crash!

## 3.5.13. Example: Aggregate Expression

In this example we will display the total absence days.

Absence							
<input type="checkbox"/>	Type	Start	End	days	Project	Work order	Interview outcome
<input type="checkbox"/>	LEAVE1 Absence to	01/12/2018	08/12/2018	7	ABW1 Project AB	ABW1-1 wo abw1-1	
<input type="checkbox"/>	SICK	03/01/2018	05/01/2018	2	ABW2	ABW2-1	

Total days absent
9.00

To get this to work we need to switch on the validation flag when days are calculated. This would need to be done when the table is initially loaded. This will then call the script we have created on the start and end dates.

Fields	Form attribute id	Trigger event	Trigger group	Trigger field	Action	Target group	Target field	SQL	#	Keep	Append	Validate target	ASQL
1	FF2	ONLOADEDVAL	HRPOSITION	employee_id_fx	FILL-TABLE	HRABS	*	SELECT absence_code type_fx,date_from start_fx,date	0	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

On the date calculation script, we also need to turn on validation so that the aggregation function is called.

Fields	Form attribute id	Trigger event	Trigger group	Trigger field	Action	Target group	Target field	SQL	#	Keep	Append	Validate target	ASQL	Comment
1	FF2	ONVALIDATED	HRABS	start_fx	FILL-COLUMN	HRABS	days_fx	select datediff(dd,@start_fx,@end_fx)+1	0	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
2	FF2	ONVALIDATED	HRABS	end_fx	FILL-COLUMN	HRABS	days_fx	select datediff(dd,@start_fx,@end_fx)+1	0	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

The following values are entered in the FILLEXPRESSION flexi-group to calculate the total number of absence days.

Description

Aggregate expression

Expressions

	Form attribute id	Trigger event	Trigger group	Trigger field	Target group	Target field	Source group	Expression	#	Keep	Append	Validate target	Comment
1	FF2	ONLOADEDVAL	HRABS	days_fx	HRCOMMENT	total_days_fx	HRABS	SUM(days_fx)	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

By turning the validation on we will have slowed down the initial table load. If there are a lot of rows being loaded the delay could become an issue. For a single page of rows there should not be an issue.

## 3.6. Flexi-group setup for examples

The example form flexi groups are setup as follows:

Flexi-field group: **HRIPOSITION**
Description: **Employee**

Type: **Flexi-field group**
Table name: **abhrposition**

View format: **Fields**
Status: **Active**

☐ Add date interval with validation
 Search criteria: **Show no search criteria**

	Title	Multilingual	Column name	Data type	Attribute	Related attribute	Display format	Field width	Data length	Mand	Hide	Sum	Read	Val	Def	Value lookup ID
1	Employee ID	<input type="checkbox"/>	employee_id_fx	Attribute	RESNO		Code	25	12	10	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	11
2	Position id	<input type="checkbox"/>	position_id_fx	Attribute	POSNO		Code	25	12	10	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	11

Flexi-field group: **HRIPOSREL**
Description: **Position detail**

Type: **Flexi-field group**
Table name: **abhrposrel**

View format: **Fields**
Status: **Active**

☐ Add date interval with validation
 Search criteria: **Show no search criteria**

	Title	Multilingual	Column name	Data type	Attribute	Related attribute	Display format	Field width	Data length	Mand	Hide	Sum	Read	Val	Def	Value lookup ID
1	Line manager	<input type="checkbox"/>	line_manager_fx	Attribute	RESNO		Code	25	12	10	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	11
2	Division	<input type="checkbox"/>	division_fx	Attribute	DIVISION		Code	25	12	10	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	11
3	Cost centre	<input type="checkbox"/>	cost_centre_fx	Attribute	COSTC		Code	25	12	10	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	11
4	Rating	<input type="checkbox"/>	rating_fx	Text			Text any case	25	25		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0

Flexi-field group: **HRABS**
Description: **Absence**

Type: **Flexi-field group**
Table name: **abhrabs**

View format: **Table**
Status: **Active**

☐ Add date interval with validation
 Search criteria: **Show no search criteria**

	Title	Multilingual	Column name	Data type	Attribute	Related attribute	Display format	Field width	Data length	Mand	Hide	Sum	Read	Val	Def	Value lookup ID
1	Type	<input type="checkbox"/>	type_fx	Attribute	ABSCODE		Code	25	12	10	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	11
2	Start	<input type="checkbox"/>	start_fx	Date				12	12		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0
3	End	<input type="checkbox"/>	end_fx	Date				12	12		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0
4	days	<input type="checkbox"/>	days_fx	Integer			Small	8	8		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0
5	Project	<input type="checkbox"/>	project_fx	Attribute	PROJECT		Code	25	12	10	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	11
6	Work order	<input type="checkbox"/>	work_order_fx	Attribute	WORKORD		Code	25	12	10	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	11
7	Interview outcome	<input type="checkbox"/>	interview_fx	Text			Text any case	255	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0

Flexi-field group: **HRCOMMENT**
Description: **Absence**

Type: **Flexi-field group**
Table name: **abhrcomment**

View format: **Table**
Status: **Active**

☐ Add date interval with validation
 Search criteria: **Show no search criteria**

	Title	Multilingual	Column name	Data type	Attribute	Related attribute	Display format	Field width	Data length	Mand	Hide	Sum	Read	Val	Def	Value lookup ID
1	status	<input type="checkbox"/>	status_fx					50	1000		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0
2	good	<input type="checkbox"/>	good_fx					1	1		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0
3	average	<input type="checkbox"/>	average_fx					1	1		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0
4	poor	<input type="checkbox"/>	poor_fx					1	1		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0
5	Unacceptable	<input type="checkbox"/>	hide_unacceptable_fx					1	1		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0
6	Total days absent	<input type="checkbox"/>	total_days_fx					8	8		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0

Attribute: **FFGENG**
Table name: **FF AUTOPOP**

	Flexi-field group	Description	Data table	Tab	Tab name	Multilingual	Section	Section name	Multilingual	Mandatory	Connect to rule	
1	HRIPOSITION	Position detail	abhrposition	1	Employee	<input type="checkbox"/>	1	Employee	<input type="checkbox"/>	<input type="checkbox"/>		N
2	HRIPOSREL	Position detail	abhrposrel	1	Employee	<input type="checkbox"/>	2	Position detail	<input type="checkbox"/>	<input type="checkbox"/>		N
3	HRABS	Absence	abhrabs	1	Employee	<input type="checkbox"/>	3	Absence	<input type="checkbox"/>	<input type="checkbox"/>		N
4	HRCOMMENT	comments	abhrcomment	2	Comments	<input type="checkbox"/>	1	comments	<input type="checkbox"/>	<input type="checkbox"/>		N

## 4. ENHANCED DIA

Functionality is provided to limit access to forms based on the logged in user. For example, in the following:

Resource	Form ID	Position	Line Manager
10001 – Brian Best	100	12345 Position A	10002 – Jane Bloggs
10001 – Brian Best	101	12346 Position B	10003 – Jenny Smith

Line manager 10002 should only be able to access form ID 100 and line manager 10003 should only be able to access form ID 101.

This is provided by the creation of a custom view detailing which forms users have access to and a system setup value holds what view is used on a form:

	Name	Pos	Text 1	Number 1	Text 2	Number 2	Text 3	Number 3	Description
?	FORM_DIA_VIEW								
1	FORM_DIA_VIEW	0 EN		0	BT01	0	uvibtttestusers	0	

Name: FORM\_DIA\_VIEW

Pos: 0-n

Text 1: The ABW client this entry relates to (can be set to \* for all clients)

Text 2: The attribute ID of the Form this entry relates to

Text 3: The name of the view to use for the lookup

The requirements for the view are that it must contain at least the following columns:

attribute\_id = This will be the attribute ID of the form

dim\_value = This will be the form ID

client = the ABW client

user\_id = the ABW user ID

As an example, if user SYSTEM opened the 'test1' form ID for form BT01 the following SQL will be run against the uvibtttestusers view (specified in the system setup value above):

```
SELECT dim_value
FROM uvibtttestusers
WHERE attribute_id = 'BT01'
AND dim_value = 'test1'
AND client = 'EN'
AND user_id = 'SYSTEM'
```

If this SQL returns 1 or more rows then the user will be allowed to open the form, if no rows are returned then the user will not be able to open that form.

An example of the view used for testing is as follows (made up of two views):

UVIBTTESTFORMS:

```
SELECT dim.client, dim.attribute_id, dim.dim_value, afx1.employee_id, afx1.position_id
FROM dbo.agldimvalue AS dim INNER JOIN dbo.afxbttest AS afx1 ON afx1.client = dim.client AND afx1.attribute_id =
dim.attribute_id AND afx1.dim_value = dim.dim_value
WHERE (dim.attribute_id IN ('BT01'))
```

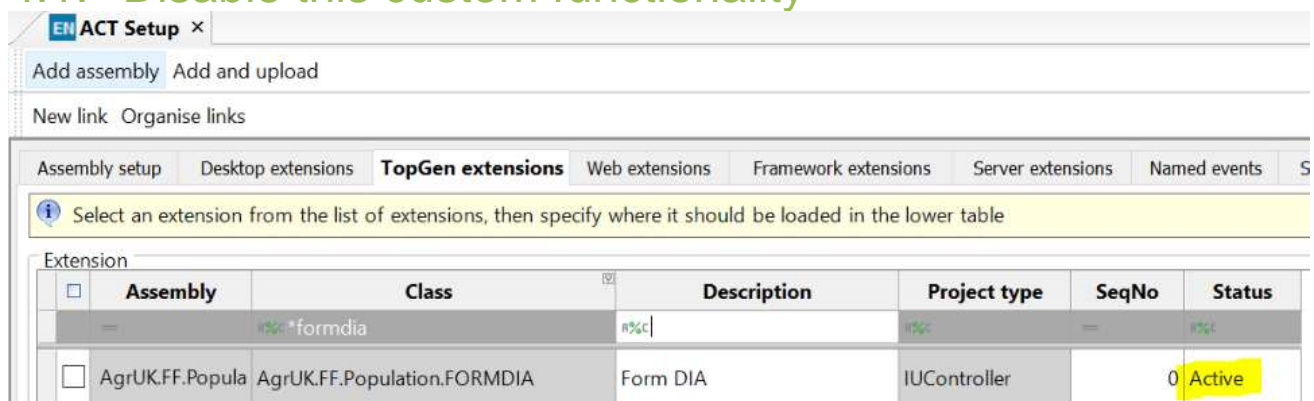
UVIBTTESTUSERS:

```
SELECT uvi.client, uvi.attribute_id, uvi.dim_value, uvi.employee_id, uvi.position_id, pos.rel_value, acr.user_id
FROM dbo.uvibttestforms AS uvi
INNER JOIN dbo.apposrelvalue AS pos ON uvi.client = pos.client AND uvi.position_id = pos.post_id AND pos.rel_attr_id
= 'C0'
INNER JOIN dbo.acruserlink AS acr ON acr.client = uvi.client AND acr.attribute_id = pos.rel_attr_id AND acr.dim_value =
pos.rel_value
```

It should be noted that this enhanced distribution access can only be used to further restrict access to forms, it cannot be used to grant access to a form that a user is prevented from accessing normally by standard setup, data control or DIA.

The accompanying field help for the Form ID field will remain un-altered by this setup, as such if a user opens field help they may be able to see forms listed that they do not have access to, although they will still be unable to open those forms if they select them.

## 4.1. Disable this custom functionality



The screenshot shows the 'ACT Setup' window with the 'TopGen extensions' tab selected. A yellow banner instructs the user to 'Select an extension from the list of extensions, then specify where it should be loaded in the lower table'. Below this is a table with the following data:

Extension	Assembly	Class	Description	Project type	SeqNo	Status
<input type="checkbox"/>	=	r%*formdia	r%	r%	=	r%
<input type="checkbox"/>	AgrUK.FF.Popula	AgrUK.FF.Population.FORMDIA	Form DIA	IUController	0	Active

In the ACT Setup screen, set the status of this extension to Closed to disable this functionality.

All the other form scripting functionality will continue to work without it.

## 5. FORM USERID UPDATE

In some circumstances, for example when creating forms to be used for performance appraisal or similar, the user who creates the form (i.e. a line manager) will have their Agresso UserID stamped against the form when it is saved. If data control is in place this may prevent the user who the form is actually for (i.e. the employee of the line manager) from accessing it at a later stage

To work around this problem functionality is provided to update the user ID of the form based on another field within the form. A look-up will be performed against another field on the form that contains a resource ID (Attribute 'C0'), the user ID for that resource will be found and that will be set as the user\_id of the forms entry on the agldimvalue table.

A new system setup value must be added for each form you require this functionality to run against and the values are as follows:

Name: FORM\_USERID\_FIELDS

Pos: 0-n

Text 1: The client this applies to or \* for all.

Text 2: The Attribute ID of the form this applies to.

Text 3: The Flexi Field section on the form that contains the resource ID field to use for the UserID lookup.

Description: The Flexi Field column name that contains the resource ID field to use for the UserID lookup (must contain a valid resource ID from attribute 'C0').

For example:

EN System setup values X

Setup Template New link Organise links

Sys.setup EN

	Name	Pos	Text 1	Number 1	Text 2	Number 2	Text 3	Number 3	Description
?	FORM_USERID_FIELDS								
1	FORM_USERID_FIELDS	0	EN		0	PD01		0	EMPSECTION
								0	employee_id_fx

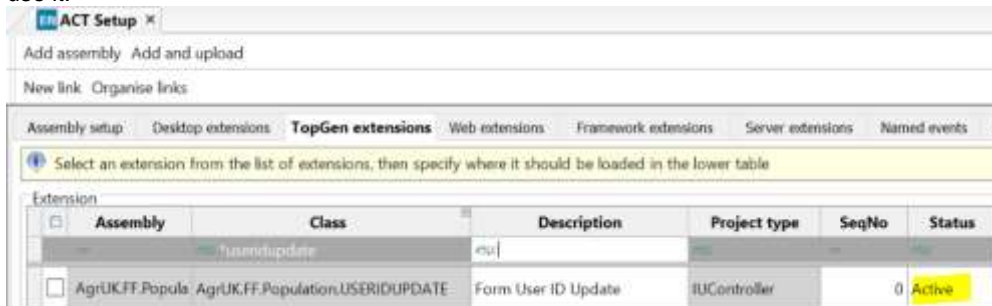
In the above when a user saves a 'PD01' form, the resource ID contained in the field 'employee\_id\_fx' in section 'EMPSECTION' will be read, this resource ID will be used to find the resources associated user ID via the acruerlink table and this user ID will be stamped against the created forms entry on agldimvalue.

This functionality only works on forms saved via the TCR022 screen.



## 5.1. Disable this custom functionality

If this custom functionality is not required it can be disabled, although it should not be a problem to just leave it on and not use it.



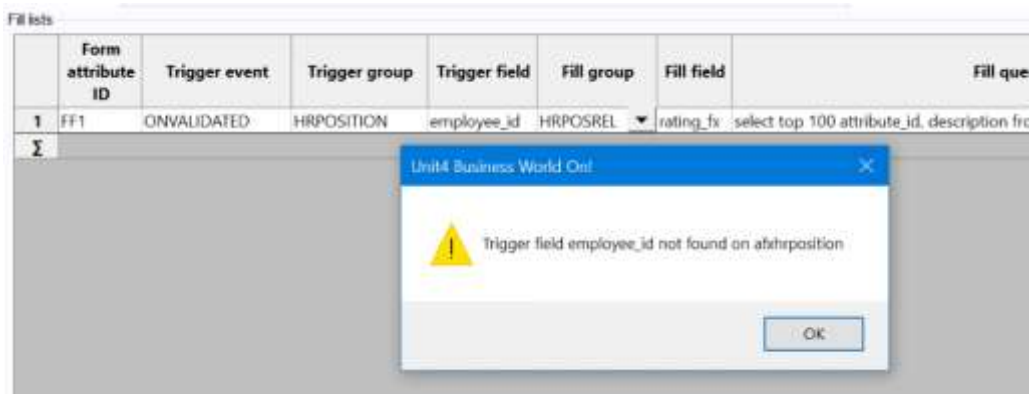
In the ACT Setup screen, set the status of this extension to Closed to disable this functionality. All the other form scripting functionality will continue to work without it.



## 6. VALIDATING SCRIPT DEFINITION

When saving an Attribute master file with linked flexi-group script definitions they will be validated. This will happen on both the Desktop and Web.

The flexi-groups and flexi-fields will be checked;



The SQL will be checked;

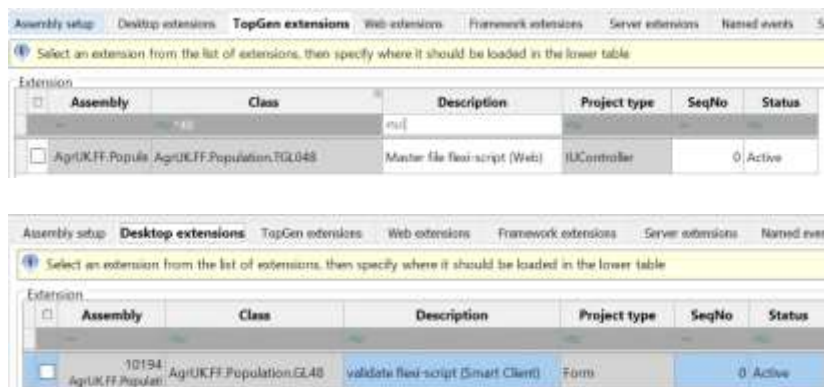


The master file data will be saved even if there are warnings.

### 6.1. Disable Validation of script when entering in the web.

In the ACT Setup screen, set the status of these extensions to Closed to disable validation.

All the other form scripting functionality will continue to work without it.



## 7. SAFE SQL

This flexi-group scripting tool runs user defined SQL queries. This introduces the risk of accidental or malicious updates to the database. To ensure that no SQL database amendments are made by this tool, the SQL queries are executed and retrieve data but no changes are committed to the SQL database.

To achieve this safe execution transactions are created and aborted. The overhead of doing this should not slow things noticeably.

### 7.1. Disable safe SQL

If for any reason you need to turn this safe transactional execution off, turn on a system parameter called ACT\_FF\_UNSAFE.

The screenshot shows the 'System parameters' window with the 'Setup' tab selected. The window displays a table of system parameters. The parameter 'ACT\_FF\_UNSAFE' is highlighted in blue. The table has columns for Name, Mod, Max length, Value, On/off, Sys. setup, and Client level. The 'ACT\_FF\_UNSAFE' parameter is currently set to 'ACT' with a maximum length of 1. The 'On/off' checkbox is checked, and the 'Sys. setup' and 'Client level' checkboxes are unchecked.

	Name	Mod	Max length	Value	On/off	Sys. setup	Client level
?	ACT_FF_UNSAFE	ACT			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1	ACT_FF_UNSAFE	ACT	1		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

## 8. DEBUGGING

There is additional logging that can be enabled within the client script processing. This uses the standard Business World Logger that can be configured in the Management console, so it is configured separately for web and desktop clients, e.g.

This feature is switched on at a form level. The list of form IDs to log is configured in system parameter FFAUTO\_DEBUG. The list of form IDs is delimited by a semi-colon, e.g. UA1;UA2;UA3;. The list must end with a semi-colon.

Sys.setup	EN	Client	EN				
	Name	Mod	Max length	Value	On/off	Sys. setup	Client level
?	FFAUTO*				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1	FFAUTO_DEBUG	ACT	25	FF1;	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

The entries in a csv log file could look something like this;

```
134, 2018-12-07 12:08:54.629, 1, 00 00:00:00:024, 240056, Information, AboveNormal, "ACT FF: DefaultField line_manager_fx"
138, 2018-12-07 12:08:54.720, 1, 00 00:00:00:021, 219990, Information, AboveNormal, "ACT FF: RunScript ONVALIDATED line_manager_fx"
139, 2018-12-07 12:08:54.747, 1, 00 00:00:00:023, 230011, Information, AboveNormal, "ACT FF: SetReadOnly cost_centre_fx"
173, 2018-12-07 12:08:55.568, 1, 00 00:00:00:021, 219999, Information, AboveNormal, "ACT FF: [FILL LIST] Field: position_id_fx Event:
ONVALIDATED SQL: SELECT p.post_id, p.description FROM aprresourcepost r INNER JOIN aprpost p ON p.client=r.client AND p.post_id=r.post_id
WHERE r.client='EN' AND r.resource_id='90020102' AND GETDATE() BETWEEN r.date_from AND r.date_to AND r.status='N'"

```