



Customizing rules

Abas APS 2024.Q3.1

VERSION: APS2201R8N03

DATE: 29.07.2025

Contents

| | | |
|----------|----------------------------------------------------------------------|----------|
| 1 | Upgrade note | 3 |
| 1.1 | FOPs and FOP.txt | 3 |
| 1.2 | Enumerations, identifiers, call parameters, keys and infosystems | 3 |
| 1.3 | Tables of variables | 3 |
| 1.4 | Settings and scripts in the simcron directory | 3 |
| 2 | Rules, notes and tips for customizing APS for Abas | 4 |
| 2.1 | IFOPs in Infosystems | 4 |
| 2.1.1 | Creating an APS standard infosystem from an Abas standard infosystem | 4 |
| 2.2 | Call parameters | 5 |
| 2.3 | Enumerations | 5 |
| 2.3.1 | Custom setup parameters | 5 |
| 2.3.1.1 | Screen fields in simulation tab in identifier | 5 |
| 2.4 | Adding customized variables | 8 |
| 2.5 | Customizing screens | 8 |

1 UPGRADE NOTE

1.1 FOPs and FOP.txt

During the upgrade process, the FOPs included in scope of supply are updated in the **allspp**, **allsstd** and **allspro** directories. Existing FOPS will be overwritten and may be deleted. The **fop.txt** is checked for the required entries. Missing entries are added. FOP calls in collective FOPs are not considered and can lead to double calls after an upgrade. Collective FOPs should therefore be avoided.

1.2 Enumerations, identifiers, call parameters, keys and infosystems

During the upgrade procedure the objects included in scope of delivery will be updated and either overwritten or deleted.

1.3 Tables of variables

The tables of variables are generally only added to during an upgrade. If a variable is renamed or deleted, this information will be included in the upgrade notes. Variables named according to the **y9** naming convention will not be affected by the upgrade.

1.4 Settings and scripts in the simcron directory

The settings and scripts will be updated automatically as far as possible.



If you are using custom scripts, we recommend contacting Abas Support to arrange what manual customized may be required.

2 RULES, NOTES AND TIPS FOR CUSTOMIZING APS FOR ABAS

The following rules and mechanisms should be observed when customizing in order to preserve the upgrade-ability of APS.

2.1 IFOPs in Infosystems

- IFOPs are programs that are not changed during an APS update and which are started from the APS standard programs (depending on the working directory in the password definition).
- IFOP.<infosystemsearchword>.EV Programs that use this name convention are called in the infosystem after the FOP event (at the end of any existing FOPS included in the standard program).
- IFOP.<infosystemsearchword>.TAB Programs that use this name convention are called after table events that create or delete rows (at the end of any existing FOPS included in the standard program), and are run for each row.



You need not necessarily customize APS standard infosystems in order to customize APS. By using IFOPs they retain their update-ability. The IFOPs should be saved in a custom FOP directory.

2.1.1 Creating an APS standard infosystem from an Abas standard infosystem

The FPPRODLIST and FPAUO infosystems are based on the related Abas standard infosystems: PRODLIST and AUO.

Customized infosystems are created based on the standard infosystems of the underlying Abas version.

The procedure for doing so is always the same:

1. Copy the infosystem.
2. Copy the FOPs from the Abas standard directory to the allspro directory and customize them so that the FOP calls in these FOPS refer likewise to the allspro directory. The standard FOPs should not be further customized.
3. Add APS fields to the Infosystems and screens as required.
4. All FOP calls in the infosystem must be adapted as APS FOPS, according to the following convention:
 - a. Copied standard FOP call (if applicable)
 - b. APS function (if applicable)
 - c. IFOP call (generally .EV, except for allspro/<aps-infosystem-searchword>.TAB in the table)

The program-controlled archive FOPs (**archivfop**) are unique. They can take effect when they are called by other programs: Report header (**bkopf**) and footer (**bfuss**), group header (**grkopf**) and footer (**grfuss**). Each individual case must be checked as to whether the above applies, and must be ensured in the event of an upgrade. This also applies to the table (**tab**). In the APS context it can generally be assumed that FOPs, which are entered there by default, are executed during row operations.

This means you can customize infosystems as follows:

1. Adapt the APS infosystem (not customize):
 You can add simple functions using IFOPs, provided the field events are not already in use, and that you follow the IFOP convention. This approach is more stable during an upgrade, if at a later time an Abas or APS function is added to these fields.

2. Adapting a copy of an APS infosystem - Add your own functions using one of the following methods:
 - a. Using the IFOP system
 - b. Using an APS FOP call according to the example above in your own FOPS

2.2 Call parameters

The call parameters included in scope of delivery should not be edited because they will be overwritten during an update.

2.3 Enumerations

The enumerations included in scope of delivery should not be modified because they may be overwritten during an update.

2.3.1 Custom setup parameters

Four setup parameters are included in the APS scope of delivery. You can add custom setup parameters. You can add setup parameter types using the **FPRUESTPARAART 4913012** enumeration, and adding new identifiers. You will need to add these customized setup parameter types to the enumeration again after performing an upgrade.

2.3.1.1 Screen fields in simulation tab in identifier

You can control the display and behavior of the setup parameter screen and setup information screen using the fields in the simulation tab, depending on the setup parameter object type in the setup information or in the setup parameter itself.

Groups (yfptpvgrouplist)

To add a new setup parameter with Abas database objects (reference fields), enter the valid groups.

Screen context (yfptpvscope)

This is the screen context that is displayed depending on the object type (ypart) in the setup parameter itself or the setup parameter (ysrparam) in the setup information.

Possible standard values:

ALLS_POLYMORPH_PM - Display for reference fields

ALLS_POLYMORPH_BEZ - Display for description (not reference fields)

ALLS_POLYMORPH_BUTTON - Display for selection buttons (not reference fields)

ALLS_POLYMORPH_TXT1 - Display for disk fields in setup parameter

ALLS_POLYMORPH_TXT2 - Display for disk fields in the setup information

If you add custom screen contexts, they must not start with ALL. These contexts are for internal use.

Field fill FOP (yfptpvffop), Field validation FOP (yfptpvfvfop), Field exit FOP (yfptpvfxfop)

Here you can add FOPs that control the behavior of the respective object type.

Setup parameter fields and setup information fields

Saving and displaying the values for setup parameters are separated to enable flexible use. The parameters can be based on Abas master data objects, Abas transaction data objects or freely defined values, as required.

Disk fields are used for saving. The display and entry can be performed, depending on the setup parameter type and field, directly or using skip fields/buttons.

| Field name | Abas type | Source | Entry via | Alias |
|---------------|-----------|-------------|-------------------------------------------|--------------|
| yobjekt | NT64 | Disk | yoselect ypobjekt ymobjekt | |
| yobjbez | GL34 | Disk | Direct | |
| ypobjekt | VP99 | yobjekt | Direct | ypobjektt |
| ymobjekt | NT25 | yobjekt | Direct yoselect | |
| yoselect | DVB1 | | | |
| yrparamvorw | NT64 | Disk | ybrparamvor ymrparamvor yprparamvor | |
| yrparamvorbez | GL34 | Disk | Direct | |
| yprparamvor | VP99 | yrparamvorw | Direct | yptrparamvor |

| Field name | Abas type | Source | Entry via | Alias |
|----------------|-----------|--------------|----------------------------------------------|---------------|
| ymrparamvor | NT25 | yrparamvorw | Direct ybrparamvor | |
| ybrparamvor | DVB1 | | | |
| yrparamnachw | NT64 | Disk | ybrparamnach ymrparamnach yprparamnach | |
| yrparamnachbez | GL34 | Disk | Direct | |
| yprparamnach | VP99 | yrparamnachw | Direct | yptrparamnach |
| ymrparamnach | NT25 | yrparamnachw | Direct ybrparamnach | |
| ybrparamnach | DVB1 | | | |

Events in setup information and setup parameters

The generic FOPs integrated in APS Standard can be used for custom setup parameter types, for controlling the events for the fields mentioned above. These contain the basic functions required. You can control custom functions using the FOPs in the setup parameter types.

Screen entry:

- Values in disk fields carried over to standard display fields

Field events (entry fields):

- Check performed, whether type was entered
- FOPs started at the end, according to setup parameter type entered
- Field information is transmitted to the FOPs. The **allspro/FP.POLYMORPH.SHOW.PREPARED.FO** FOP can be used to test the variable output while developing and testing. The variables can serve as templates for custom generic FOPs (by replacing dots or format strings).

Field fill event (FELDFUELL):

- For reference fields: Selection according to defined group

Field validation event (FELDPRUEF):

- Check performed for double rows
- In setup information, yrparamvorw and yrparamnachw checked for variances in values
- For reference fields: Check performed for correct group

Field exit event (FELDAUS):

- Disk field assignment

Button after event (yoselect, ybrparamvor, ybrparamnach):

- The button after events execute the FELDFUELL, FELDPRUEF and FELDAUS FOPs after each other.

Custom fields

If your application requirements cannot be met using the customizing options for entering and displaying text, you can add custom y9 fields with a custom screen context (if required). The simulation and Production Planner require the values to be saved in the disk field mentioned above. Transferring disk fields to custom display fields on screen entry is not currently supported and must be run separately.

The customized field control programs should be written similarly to the FO calls for the field events in the standard APS fields for setup parameters.

2.4 Adding customized variables

All APS variables are already created as y-variables according to the Abas customization rules. In order to be able to distinguish APS variables from custom variables during the upgrade procedure, all new variables in APS objects must have the prefix **y9**. New APS variables in standard objects are created as **yfp** variables. Variables used in customizing must **not** begin with yfp.

2.5 Customizing screens

All APS screens included in the scope of delivery are custom screens that can be further customized. When you upgrade APS, you will be requested to confirm whether you want to overwrite the screens.



Forterro Deutschland Abas GmbH
Gartenstraße 67
76135 Karlsruhe, Germany