Financial Instruments

Customization Upgrade Guide
Release 5.7
Area: Market Data

Latest version of this document

The latest version of this document can be found at https://docs.avaloq.com

Feedback

Please send any feedback to documentation@avaloq.com

Avaloq Group Ltd. Allmendstr. 140 ¦ CH-8027 Zürich ¦ Switzerland

Version history

Version / Date	Section	Description of the change
5.7v2 / 29 April 2024		Added a note to the Replacement of EVAL_SCEN_CSS by EVAL_CSS section, that two base parameters need to be set to false in 5.7 and above. Moved information from the introduction to each specific section to make the information more clear.
5.7v1 / 11 December 2023		Updated to add description that cf_mdl_id and opt_mdl_id are now mapped by default, and added a note about the de-released classification 528.
5.7v0 / 21 March 2023		This is a new document for Release 5.7.

Contents

1 Introduction

This topic is for customization specialists. It describes changes that you may need to make to your existing customization **before** you upgrade to Avaloq Core release 5.7.

1.1 Replacement of EVAL_SCEN_CSS by EVAL_CSS

You can opt to update your customization for this change before upgrading to Avaloq Core release 5.7:

- In releases 5.2 to 5.6, the Asset Evaluation Engine supports both the new and the old customization. By default it continues to use the old customization, but you can set the `avq.ae` base parameter's `use_legacy_param` item to false to enable the new customization described in this section.
- Starting with release 5.7, the new customization will automatically be used.
- For releases 5.7 and above, both parameters need to be set to false. **This is mandatory for all clients.**

The following changes in customization are included in the Avaloq Asset Evaluation Engine in release 5.7:

- The source EVAL_SCEN_CSS is removed and replaced by the EVAL_CSS source.
- The task **Calculate evaluation scenarios** (id: 1847; task_eval_scen) no longer exists. It was originally used to improve performance, but is no longer needed. There is no replacement.
- The checkbox `use pre-calculated values` on task **Evaluation of financial products program** (id: 1848; task_eval) is removed because task **Calculate evaluation scenarios** no longer exists.
- The following `eval_css` functions need to be overwritten in customization, as the return values of ‘null’ in kernel are no longer mapped automatically to the kernel models:
  - `cf_mdl_id`: must return `code_eval_mdl.dcf`. At the moment there is no other model available.
  - `opt_mdl_id`: must return the appropriate option pricing models, depending on the asset class, for example `code_eval_mdl.bs` for equity, commodity and FX options.

In the past, the Asset Evaluation Engine had a lot of performance and memory problems associated with the use of the MEM_DOC_ASSET data dictionary by the EVAL_SCEN_CSS source. The problems were caused by the fact that the asset order was loaded into memory.

To solve these problems, in Avaloq Core release 5.7 we:

- Changed the data dictionary used by the Asset Evaluation Engine's customization from MEM_DOC_ASSET to a new data dictionary, MEM_ASSET
- Replaced the CSS source used by the Asset Evaluation Engine from EVAL_SCEN_CSS to a new source, EVAL_CSS

The new MEM_ASSET data dictionary is faster because asset orders are no longer loaded into memory. It is accessed by the new customization source EVAL_CSS.

For all functions of the new EVAL_CSS:

- The `i_doc_asset` parameter from the MEM_DOC_ASSET data dictionary is replaced by `i_asset` from the new MEM_ASSET data dictionary
- The `i_eval_date` parameter from the MEM_DOC_ASSET data dictionary is not present in the new MEM_ASSET data dictionary

The following EVAL_SCEN_CSS functions are removed completely:

Function removed	Explanation
`cf_hor_in_years`	Perpetuals need to have a call or put date otherwise no evaluation is possible
`use_par_flo`	The new Avaloq kernel code now behaves as if `use_par_flo` is set to false
`notnl_date_expr`	Depended on `use_par_flo`

The following functions are renamed:

Old function name in EVAL_SCEN_CSS	New function name in EVAL_CSS
`pb_rfig_intr_calc_method_id`	`intr_calc_method_id`
`pb_rfig_compd_mtd_id`	`compd_mtd_id`
`pb_rfig_src_price_type_id`	`src_price_type_id`
`vlt_struct_collect_id`	`vsc_id`

The following functions have new default behaviour:

Function name	New default setting in EVAL_CSS
`opt_mdl_id`	Null
`alw_vlt_fallbk`	False
`intr_calc_method_id`	Null
`compd_mtd_id`	Null
`src_price_type_id`	Null

1.2 Use of MEM_ASSET instead of MEM_DOC_ASSET

Note regarding de-release of classification 528 (previously announced in Financial Instruments Customization Upgrade Guide (Doc. ID: 4369), release 4.9):

If you are relying on classification 528 for downstream consumers, one solution would be to create your own derived classifications and classes to mimic the prior 528 classification.

In Avaloq Core 5.7, the Asset Evaluation Engine customization no longer uses the MEM_DOC_ASSET data dictionary. Instead it uses the new MEM_ASSET data dictionary.

The following fields are no longer available; they are completely removed:

Field removed	Description
`block_id`	It’s redundant because the new Asset Evaluation Engine already discards template assets.
`dflt_trade_curry_id`	Checking for exotic currencies can now be done with a derived class.
`outstand_pieces`	Checking for jumbo bonds can now be done with a derived class.
`sort_alpha`	This was derived from four keys: iso, tks, isin and sor_nr. The aim was to classify certain assets. This can now also be done using classes.
`eff_date`	The `eff_date` from the order DDIC is not available in the new MEM_ASSET data dictionary. The replacement has to be found on a case-by-case basis.

Keys

Keys are no longer available the customization should not be asset or key specific. If you still have such a use case, you should mark it with a specific class.

Additions

Additions are no longer available. They will have to be replaced with a corresponding class.

Classes

All kernel classes are supported. However the syntax used to access classes has changed:

- Old pattern: mem_doc_asset.obj_extn.classif_id
- New pattern: mem_asset.class(classif_id)

Here's a code example using the old pattern:

```
  function opt_mdl_id(
    i_doc_asset                                 mem_doc_asset
   ,i_calc_md_scen_id                           id table code_md_scen
   ,i_eval_date                                 date
  ) return id table code_eval_mdl
  is
  begin
    if i_doc_asset.obj_extn.ass_type_id in
        (btt.asset.obj_extn.ass_type.class.cfc) then
      return code_eval_mdl.bn;
    else
      return null;
    end if;
  end if;
  exception
    when others then
      err.raise_fa('eval_scen_css.opt_mdl_id('||i_doc_asset||', '||i_calc_md_scen_id||', '||i_
eval_date||')');
  end opt_mdl_id;
```

And here's a code example using the new pattern:

```
  function opt_mdl_id(
    i_asset                                    mem_asset
   ,i_calc_md_scen_id                           id table code_md_scen
```

```
) return id_table_code_eval_mdl
is
begin
  if i_asset.class(btt.asset.obj_extn.ass_type_id) in
      (btt.asset.obj_extn.ass_type.class.cfc) then
    return code_eval_mdl.bn;
  else
    return null;
  end if;
end if;
exception
  when others then
    err.raise_fa('eval_css.opt_mdl_id('||i_asset||', '||i_calc_md_scen_id||')');
end opt_mdl_id;
```