avaloq
An NEC Company

Stock Exchange

Customization Upgrade Guide
Release 5.7
Area: Securities Trading

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
5.7v0 / 22 September 2023		This is a new document for release 5.7.

Contents

1 Introduction

This document is for customization specialists. It describes the changes you might want to make to your customization when you upgrade to Avaloq Core release 5.7, as a result of bugs that were fixed in 5.7.

1.1 Deactivate duplicate quantity entry in CODE_DOC_REVS_CAUSE

There are two entries in CODE_DOC_REVS_CAUSE that indicate an incorrect quantity for the STEX metatype, as shown in the following table.

In addition to these two entries, the code table contains several more STEX related entries delivered by kernel and their setup has not been changed.

ID	Activ	Intl_id	Meta_typ_id	Name
7	+ (true)	quantity	1 (stex)	Wrong quantity
8	+ (true)	quantity	1 (stex)	Wrong quantity

In previous releases, both entries were active by default. If customization did not rule out one of the entries, a user would see both identical entries and would need to choose between them when entering the reversal cause.

Fix in release 5.7

The entry with row ID =8 has been deactivated and the two rows are now delivered as shown in the following table.

ID	Activ	Intl_id	Meta_typ_id	Name
7	+ (true)	quantity	1 (stex)	Wrong quantity
8	null (false)	quantity	1 (stex)	!Wrong quantity

This means that the user will now see only one entry (with row ID =7); there is no longer a need to choose.

In general, it is up to the customer to decide which of the two identical entries is used. You could also activate the row with ID =8 again to show both entries. This is entirely up to the customer. It is recommended, however, to use only one of these entries.

Orders referencing the row with ID =8 can still be loaded and processed (verified, reversed, rectified, partially billed) without error. Such orders will show the name with a preceding “!” indicating that this is an inactive entry. But the user would not need to act on this (for example, by changing the value).

One benefit of this action in kernel is that there is now an entry for the “quantity” field in the generated CODE_DOC_REVS_CAUSE DDIC. That means, no lookups are necessary if you want to reference (or compare) the quantity entry with row ID =7 in Avaloq script code. You can now directly access this entry using the DDIC expression `code_doc_revs_cause.quantity`.

Be careful when you are not using the kernel setup and you are comparing code table IDs (see table above). In this case, you might still want to use lookups.

Results of fix in release 5.7

The following cases need to be considered based on this fix:

- Customer has not intervened in previous releases regarding the setup; that is, both entries were active and shown to the user.

  Action needed: None needed.

  Consequence: None. The user will no longer see two identical entries and, therefore, does not need to choose between the two.

- Customer has not intervened in previous releases and changed the setup in customization.

  Different sub-cases are possible:

  - Customer deactivated entry with ID =7, using only the entry with ID =8.

    Action needed: If the customer decides to continue with this setup and has not yet overridden the CODE_DOC_REVS_CAUSE related source to activate the entry with ID =8, then this must now be done explicitly. Activate the row with ID =8 if you wish to continue using this entry. Also maintain the setup that includes deactivation of the entry with ID =7.

    Consequence: None. The user will still see only one of the two entries (as before, the entry with ID =8).

  - Customer deactivated the entry with ID =8, using only the entry with ID =7.

    Action required: None. If you want, you can delete the overwrite for these two entries.

    Consequence: None. The user will still see only one of the two entries (as before, the entry with ID =7).

  - Customer deactivated entries with IDs =7 and =8.

    Action required: None. The deactivation of the two entries is still in place. If you want, you can remove the deactivation override of entry with ID =8 because it is not needed.

    Consequence: None, as still neither entry is shown to the user.

It is recommended that you check your customization code for use of the two entries and ensure that access to the field is consistent and adheres to your business decisions for/against use of row with ID =7. You can use the g-search feature in ice to support you in this task.

1.2 FIX Doc New Single IN: presence of PreAllocGrp creates market order and triggers allocation

In previous releases, a client order was created by default from an incoming FIX Doc New Single message.

In release 5.7, different logic is triggered depending on the presence or absence of the PreAllocGrp component in an incoming FIX Doc New Single message.

PreAllocGrp included

The presence of the PreAllocGrp component in an incoming FIX Doc New Single message triggers the logic shown in the following table.

Tag 78 NoAllocs value	Result
Higher than 0 (zero) and lower than the value of the `avq.intf.fix\max_alloc_supported` base parameter (default 200)	• Market order is created • The incoming message is put in a "Wait for Processing" status • TASK_STEX_PRC_ALLOC_INFO is submitted to asynchronously create allocation orders based on the message content
Higher than the value of the `avq.intf.fix\max_alloc_supported` base parameter (default 200)	• Client order is created with a post-it of type "Information STEX", and sub-type "Trader Notification" • The message is put into an "Error" status

PreAllocGrp not included

If the PreAllocGrp component is not present in the incoming message, a client order is created by default. You must define the logic in the fix_doc_new_single_in_css.msg#action function to trigger different logic.

The following table shows the values that can be returned based on the logic defined.

Value	Result
Null	Default logic calculated by kernel is triggered
`code_stex_intf_msg_action.new_clt_doc`	Client order created
`code_stex_intf_msg_action.new_mkt_doc_id`	Market order created
`code_stex_intf_msg_action.new_mkt_prealloc_doc_id`	• Market order is created • The incoming message is put in a "Wait for Processing" status • TASK_STEX_PRC_ALLOC_INFO is submitted to asynchronously create allocation orders based on the message content

1.3 Outgoing message FIX Allocation Instruction (35=J): tag 79 AllocAccount filled

Based on the FIX protocol, tag 79 AllocAccount is optional in the AllocGrp component, but it is mandatory when the component is present, and is required by most of the FIX engines.

Absence of tag 79 AllocAccount triggers a rejection from the Avaloq FIXBridge.

Fix in release 5.7

In release 5.7, the tag is now filled with the following logic:

- **Prio 1:** external container number (`cont_extl`)
- **Prio 2:** from the key defined in `avq.intf.fix\cont_obj_key_id`
- **Prio 3:** container symbol (`cont_sym`)
- **Prio 4:** container number (`cont_nr`)

Customization might need to be adapted if you are filling this tag in `fix_alloc_instr_css.ovr_msg`.