6.0 BRO___Changelog 2025-09-16, 6.0

Summary

1. The BRO Transporttool supplies now the option to export rules as JSON. Thus, the size of an exported decision area can be drastically reduced.
2. A bRobots® installation is now managed by transaction *SAINT*.
3. If you want to upgrade to 6.0, then you first must upgrade to 5.4.9.
4. WebScreen:
    1. There are still known issues. Only import this release if you are not relying on WebScreen.

Changes

bRobots®

Features

(FR2475) New version scheme and introduction of BC Sets

Version system

Until now the scheme was the following: *&ltMajor Part1&gt.&ltMajor Part2&gt.&ltService Pack&gt*. The version scheme has been updated in the following way: *&ltMajor&gt.&ltService Pack&gt.&ltPatch&gt*.

Additionally the release strategy has been altered in the following way:

1. A change log page in [[wiki.akquinet.at]]will be created only for a *&ltMajor&gt.&ltService Pack&gt* release (e.g. 6.0). A *&ltMajor&gt.&ltService Pack&gt* combination is from now on referred to as *release*.
    1. If the *&ltService Pack&gt* = 0 then this is referred now to as *major release*.
    2. If the *&ltService Pack&gt* &gt 0 then this is referred now to as *service pack release*.
    3. For maintenance contracts: a *major release* is treated the same way as a *service pack release*.
2. Non-breaking changes will be released as patch on top of an existing release as *&ltMajor&gt.&ltService Pack&gt.&ltPatch&gt* (e.g. 6.0.1). A *&ltMajor&gt.&ltService Pack&gt.&ltPatch&gt* combination is from now on referred to as *patch*.
3. The release notes of a patch will be added to the change log to the release it belongs to (e.g. the notes for patch 6.0.1 can be found in the change log of release 6.0).
4. Breaking changes will result in a new release.

Update of installation/upgrade procedure

Until now each release was shipped as transport of copies. This has been altered. From now on files are shipped that can be imported using the transaction *SAINT*.

Customizing will no longer be shipped at all. Instead, all necessary customizing will be supplied by BC Sets.

BC Sets

Alongside this versioning change the shipping of BC Sets is introduced. This should help during upgrade scenarios. BC Sets will be shipped for each release and patch. The contents are as follows:

1. A BC Set for a release will contain all customizing changes of the last release plus the customizing introduced by this release (excluding its patches).
    1. Such BC Sets are named `/HKS/BRO_&ltMajor&gt_&ltService Pack&gt` with a description of *akquinet bRobots &ltMajor&gt.&ltService Pack&gt Release (Client specific)*.
    2. If a BC Set is not client specific (i.e. contains data for all clients) then the BC Set is named `/HKS/BRO_&ltMajor&gt_&ltService Pack&gt_S` with a description of *akquinet bRobots &ltMajor&gt.&ltService Pack&gt Release (System specific)*.
    3. This means that each BC Set for a release can be considered as a BC Set for an installation (see statement below the enumeration for release 6.0).
    4. Such BC Sets must be used for upgrading bRobots® too.
2. A BC Set for a patch will contain all customizing changes since the last patch or its according release if it is the first patch.
    1. Such BC Sets are named `/HKS/BRO_&ltMajor&gt_&ltService Pack&gt_&ltPatch&gt` with a description of *akquinet bRobots &ltMajor&gt.&ltService Pack&gt.&ltPatch&gt Patch (Client specific)*.
    2. If a BC Set is not client specific (i.e. contains data for all clients) then the BC Set is named `/HKS/BRO_&ltMajor&gt_&ltService Pack&gt_&ltPatch&gt_S`with a description of akquinet bRobots &ltMajor&gt.&ltService Pack&gt.&ltPatch&gt Patch (System specific)_.

The only exception for the ruling above is the BC Set for the release 6.0 as this is the release which introduces BC Sets. The BC Set of release 6.0 contains all customizing which was previously available as installation customizing transport request.

General implications

1. For an upgrade to a new release all release BC Sets have to be imported starting from the release that is currently available in the system.
2. For a patch import all patch BC Sets have to be imported starting from the release that is currently available in the system.
    1. If an upgrade is done from a previous release to a patch of a newer release then step 1 and step 2 must be performed.
3. A fresh installation using directly a specific patch level must first include the release BC Set and then all the patch BC Sets within the release.
4. No BC Sets will be created for single tickets. If you need the customizing singled out, then please contact your bRobots® advisor at akquinet HKS business technologies GmbH.

Reengineering actions:

ATTENTION: BC Sets must be handled with caution.

BC Sets will include data of the following transactions:

1. */HKS/BRO_USERGROUP* – BRO: Authority groups
    1. For decision area *KEN_HKS*
    2. For decision area *BUP_HKS*
    3. For decision area *MAT_HKS*
2. */HKS/BRO_CUST_ACTTYP* – BRO:Action types
    1. Lines with the column *Implementing class* with the pattern */HKS/**
3. */HKS/BRO_AU_REG* – Registering Utils-functions
    1. Lines with the column *Implementing class* with the pattern */HKS/**
4. */HKS/BRO_RTCONST02* – BRO Runtime constant maintenance
    1. Lines with the column *Customizing source* set to *P*
5. */HKS/BRO_MDM_OBJCU02* – BRO MDM Change object customizing
6. */HKS/B_SHCTH_CFG* – Process & Service Workitem shortcuts
    1. Lines with the column *Customizing source* set to *P*
7. /HKS/BRO_MDM_CPTREE – Process & Service Cockpit
    1. Layouts starting with */WF+*
8. */HKS/B_SLA* – Process & Service Monitor
    1. Layouts starting with */WF+*

If you have overwritten any of the above stated data then you should not just import the entire BC Set. Instead, you must manually check which parts of the BC Set you can actually import on your system.

Some example modifications that you must manually consider when importing BC Sets:

1. Changing a class of a custom action type in transaction */HKS/BRO_CUST_ACTTYP*.
2. Changing a line with the column *Customizing source* set to *P* in transaction */HKS/BRO_RTCONST02*.

Examples which you can ignore when importing BC Sets:

1. Extra structures or key attributes in transaction */HKS/BRO_MDM_OBJCU02*.

ATTENTION: Special step needed for existing installations.
If you are upgrading your bRobots® installation to 6.0 then you must import bRobots® 5.4.9 first. This is necessary as 6.0 does not ship with deletion statements for deleted components in the namespace */HKS/*. The next releases will include deletion statements that were issued after bRobots® 6.0.

Afterwards import the transport of copies H4DK902775 which contains the report _/HKS/W_UREENG_FR2475_. The transport of copies is available here. Run that report in SA38 without any parameters.

Finally import 6.0 via transaction SAINT.

BRO

Features

(FR2540) Export and import of decision area data is supported to / from JSON

files

It is now possible to import and export data in JSON notation using transaction /HKS/BRO_TRANSPORT. Therefor the new checkboxes “JSON file” were added to the user interface. If the checkbox is selected during file import or export, the resulting file will be in JSON notation.

Reengineering actions:
If you used Method /HKS/IF_W_STREAM_EXCHANGE_RT~SERIALIZE of class /HKS/CL_W_STREAM_JSON_RT for converting ABAP to JSON in company-specific Z* coding, please check if you defined the following (optional) parameters which have been removed with this ticket. Adopt your coding accordingly:
I_FIELDNAMES_UPPERCASE, I_DONT_ESCAPE_LTGT, I_SUPPRESS_ITAB, I_IGNORE_BOOLEAN, I_NUMC_AS_NUMERIC, I_DATE_FORMAT, I_DONT_REPLACE_LINEBREAKS

If you used Method /HKS/IF_W_STREAM_EXCHANGE_RT~DESERIALIZE of class /HKS/CL_W_STREAM_JSON_RT for converting JSON to ABAP in company-specific Z* coding, please check if you defined the following (optional) parameter which have been removed with this ticket. Adopt your coding accordingly:
I_DONT_REPLACE_LINEBREAKS

Bugfixes

(BR2621) Update function modules were called even on no changes

The following update function modules were always processed on a commit work even if there were no changes:

1. /HKS/B_AW_SAVE_WFCTI_DB
2. /HKS/W_AS_INTR_SAVE_WFD_DB

From now on they are just called if there were registered changes.

Screen Definition

Bugfixes

(BR2625) Using numerical class characteristics with fixed values as screen definition drop-down fields terminated with a short dump

A class characteristic which was defined numerical having a fixed value list (no additional values were allowed) could have led to system dumps if the class was used in screen definition field “Classification”.
This happened if the field type “Classication” was used with the option Show Fixed values as drop-down

Solution:

Adopted comparison in case of drop-down of numerical characteristics values and returned external value representation to drop-down in case of pre-defined values.

Impacts:
The program ended with a short dump. Unsaved data was lost.

(BR2624) Screen definition failed if a class name contained special characters

If a class name contained characters with others then the following then the screen definition was failing:

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ_0123456789
```

This error could be noticed immediatly when field labels were all missing. After any user input, the program ended with system dump.

Solution:
All forbidden characters are now removed from the resulting screen attribute name.

Impacts:
System dump. All unsaved data got lost.

(BR2620) Default value for read-only search help parameter was not respected at field value check

When checking an entered field value with the attached elementary search help, an eventual default value of the search help parameter was cleared even if it was defined without the input-flag.
So the search help could never return the expected value as the empty parameter triggered no result.
The user could not notice this behavior when opening the search help. In that constellation, the default value was respected and all allowed values were shown in the search help popup.

It was also not possible to overcome this issue by setting the default value explicitly by rule as the screen definition compiler returned the following message then: Message class: /HKS/W_AS Numer: 075 Text: Parameter &1 is no input parameter

From now on the default values are not cleared during value check if the respective parameter is not inputable.

(BR1919) Using the same field for multiple table selection was allowed

Until now it was allowed to use the same field for a table selection in the action sub type *Table*. This resulted in multiple issues:

1. Data loss since only the table selection of one table was eventually passed on to the rule set

2. Inconsistent behavior between SAPGUI and Webscreen: Screen Definition in SAPGUI allowed this behavior whilst doing this in Webscreen resulted in a systemp dump in the backend.

Reengineering actions:
ATTENTION: You have to manually check your rules if this behavior was utilized and change the field accordingly. Since normally this is not intentionally it should be sufficient to either remove the field altogether or to replace it with a unique one.

If this re-engineering is not performed, then it might result in processes crashing.

Solution:
Using the same field for multiple table selections is no longer allowed. This is checked during runtime and will result in an error stopping the execution.

MDM+ BP/CMD/VMD

Bugfixes

(BR2613) Update function modules to use new MOT structures

The following function modules used old MOT structures. This resulted in a system dump upon using them.

1. */HKS/BRO_MDM_READ_CUSTOMER*: Used *CMDS_EI_EXTERN* and is now using */HKS/SW_MDM_CMDS_EI_EXTERN*.
2. */HKS/BRO_MDM_READ_VENDOR*: Used *VMDS_EI_EXTERN* and is now using */HKS/SW_MDM_VMDS_EI_EXTERN*.

MDM+ Workflow

Bugfixes

(BR2612) /HKS/B_AW_SAVE_WFCTI: Switch to update function module

The function module */HKS/B_AW_SAVE_WFCTI* directly wrote the database table */HKS/BT_AW_WFCTI*. If in the same call sequence other ABAP code (e.g. function module */HKS/B_AW_AC_APPLY_WFCTI*) used the update function module to write the database tale */HKS/BT_AW_WFCTI* then a race condition occurred. This in turn resulted in undefined behavior and resulted in sometimes losing updates on the database table */HKS/BT_AW_WFCTI*.

(BR2525) /HKS/B_SLA: Infinite loop

When analyzing requests using the Process & Service Monitor (transaction */HKS/B_SLA*), then an infinite loop might have occurred.

The infinite loop was triggered if the display type “Request reasons” was used on an object type that has Z fields available to the database table */HKS/BT_AW_WFCTR* and also configured them properly using the transaction */HKS/B_WF_CTA*.

Templates

Features

(FR2609) /HKS/BRO_MDM_GEN_MOT: Added default variants

The transaction */HKS/BRO_MDM_GEN_MOT* is now shipped with default variants for the MOT IDs shipped with bRobots®. The variants include:

Bugfixes

(BR2583) Table definition might have resulted in weird lines

If a table definition was performed using input tree nodes, then the resulting grid might have received values that were not input by the user.

The exact case to reproduce:

1. Create a table containing at least 2 database key nodes in a process which loads data into that grid by linking MDM data (i.e. using the business partner data).
2. Start the process.
3. Enter at least one line in the grid of step 1.
4. Perform a function return triggering a screen refresh.
5. The grid now contains now an additional line that was not entered.

Examples were this was noticed: business partner tax numbers and identification numbers.

Solution:
The error was caused by the MDM module automatically selecting the last database key and passing the selected value on to the BRO runtime values. This leads to undefined behavior as soon as a grid is shown using such database keys as then the value is directly selected plus the grid containing the values is shown.

An automatic selection of database keys is not correct generally anyway and has been removed. This in turn solves the issue with table definitions.

Template KEN_HKS

Bugfixes

(BR2601) Read-only supplied an "Execute" button

If a process was set to read-only (*SV_HKS_CHANGE_MODE = C_NO*), then an *Execute* button was supplied in case multiple screens were present. This is not correct as a modification is not possible and therefore triggering a request or saving something does not make sense.

Impacts:
The *Execute* button is not shown anymore if a process is set to read-only.

License tools

Bugfixes

(BR2596) /HKS/L_LICENSE08 did not show correct bRobots® version

The transaction to show the bRobots® license information of the current system (transaction */HKS/L_LICENSE08*) did not show the correct bRobots® version.

Patches

No patches are available for this release.

Downloads

First time installation

1. akquinet bRobots 6 Release - Add-On installation package
2. akquinet bRobots 6 Release KEN_HKS - Decision area KEN_HKS and its configuration

Upgrade

This release does not include upgrade files. For more information see FR2475.