Handle multiple module configurations

Feature request specification

Authors:

|  |  |  |
| --- | --- | --- |
| Name | Changes | Date |
| Fabian Kliemannel | Initial creation | 16.09.13 |
| Fabian Kliemannel | Added Conclusions, details on UI changes, persistence decision | 23.09.13 |
| Fabian Kliemannel | Added comparison dialog changes | 24.09.13 |

Concerned software tool:

E/E Browser

Content

[1. General description 3](#_Toc367781137)

[2. Persistence of module configurations 3](#_Toc367781138)

[2.1. Persistence with proprietary xml format 4](#_Toc367781139)

[2.2. Persistence with KBL format 4](#_Toc367781140)

[3. Realization in E/E Browser 5](#_Toc367781141)

[3.1. Conclusion 6](#_Toc367781142)

[4. User interface proposal 6](#_Toc367781143)

[4.1. Ribbon menu changes 6](#_Toc367781144)

[Conclusion 6](#_Toc367781145)

[4.2. Changes to modules pane 7](#_Toc367781146)

[Conclusion 8](#_Toc367781147)

[4.3. Changes in module configuration manager 9](#_Toc367781148)

[Conclusion 10](#_Toc367781149)

[4.4. Changes in compare dialog 12](#_Toc367781150)

[Conclusion 13](#_Toc367781151)

# General description

This specification is about enabling the E/E Browser to effectively handle several module configurations at a time. In E/E Browser a “Module Configuration” is a set of harness modules. Each module is defined in a KBL file. The defined modules are unique identified by their part number. A module configuration can therefore be seen as a set of part numbers each one referencing a single module. Several module configurations need to be temporary saved in E/E Browser and persistently in a specific data format.

Supporting several module configurations will enable following use cases:

* Rapidly changing between module configurations
* Import/export of module configurations, e.g. a specific configuration of a test vehicle or a produced vehicle
* Comparison of module configurations
* Easier exchange of user created module configuration between users or business partners

# Persistence of module configurations

There are two ways a configuration is made persistent.

1. Daimler will add the configurations to the KBL file contained in HCV. This file is read only for the E/E Browser. This is why E/E Browser will never change a configuration in the KBL file or add a configuration to KBL file.
2. User defined configurations added in E/E Browser are saved in a proprietary xml format. We cannot store user defined configurations, because HCV specification forbids a second KBL file.

The module configuration (proprietary) xml files can be integrated in the HCV files or provided as extra files. Both ways have suitable use case and both should be supported by E/E Browser.

**Aspects for monolithic HCV file with included module configurations**

* User chooses a harness or module in Connect, also he chooses a configuration and the resulting HCV is provided via download
* One file exchange with partners
* No confusion about assignment of module configuration to proper HCV

**Aspects for independent module configuration file**

* **Adding new configurations to opened HCV**
* **Exchange user created configurations (import/export of module configurations)**
* **Configurations should contain information about their harness to ensure mapping between HCV and module configuration files**

Generally there has to be further research on how module configurations look at . How are they identified and how does the list of modules of a module configuration look like.

## Persistence with proprietary xml format

Module configurations can be saved in a proprietary XML format. This format has to specified and integrated in the HCV container. The E/E Browser needs a new interface for handling those files including export of user configurations. There also has to an interface in the source system, where module configurations are normally created and saved. A module configuration file can contain several module configurations. Using an extra xml file for module configurations has the advantage of transporting module configurations of multiple harnesses in one file. This is especially useful in the scope of whole car configuration, where multiple harnesses are configured at a time.

The file will contain 3 xml classes:

1. **Root object “ModuleConfigurationContainer”** will simply be the root container for all other classes in a module configuration file.
2. **“HarnessModuleConfigurations”** is a container for all module configurations of a single harness. It contains attributes to identify the concerning harness and a list of module configurations.
3. **“ModuleConfiguration” contains finally a single module configuration. It consists of identifying attributes and a list of modules that are part of the configuration. A string containing a list of part numbers of the modules will be enough, since they are unique in Daimler AG. The part numbers will be separated by whitespace as known from KBL.**

**There will be further information about the needed attributes after we analyzed examples from Daimler.**

## Persistence with KBL format

It is possible to include the module configuration in KBL. Therefore the KBL standard contains the class “Harness\_configuration”. A KBL harness configuration equals the module configuration mentioned before. It is identified by several attributes to identify a configuration. Some are senseless in the scope of configurations.

Obligatory attributes for “Harness\_configuration”:

* Part\_number
* Company\_name
* Model\_year
* Car\_classification\_level\_2
* Description
* Abbrevation
* Version

The exact mapping of Attributes will be available as soon as we analyzed a Daimler configuration example.

A “Harness\_configuratoin” has furthermore associations to modules.

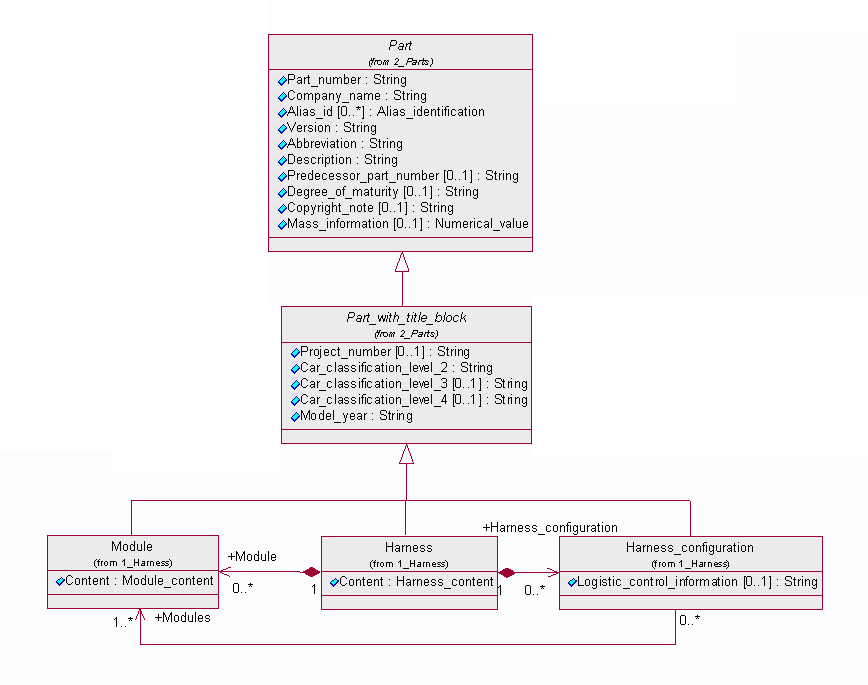


Figure : Harness\_configuration in KBL model

# Realization in E/E Browser

As mentioned the E/E Browser needs to handle several configurations. Configurations can be imported separately or as part of a HCV. Configurations contained in a KBL cannot be changed once they are imported. Export of configurations should also be possible in separate files or as a new file in a HCV.

In addition to the file based exchange of modules, configurations can be pasted from clipboard. The pasted information is interpreted as a list of module part numbers. All part numbers contained in the pasted data get activated, all others get deactivated. If part numbers are not available in current harness, they are ignored and a warning is displayed.

E/E Browser doesn’t change any approval relevant files in a HCV. As result there will be two configuration files in a HCV. One is provided by a source system (PDM, Connect …) and approval relevant. The other one contains user created configurations. Depending on which persistence option is used there will be following files in HCV:

|  |  |  |  |
| --- | --- | --- | --- |
| Persistence option | File | Not editable | Editable |
| independent from option | Harness data (document.kbl) | + | - |
| independent from option | Drawing (SVG) | + | - |
| independent from option | Bill of material (index.xml) | + | - |
| independent from option | RedLiningInformation.xml | - | + |
| independent from option | Memolist.xml | - | + |
| Configuration persistence with XML | UserConfiguration.xml | - | + |

## Conclusion

The E/E Browser will support following operations to handle configurations:

* Open HCV with included configurations (in KBL file **and** UserConfiguration.xml file)
* Import configurations from separate file (only UserConfiguration.xml)
* Viewing configurations, Configurations are identified by a name or similar
* View configurations from KBL as read-only
* Assigning a configuration to the current module selection
* Create, modify, rename and delete user defined configuration
* Copy configurations
* Export configurations to separate file (xml format)
* Add user defined configuration when saving a HCV
* Compare configurations in 2 steps:
  + Compare just modules in configuration without used parts
  + Compare complete configuration with all used parts
* Configurations can be pasted from clipboard
  + Pasted information is interpreted as list of module part numbers
  + Contained part numbers are active, all others are not active
  + Part numbers that are not available in the current harness are ignored

# User interface proposal

Most of the mentioned operations will be available through an extension the module configuration manager.

## Ribbon menu changes

The ribbon menu already contains a group for module configuration buttons. There will be a paste from clipboard button in this group. This button will apply the clipboard information to the current module configuration. After this the configuration is automatically applied to the drawing and the data table.



Figure : New ribbon menu entry

### Conclusion

* Add button to paste module configuration from clipboard
* Pasted data is interpreted as list of module part numbers, especially copied from excel or text editor
* Contained module part numbers are selected, all others deselected
* Part numbers that are contained in clipboard but not in current harness are ignored
* Paste button is disabled when:
  + Clipboard is empty
  + Clipboard data is not compatible (see office: disabled when file)

## Changes to modules pane

The modules pane is used for rapid change of module configuration. To allow user rapid change between saved configurations a dropdown control is added on the top of modules pane. The dropdown list contains all saved module configurations. If there are no saved configurations, the list is empty. Picking a module configuration from the dropdown will not automatically apply the configuration to the drawing and to the data tables. It just changes the checked modules in the module configuration tree.

The button to the right of the dropdown replaces the “apply” button. This button applies the current configuration to the drawing and the data tables. The button is enabled after the user changed the current configuration. This happens when a checkbox of a module is pressed or a module configuration is picked from the dropdown.

If user starts editing the current configuration by clicking the module checkboxes, the selected module configuration in the dropdown is switched to a custom module configuration. This custom configuration is identified by the name “(custom)”. This module configuration name is reserved in E/E Brower and cannot be used to save a module configuration.

If user switches from custom configuration to any other configuration the current state is saved in the custom configuration. This makes it possible to switch back to the last user edited configuration. The custom configuration is replaced when user presses a module checkbox.

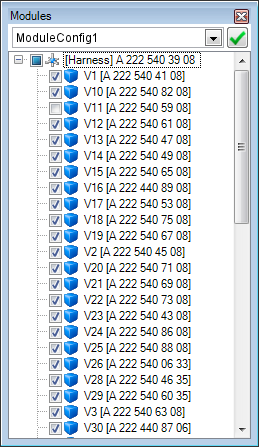


Figure : New modules pane

### Conclusion

* Additional dropdown shows the current configuration (not neccessarely the one that is applyed to drawing and data table)
  + Choosing a configuration from dropdown changes the selected modules
  + It is not automatically applied to drawing and data table, user maybe just wants to have a quick look on a configuration without applying it
* Smaller apply button in the right of the dropdown
  + Apply is symbolized with icon
  + Icon in screenshot is a proposal, could be changed if there is a better one
  + Apply button should still be highlighted if not pressed (like the red border of the current apply button)
  + Apply button is disabled after it is pressed
  + It is enabled again if:
    - A module is activated/deactivate a module by clicking it’s checkbox
    - A configuration is picked from dropdown
* Handling of “(custom)” configuration
  + “(custom)” is a reserved name for E/E Browser, user cannot use it for any other configuration
  + Every time a module is activated/deactivate a module by clicking it’s checkbox, dropdown changes to “(custom)”
  + “(costum)” configuration is the only configuration that can be changed in the modules pane
    - It is changed every time a user activates/deactivates a module
    - It is not changed when user picks a configuration from dropdown
    - 🡪 user can switch between “(custom)” configuration (the last user modified configuration) and any other configuration without causing “(custom)” configuration to change

## Changes in module configuration manager

The module configuration manager will be extended with multiple columns for module selection. Every column represents a module configuration. The module configurations can be renamed, deleted, created, copied and modified. Also there is the possibility to paste configuration from clipboard. The modules in clipboard are a activated in the concerning configuration. Greyed out columns are read only. They cannot be deleted, renamed or modified.

There is always one “active” configuration. This is the configuration that is set in the modules pane. The active configuration can be changed in the context menu. When the dialog is closed, the active configuration is applied to the modules pane, the drawing and the data tables. Alternately this can be explicit trigger by pressing the apply button. It is also possible to remove the apply button completely from the dialog. The active configuration is then automatically applied when the dialog is closed.

All actions can be triggered by a context menu and a menu button. The context menu is more enables a faster usability, but it is hidden and not found by all users. Due to agreed design guidelines for E/E-browser all context menu actions should be available with alternative menu.

Import or export of the module configuration is done by using the buttons on the left bottom of the dialog. Before export file dialog appears, the user is asked if he wants to export all configurations, user defined configurations or the active configuration. Imported configurations will not override existing ones, even if they have the same identifier. If the identifier already exists, it is extended with a suffix (e.g. “\_1”). Both will handle xml configuration files as they are defined in chapter 2.1.

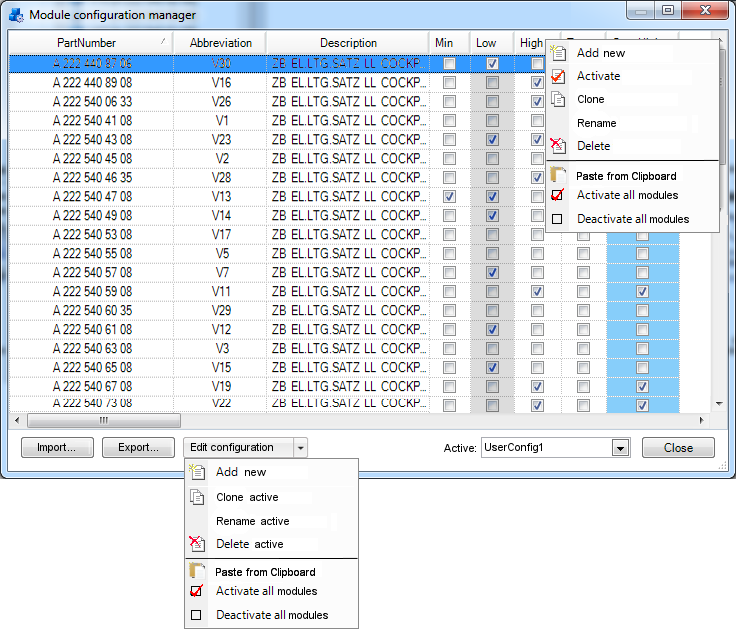


Figure : New module configuration manager

### Conclusion

* Import button
  + Importing configurations with already existing identifier
    - Dialog to question the user if he wants to
      * Override existing
        + Configurations with same identifier are overridden
        + If existing configuration is read-only, the imported configuration is extended with a suffix (“\_1”)
      * Don’t override
        + Configurations with existing identifier are ignored
* Export button
  + Before export file dialog appears, user is asked with dialog (e.g. with radio button) if he wants to
    - Export all configurations
    - Export user defined configurations
    - Export active configuration
  + The “(custom)” configuration is as well exported
    - in all three options, e.g.
      * export all
      * export user defined
      * export active if it is active
* A single row for each module
* Columns for
  + Module part number
  + Abbrevation
  + Description
* Additional columns for each configuration, including a “(custom)” configuration
* “(custom)” configuration is always in the fourth column (not displayed in screenshot)
* There are three types of configurations:
  + Read-only, shown with grey background
  + User defined
  + “(custom)” configuration
* Context menu on the header of table, context is the clicked column (clicked configuration)
  + Actions independent from clicked column (clicked configuration):
    - “Add new”:
      * adds a new configuration
      * name is “new\_module” or similar (suffix if name already exists)
      * no module selected
  + Read-only configurations:
    - Clone
      * Creates clone of existing configuration
      * Name suffix “\_copy” or similar
    - Activate
    - Following menu items are disabled:
      * rename
      * Delete
      * Paste from clipboard
      * Activate all
      * Deactivate all
  + User defined configurations:
    - Clone
    - Activate
    - Rename
    - Delete
      * If configuration is active, the active configuration is switched to “(custom)” configuration
    - Activate all modules
    - Deactivate all modules
    - Paste from clipboard:
      * activates modules whose part numbers are in clipboard
  + “(custom)” configuration
    - Clone
    - Activate
    - Activate all modules
    - Deactivate all modules
    - Paste from clipboard
    - Disabled actions are:
      * Delete
      * Rename
* Active configuration is displayed with light blue background
  + Could be changed by context menu
  + Is applied by pressing closing the dialog
* Alternative appliance via dropdown button “Edit configuration”:
  + Most actions are applied to the active configuration
  + Independent from active configuration (Always enabled):
    - “Add new”
  + Disabled when active configuration is read-only:
    - “Rename”
    - “Delete”
    - “Paste from clipboard”
    - “activate all”
    - “deactivate all”
  + Disabled when active configuration is the “(custom)” configuration
    - “Rename”
    - “Delete”
  + When active configuration is a user defined configuration all actions are enabled
* Active configuration can alternatively set by a drop down box in the lower right corner
* Changing active configuration with context menu will also change the dropdown box with the active configuration, so both are synchrony all the time.

## Changes in compare dialog

Comparing two configurations is one of the main advantages of saving configurations. It is planned to compare the harness in two different granularities. First is to simply compare the module list of configurations and the second is a complete comparison of all parts (contained in active modules). The last one is more like the current compare. And so it is integrated in the compare dialog.

There will be two new dropdown lists beneath the document selection to choose the configuration that will be compared. In each dropdown there are the configurations of the concerning harness documents. The active document includes a “(custom)” configuration in any case. In the compare document a “(custom)” configuration exists only if it is contained in the compare file or if the compare document is already open.

There is an additional entry in the configuration dropdowns called “(entire harness)”. This is the compare of the complete harness, like it is today.

The comparison of configurations is also possible between KBL files, if they contain configuration information. If the compare document is a HCV file, we need to merge the xml persistent configurations with the KBL persistent configurations to fill in the configuration dropdown, especially when compare document is not open in E/E Browser.

It is also possible to compare a document with itself. So the user can compare different configurations of the currently opened document. There is no need to forbid the comparison between a file and itself where both have the same configuration. This will result in no differences.

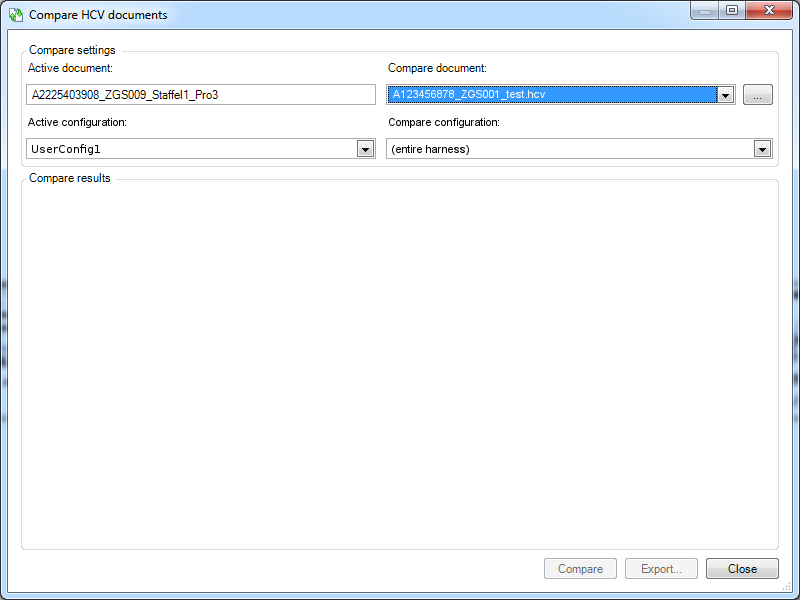


Figure : Changes in compare dialog

### Conclusion

* New dropdowns to choose configuration, the will contain:
  + All configurations of the concerning file
    - If hcv, from kbl file and from xml file
    - “(custom)” configuration in active document
    - “(custom)” configuration in compare document
      * if it is open in E/E Browser
      * if it is hcv file with a persistent “(custom)” configuration
  + additional “(entire harness)” entry to compare whole harness, like it is today
  + “(entire harness)” is default value for both configuration dropdowns
  + If there are no configurations (e.g. a not opened KBL file or not opened HCV without configurations)
    - “(entire harness)” is the value of configuration dropdown and cannot be changed (e.g. it is the only entry)
    - OR dropdown is disabled
* Compare with configuration is possible between:
  + Active document (KBL or HCV) and open hcv
  + Active document (KBL or HCV) and imported kbl
  + Active document (KBL or HCV) and not opened hcv
  + Active document (KBL or HCV) and not imported hcv
  + Important: Between active document and itself, to enable user to compare two configurations of active file

# To define

## Quick comparison of configurations

This could maybe be included in the module configuration manager with an additional button leading to a new dialog. Maybe this dialog can be based on the standard comparison dialog.

Alternatively we could integrate a quick configuration comparison in the comparison dialog.