ADDS

Best Practices

Version	Author	Date	Content (Detailed Modification Protocol)
1.0	Franz Lohberger, Visu-IT!	20.05.2010	First Version

Contents

Code Sharing Adapter

The **C**ode **S**haring **Ad**ap**ter** (**CsAdp**) can be used to define/specify the interface of the own functions/containers to the outside-world on project level. Typically, the CsAdp is used in code sharing projects with OEM and supplier. In this case, the 'own' ADD functions/containers are seen as a encapsulated system with interfaces to the outside world. The interfaces are specified via the CsAdp container.
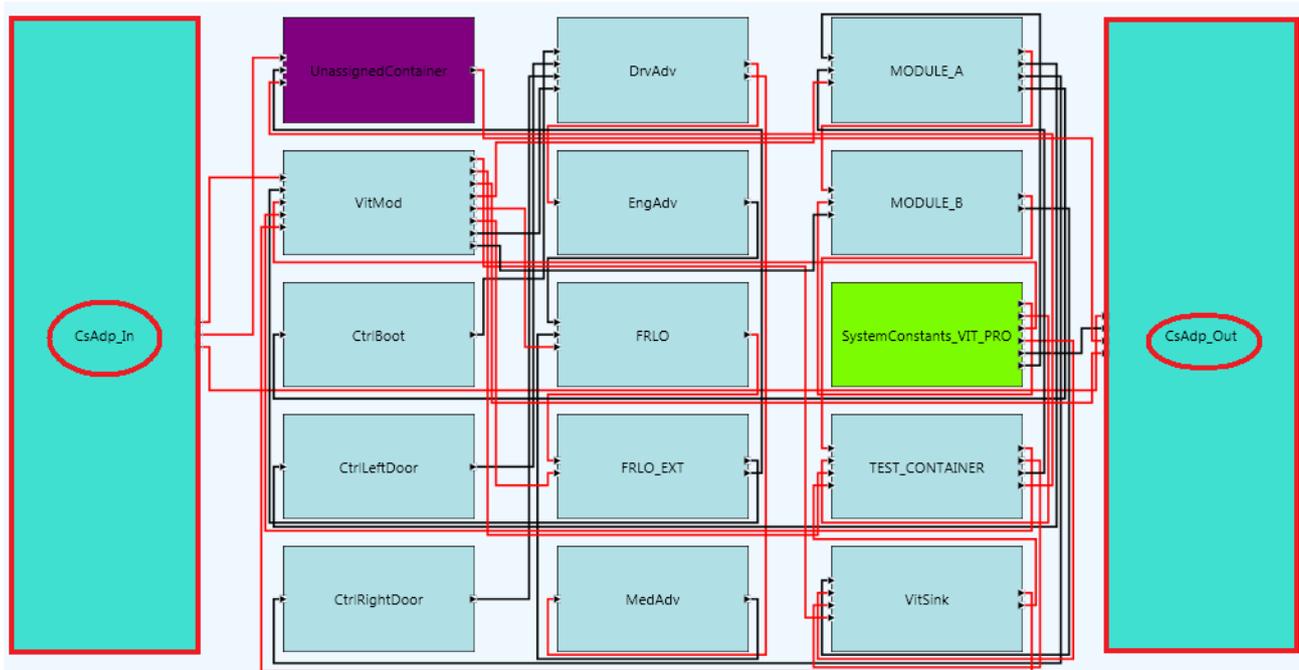
Configuration
The CsAdp is a special container in ADD (with a special *<Container-Type>*). This container can't be manually created by the user but must be pre-configured.

Specifics:
- There must only be **one** CsAdp.
  The CsAdp can be used for several projects with several suppliers. This is possible by using "CsAdp-custom tags". For different suppliers, different custom tags are used. The CsAdp can be filtered according to custom tags. Different versions / revisions / variants of the CsAdp container are possible
- Only online data are allowed in the CsAdp container.
  Calibration data are not applicable for the code sharing interface
- The CsAdp container does only have INPUT's and OUPUT's (but no LOCAL's)

CsAdp – Exchange
It is possible to:
- exchange the data of the CsAdp container via DDX and MDX (in the new format V1.1!)
- identify the actually used interface data automatically
- export the actually used interface data to DDS and e.g. generate c-/h-file, System-Constants Konfiguration file, etc.
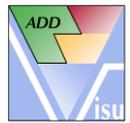
Handling of System Constants

There is one central special ADD container where are System Constants are managed -> SysDefConf.

Configuration
The SysDefConf is a special container in ADD (with a special <Container-Type>). This container can't be manually created by the user but must be pre-configured.

Specifics:
- All System Constants must be defined as OUTPUT data in SysDefConf.
  Note: The SysDefConf contains **all** versions/revisions of a System Constant
- There is the special role "SC_ADMIN". Only users with this role can edit the SysDefConf container and create new System Constants.
- System Constants are used as INPUT's in all other containers. Consequently, a System Constant:
  o must never be LOCAL
  o must not be used as OUTPUT in other containers except SysDefConf

ADD
On project level (in the dialog "Projects | System Constants"), ADD can automatically detect which system constants are used in the current project. These system constants can then be configured/defined on project level (-> project specific information).

DDS
The actually used system constants on project level can be dedicated imported to DDS and further processed (e.g. generation of a separate System Contstants *.h file)
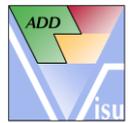
Variant Coding (data variants)

The VarCod container can be used to centrally define the variant selectors and variant criteria.

Configuration
The VarCod is a special container in ADD (with a special *<Container-Type>*). This container can't be manually created by the user but must be pre-configured.

Specifics:
- All variant selectors must be defined as OUTPUT data in VarCod.
  Note: The VarCod contains **all** versions/revisions of a variant selector
- The variant coding information (variant name, variant criteria, etc.) can only be managed inside the VarCod container.
- There is the special role "SD_ADMIN". Only users with this role can edit the VarCod container and create new variant selectors.

For more detailed information please see the document
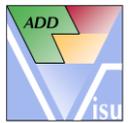http://www.visu-it.de/ADD/ADD_Info/Variantenkodierung.pdf
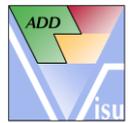
Obsolete conversion types

In previous versions of ADD, the following conversion types were provided:
- BinPoint
- Float
- Ascii

Problem: These conversion types can't be mapped to ASAM/Autosar COMPU-METHOD-types. Thus, ADD will no longer provide these types. For users which do already use these types we strongly recommend to replace them with ASAM/Autosar compliant ones. Please contact Visu-IT! in order to get support how to replace the types.