

Data Declaration System

What's new in Version 4.2.0?

August 01st 2005

DDS Contact

email: dds@visu-it.de

Internet: <http://www.visu-it.de/dds>



© Copyright 2005
Visual Information Technologies GmbH
An der Schergenbreite 1
93059 Regensburg

New Features, Major Improvements

Enhanced support of ODX-FLASH (including generic editor)

DDS covers one special part of the overall ODX (Open Diagnostic Data Exchange) contents: the ODX FLASH part.

In this version, DDS provides:

- a) a generic ODX-FLASH editor
Similar to an XML editor, the ODX-FLASH editor can create, modify and delete arbitrary XML elements and attributes. Special semantic relationships between elements of the ODX-FLASH contents are supported via a new kind of smart variable (for detailed information please see the SysConf online help)
- b) an export filter, generating the ODX-FLASH file (*.odx)
The ODX-FLASH export is provided as a prototype filter without any documentation.

Minor Improvements / Correction of Bugs

Delete configuration unit via command line and GUI

Now, it is possible to delete a configuration unit (GRL file)

- a) directly in the DDS editor and/or
- b) via command line (by launching the CompareMerge filter in "DeleteConfUnits" mode)

Crash with 'SaveAs' while working on eval.-define (#29)

The following bug is fixed now:

DDS crashes with 'Save As' button while working on evaluable #define:

- Create an evaluable #define named a
 - Enter a formula like '10 + 2'
 - Save your entity
 - Now change the name of this entity to b (without saving it!)
 - Change the formula to '10 + a'
 - Perform a Save as operation of the entity
- > DDS crashes, and the editor closed without any 'Save' operation inside the database

Creation of an array based on a define which doesn't yet exist (#30)

The following problem is fixed:

- Creation of an array based on a #define which does not yet exist:
 - Create a parameter named 'a'.
 - Go to the array size attribute then click on the 'right arrow' at the entity editor.
 - Then a window will be displayed where you can write the name of the #define you want to create. Type a name 'b'.
- > Then a warning message will be displayed: "You can't add a child to this node".

'!' character must not be allowed in calFunction names (#31)

Although the '.' character is supported also for functions in ASAP2, it will be handled in a more restricted way in DDS, since:

- there is no big benefit of supporting the '.' inside function names
 - using the '.' character causes confusions and 'double' naming when the combination of function and subfunction names equals to another function name containing a '.' character
- Note: This problem also arises when extending the 'PartialExportViaFunction' functionality on sub-functions!

Display a tag Hidden when selecting a hidden calFunction (#37)

When selecting a calFunction for a data (via Sel-list tab + DefFunction combo box) the selection list does now display when a calFunction is defined as hidden.

DDS crashes when closing an minimized project (#43)

The following bug is fixed now:

DDS crashes when closing an iconized (minized) project:

- *Open a project inside DDS editor. Iconize this project (press the minimize icon).*
- *Open another project inside DDS editor.*
- *Close project 2 by closing the window ('x' icon).*
- *Close project 1 by clicking directly on the 'x' icon.*
- *DDS crashes: 'Memory cannot be read'.*

DataPool Check: 'NoRepair'-mode must not change the DB "layout" (#52)

The DataPool Check does no longer change the contents/layout of GRL files when launched in 'NoRepair'-mode. That means that the GRL will not be modified in that mode.

Import of array during ASAP2 import (#56)

Today, the ASAP2 standard supports arrays where each element can have its own attributes (because each array member is described separately in the ASAP2 file). DDS has been improved in order to correctly import such arrays (array-members) into the DDS database.

All possible attributes of an array are taken into account with a general rule:

- If all array elements have the same attribute then move it on the parent entity level
- In other cases, just assign this attribute on each sub-entity

Note: There is a difference between the "display name" attribute and others, because in case of the "display name", it would be necessary to find a suitable display name pattern for an entity, which is not so obvious. The "display name" is not supported in this version.

Default directory for export filters (#58)

When exporting some files with a DDS filter (Source Export, ASAP2 export...), there is now the possibility to export these files in a default directory.

For ExpA2, ExpA_2D, ExpMSR and ExpRep the default directory is "OUTPUT" (there is no need for further subdirectories, since each output file has a different suffix).

For ExpSrc, the default directory is "OUTPUT\SRC".

SourceSections: GUI not available with non-calib. data (#60)

The following problem is fixed:

It is not possible today to assign Source Sections to non-calibration data.

ExpSrc: MemberList is missing, different exports (#61)

The following problem is fixed:

Scenario: A database has C files where the flag "keep order in database" is set. However, the member lists for these c Files are missing...

Perform a source export, then close DDS. Open the db again, perform another source export: we get different source files even if the source export is done with Order mode set to 'None'.

Even if DDS must only be deterministic when working on a 'correct' database regarding semantic layer, a warning must be raised saying that the source export order might not be deterministic!

DPC: structure members referenced but do not exist (#66)

The DataPool check detects and fixes those invalid structure members.

Reduction of messages in command line (#24)

There is a new filter option "OnlyErrorsOnStderr" which reduces the amount of messages emitted to 'stderr' during a filter run in command line.

In command line, only messages of the type 'Error' and 'InternalError' are emitted to 'stderr' if the option "OnlyErrorsOnStderr" is set to 'true'. All other messages are skipped.

This option only influences the amount of messages at 'stderr', it does neither influence the log-window in GenEdit nor the log-file generated via the filter option 'ErrorFile'

Source Export: Sequence of anonymous structures in declaration file (#70)

When a structure type is used from another structure type, the source export ensures that this 'nested' structure type is declared before its usage within the 'parent' structure type.

This works now also for anonymous structures.

ODX/XML: Smart Variables for init values (#71)

Requirement: With the current smart variable 'useDefine' it is possible to reference a simple 'cDefine' entity and use its value for further processing. However, in case of ODX, there is the need for the following extensions:

- It should be possible to resolve the referenced cDefine value. That means, when the cDefine is of the type 'token string', its value must be checked whether there are other cDefines referenced. If yes, the value of these cDefines should be resolved again. The final value of the cDefine will contain a 'flattened' token string where all referenced cDefines are solved.
- It should be possible to convert the (string-)value of a token string cDefine into a hexadecimal ASCII format, e.g. "111" -> "31 31 31"

Solution: A new smart variable of kind 'cDefine' has been introduced:

```
smartVariable 'cDefine - resolve & convert to hex-ASCII' {  
    predicate = "ResolveDefine(); ConvertToHexAscii()";  
    kind = cDefine;  
}
```

This smart variable contains 2 commands in the 'predicate' attribute:

- ResolveDefine()
Works like a pre-processor resolving all cDefines within the value string of a token-string cDefine. Does not operate on string cDefines, but only on token-string cDefines!
- ConvertToHexAscii()
Converts all characters of a string into their hexadecimal ASCII representation. The ASCII codes are separated with blanks.

ODX/XML: Smart Variables for enumerations (#74)

Requirement: It should be possible to select only some predefined values for special 'xmlElement's.

Example: Inside ODX/XML it would be better for the user to select possible values inside a dropdown list rather than entering text strings

Solution: please see the SysConf online help

ODX/XML: Smart Variables for automatic ID generation (#75)

Requirement: ASAM group recommends generating ODX/XML ID's in a way that we can read the "absolute" path of an element. Therefore there should be a smart variable inside ODX/XML in order to automatically generate such ID's. The rule is pretty simple: Path and short names of ODX/XML elements.

Solution: There is the new smart variable 'XPath_AutoID' which can be used to automatically generate this ID value (for details, please see the SysConf online help)

Universal XML support (#76)

Similar to the ODX-FLASH functionality, DDS provides:

- a) a generic XML editor
The XML editor can create, modify and delete arbitrary XML elements and attributes. Special semantic relationships between elements are supported via a new kind of smart variable (for detailed information please see the SysConf online help)
- b) an export filter, generating arbitrary XML files (*.xml)

Make use of arbitrary C-labels (#77)

A "code symbol" is a C definition which has no real representation in DDS. That means, in DDS, only the name of the C definition is known (important information like data type, etc. is missing). Code symbols can be created/modified/deleted in SysConf and will be supplied with address information (physical address, size) during an ELF or IEEE import. Code Symbols can be used as supplier of address and size information for different purposes, e.g. within Memory Regions to specify the start address and/or the byte size of the respective entity (for detailed information, please see the SysConf online help)

XCP protocol V1.0 (#83)

This DDS version contains:

- the AML files/settings for the XCP V1.0 protocol
- a corresponding ifData-Template file

Limited support:

- only XCP on CAN
- only global settings (no individual settings for MEASUREMENT (e.g. DAQ_EVENT) and MEMORY_LAYOUT)

(Filter-)Option 'Project' in pre/post-processing (#84)

It was not possible to access the common filter option 'project' during pre/post-processing. The workaround was to change the bitmask of this option in the DDS.ini. However, this was not officially supported. This version of DDS supports this option (as \$Project\$) during pre/post-processing.

Support of online array elements with DServer (#89)

In case of online data, there are two kinds of (local) members possible:

1. array elements
2. onlineUnion members

The old DServer version always created onlineUnion members when creating a (local) member.

This version of DDS solves this problem:

-> When the name of the member, which has to be created, contains square brackets, an array member will be created (*). Otherwise, an onlineUnion member will be created (**).

Example:

' --- Visual Basic ---

```
...
set onl = DDSEntityMan.Create("online", "onlineArray")
success = onl.SetAttribute("elemType", "int", 0, "numberType")
set embed = onl.AddEmbedded("arraySize", "integer")
success = embed.AddAttribute("value", 7)
```

```
set m1 = onl.Create("online", "[0]") ' -> create array element at index '0'
```

...

(*) It is only possible to create an array member when the 'parent' definition has an array size!

(**) It is only possible to create an onlineUnion member when the 'parent' definition has no array size!