

Data Declaration System

What's new in Version 4.1.0?

February 17th 2005

New Features, Major Improvements

Init of RAM definitions

Until version 4.0.1, online definitions and structure instances that were located in RAM couldn't be initialized. Extension in version 4.1.0: All kinds of definitions which can have init values when located in a ROM region can also get init-values when located in a RAM region.

Reprogramming & Diagnostic filter (ODX FLASH)

DDS is now able to generate the Flashing part of the ODX standard (version MCD 2D v2.0). The ODX Flashing export is provided as a prototype filter without any documentation.

CompareMerge, Consistency & Check

Compare/Merge

Function/Sub function

Improvements:

- smooth merging of functions and subfunctions
 - detection and deletion of undefined entities
 - detection and deletion of unreferenced (empty) entities, e.g. 'empty' calFunctions
- Also possible within the DDS editor via CTRL+F and the flag 'only Unused', or via the context menu command "Display Unused Entities"

Global Correction Groups

The DDS check is extended, so that superfluous references from a Global Correction Group are detected and removed.

Address Info

Improvement of the 'Compare' functionality: The option 'without addresses' is combined with 'without ByteSize'.

Data Pool Check filter improvement

Until version 4.0.1, the data pool check had 3 different modes. Now, the check has only two modes:

1st mode: Show all Warnings & Errors in the database without modifying them.

2nd mode: Same as first one, plus an automatic correction of errors/warnings when possible.

Missing definitions improvements

DDS emits an error in case of undefined definitions. However, the detection only works for global definitions. The checking has been extended in order to detect also local undefined definitions.

Source Export: Extension of Source Section concept

Introduction of "Source Section Variables"

DDS is now able to dynamically create arbitrary C code fragments which are **dependent** on the current definition. Due to this extension, it will be possible to provide a small set of customer specific code fragment templates in DDS which can be assigned to the definitions.

For this purpose, the 'beginSection' and 'endSection' attributes of the Source Sections can contain the following "Source Section Variables" which are processed during the source export:

Variable	Description
<code>\$this.Name\$</code>	Name of the current definition Applicable for the following entity/definition types: online, parameter, axis, map, cBase (non-calibration), cDefine, pointer
<code>\$this.ElemType.Name\$</code>	Name of the ElemType (dataType) of the current definition Applicable for the following entity/definition types: online, parameter, axis, map
<code>\$Axes.X.Name\$</code>	Name of the first axis. Only applicable for maps.
<code>\$Axes.Y.Name\$</code>	Name of the second axis. Only applicable for maps.
<code>\$Axes.X.AxisSize\$</code>	Axis-size of the first axis. Only applicable for maps.
<code>\$Axes.Y.AxisSize\$</code>	Axis-size of the second axis. Only applicable for maps.
<code>\$Axes.X.Online.Name\$</code>	Name of the online variable which is used/referenced from the first axis. Only applicable for maps.
<code>\$Axes.Y.Online.Name\$</code>	Name of the online variable which is used/referenced from the second axis. Only applicable for maps.
<code>\$Axes.X.ElemType.Name\$</code>	Name of the ElemType (dataType) of the first axis. Only applicable for maps.
<code>\$Axes.Y.ElemType.Name\$</code>	Name of the ElemType (dataType) of the second axis. Only applicable for maps.
<code>\$Axes.X.Online.ElemType.Name\$</code>	Name of the ElemType (dataType) of the online variable which is used/referenced from the first axis. Only applicable for maps.
<code>\$Axes.Y.Online.ElemType.Name\$</code>	Name of the ElemType (dataType) of the online variable which is used/referenced from the second axis. Only applicable for maps.
<code>\$this.AxisSize\$</code>	Axis-size of current definition Only applicable for axes.
<code>\$this.MappingScheme.Name\$</code>	Name of the mapping scheme (representation model) of the current definition. Applicable for the following entity/definition types: online, parameter, axis, map

For further details please see the DDS online help.

ASAP2: Export Improvements

Export of hidden functions

In general, it is not possible to modify the hidden flag of a subfunction when the parent function is hidden (because the hidden flag is greyed). This behaviour can be bypassed when setting the hidden flag of the subfunction before setting the hidden flag of the parent function.

New handling of "hidden flag inheritance":

1. Only an active hidden-flag (of value 'true') will be inherited. There is no inheritance when the flag is set to 'false'.
2. The hidden flag must be inherited in readOnly mode in any case, even when subfunctions have already an individual hidden flag setting.
3. When a subfunction has the hidden flag explicitly set to 'false', the hidden flag will be explicitly removed (from the database). When the hidden flag of the parent function is reset to 'false', the subfunction is implicitly – and no longer explicitly - set to not hidden.
4. When a subfunction has the hidden flag set to 'true', the inherited flag has a higher priority. Although the individual flag is redundant and superfluous in this case, it must not be removed (e.g. from the data pool check) since the flag is needed again when the inheritance is no longer active (when the hidden flag of the parent is removed).

Support of empty container

In ASAP2 it is possible and valid to declare and maintain container definitions like functions and groups which have no contents but are empty. However, these empty container definitions might cause problems in measurement and/or calibration tools. The DDS ASAP2 import and export filters are enhanced in order to control whether empty containers should be imported/exported or not.

ASAP2 Import Improvements

Import of Memory Segment

There are different modes when importing ASAP2 MEMORY_SEGMENT / MEMORY_LAYOUT definitions. They can be controlled via the filter option "MemRegionMode"...

Online Help

Rework of DDS online help:

- harmonization of different help contents
- addition of further topics
- conversion to HTML based *.chm files

Minor Improvements / Correction of Bugs

TIS 814 (CVX import)

1. Support Axis/Maps initialization for string range conversions
In previous versions, the import failed when using string conversions for axes and maps
2. Correction of bug in array initialization when there is no "specific attribute" for an array element
In previous versions, the import failed, when array elements which have no individual attributes should get an init-value.

TIS 872 (INTERN & EXTERN for ExpA2)

...the smart variable "memSegmentAttribute" generates the enumerator values "EXTERN" and "INTERN" instead of "0" and "1".

TIS 884 (applicableSmartVariable)

...fix of SysConf problem when using simple smartVariables like dbAddressOf or useDefine in the instance tree without declaring them as applicableSmartVariable's in the type declaration first.

ASAP2 Import (EDICMobil)

...fix of ASAP2 import problem.

ASAP2 Parser

...the ASAP2 Parser should optimize superfluous empty AML declarations like:

```
"RMBA" taggedstruct { }; * and  
"BREAK_COMMUNICATION" struct { };
```

TIS 854 (after IEEE-Import a fatal error occurs during SAM2000-Export)

...fix of a very rarely SAM2000 export problem

SysConf (generic IF_DATA editor)

Improvements/fixes when

- switching between raw and smart values
- adding new SOURCE and RASTER blocks
- extending the handling when switching between raw and smart values

TIS 842 (Pre-/Postprocessing in command line)

The filter options pre-/postprocessing are now also supported in command line mode.

TIS 880 (ASAP2 export: Lower-/Upper-Limits)

The lower- and upper-limits of an instrumentation entity are now correctly exported to ASAP2 when:

- the entity has no own limits
- the entity uses a table conversion which has a negative resolution

IMES 11 (Source Export, brackets for map 8x1)

...fix of Source Export problem when working with maps of length '1'

IMES 15 (Address Import: MAP where X axis size is set to 1)

...fix of address import problem

IMES 8 (Wrong display name in empty database)

...minor change in empty database

IMES 9 (ASAP2 export failure: Memory Region Visibility)

When a memRegion is used as 'definedPage' but set to 'invisible for ASAP2', then no real error message occurred in the previous versions. This is fixed now.

IMES 16 (BYTE_ORDER keyword for ExpA2 (only for EDIC MOBILE))

There is a new filter option "SupportEdicMobileV3". If this option is active, the filter exports MSB_FIRST/MSB_LAST instead of LITTLE_ENDIAN/BIG_ENDIAN.

Note: This option is only taken into consideration when the ASAP2 export filter runs in EdicMobile-mode.

IMES 17 (Individual data types in source export of the DDS)

Basic idea is to have a differentiation between "code-relevant" typedefs and "non-code-relevant" typedefs. "non-code-relevant" typedefs are used within DDS only, in order to set additional attributes like conversion, formatString, physicalUnits, etc. (this concept is based on the representation model '20_ME').

IMES 7 (COM-API enhancement: Recover Byte-Size & Address attributes)

It is possible to access to the following attributes with COM-API: `byteSize`, `address`

IMES 10 (Display Names enhancement with arrays)

The following extensions are implemented:

1. Display name (pattern) inheritance at entity-level
2. Possibility to have 1 based arrays
-> For the new 1-based arrays, the modifier '1' will be provided. This modifier can be used in conjunction with the pattern '%x', '%y', '%z', '%i' and '%j'.

IMES 14 (Wrong alignment calculated inside structures at address import)

Goal is to detect whether the size mismatch comes from alignment issues or not.

If the mismatch is caused because of alignment, the address import emits another warning/information message (with an own message id). This message can be disabled/hidden from the user without missing 'real' size mismatches.

IMES 22 (Display DDS version within the "responsible" splash windows)

...displays DDS version within the "responsible" splash window

Address import without initial reset of addresses

The existing filter option "ResetAddressesOnly" is enhanced with an additional mode: "Perform address import without initial reset of addresses"

Reset of addresses without valid locator file

The address import filters must not check the existence of a valid 'i3e' or 'elf' file when the filters run in 'Reset addresses only' mode.

IMES 32 (Default value for table conversion)

The ASAP2 attribute "DEFAULT_VALUE" is now supported in DDS.

Support of LOC_MEASUREMENT

The ASAP2 attribute "LOC_MEASUREMENT" is now supported in DDS.

Address-Offset file

DDS provides a functionality to optionally create an "Address-Offset" file which contains the names of the ASAP2 labels, their root instances and their address offsets. The value of the address offset is defined in the following way:

- global definitions have no offset, consequently the values must be set to 0
- local definitions like e.g. structure members, array members, etc. should get their (address) offset related to the global (root) instance

Extension: The ASAP2 export filter provides an additional filter option "CreateAddressOffsetFile". If this option is set, the export filter creates an additional ASCII file with the additional suffix "*.AddressOffset.txt".

For further details please see the DDS online help.

Address-Import Prefix

DDS provides a functionality to optionally support a prefix for labels in the ELF file.

Scenario:

- 1) In DDS, there exists a label "myParameter1"
- 2) However, for some reasons, in the C code the label contains a prefix, e.g. "_myParameter1".
- 3) When performing an address import filter run the DDS label will not get an address, since the label can't be found in the locator file.

Extension: The address import filter "Import ELF" is enhanced with an additional filter option "ExternalPrefix".

For further details please see the DDS online help.