

Data Declaration System

Version 23.2

Release

28 July 2023

DDS Contact

email: hotline@visu-it.com

Internet: visu-it.de/products/dds/



© Copyright 2023
Visual Information Technologies GmbH
An der Schergenbreite 1
93059 Regensburg

Table of Content

1	KNOWN DEFECTS AND RESTRICTIONS OF VERSION	3
1.1	OVERALL DDS SYSTEM.....	3
1.2	EDITOR	4
1.3	DATA POOL LIBRARY (DPLIB).....	6
1.4	DDS CHECK AND SEMANTIC LAYER	8
1.5	ASAP2 IMPORT.....	9
1.6	ADDRESS IMPORT (I3E-695, ELF)	11
1.7	AUTOMOTIVE DATA DICTIONARY IMPORT (ADD-IMPORT)	14
1.8	CVX IMPORT (CALIBRATION VALUES).....	14
1.9	DDX IMPORT	14
1.10	SOURCE EXPORT	15
1.11	ASAP2 EXPORT	17
1.12	DDX EXPORT.....	19
1.13	ASCET EXPORT	19
1.14	ASCET IMPORT	19
2	'NOT SUPPORTED' OR OBSOLETE FILTERS AND COMPONENTS	20
2.1	NOT SUPPORTED.....	20
2.2	OBSOLETE	20
2.2.1	<i>Import Configuration Unit (GRL) filter</i>	<i>20</i>
2.2.2	<i>Report Generator</i>	<i>21</i>
2.2.3	<i>Export SAM2000 filter</i>	<i>21</i>
3	SUPPORTED COMPILERS (ADDRESS-IMPORT)	22
3.1	IEEE-695.....	22
3.2	ELF – DWARF 1.0.....	22
3.3	ELF – DWARF 2.0.....	22
3.4	ELF – DWARF 3.0.....	24
3.5	ELF – DWARF 4.0.....	24

1 Known Defects and Restrictions of Version

The following provides a list of known defects and restrictions of DDS V23.2. Productive use is anyhow possible.

1.1 Overall DDS system

Defect Tracking Number	Title	Comment / Explanation
	Limited support of DD proxy concept	The following restrictions of the DD proxy concept are currently known: 1.) Manual DD proxy assignment does not work correctly when assigning a DD proxy to an array-element of an array which is a member of structure
	'InstrEntity'-types are not fully supported	The entity types 'parameter', 'online', 'axis' and 'map' can occur both as (C-)instances and as (C-)types in DDS. The instances are managed within the 'Data' node of the entity browser of the DDS editor. The types are managed within the 'Types' node of the entity browser of the DDS editor. Since these kinds of types are rarely used (usually, a numberType typedef is used instead) they are hidden in the entity browser by default. Moreover, these types are not fully supported in DDS. Especially when using nested/complex representation models (e.g. the map is represented as a struct containing also the axes values), the DDS filters might not operate correctly.

1.2 Editor

Defect Tracking Number	Title	Comment / Explanation
EHI 37333 (TIS 403)	Crossing from struct instances to struct types	When 'double clicking' a structure member instance in the right part of the DDS browser window, the editor displays the corresponding structure member type, and not the structure member instance, in detail.
EHI 39348	Managing corresponding forbidden combinations of variants	When a variant criterion value has a forbidden combination, the forbidden combination is only displayed once. <u>Example:</u> When the variant criterion value 'a' has a forbidden combination with the variant criterion value 'b', the forbidden combination is only displayed once in the DDS editor (within the property page of 'a'). The forbidden combination is not displayed within the property page of 'b' although from a logical point of view also the variant criterion value 'b' has the same forbidden combination.
EHI 39587 (TIS 354)	Corruption with "used by / Editor Sel.List" with the action "save as" / Refresh of the treeview	Bug is at 99% fixed. The only remaining problem is that the tree is not updated immediately, but only after closing and re-opening the editorList-item.
EHI 37339 (TIS 413)	Editor List: Update problem when changing the array dimension of a data	When the array dimension of a data (which is referenced within an Editor List item) is changed, the reference itself is not updated. Example: Defined an Editor List, e.g. 'abc' ; set reference to a map (one dimensional); save the database; change the array dimension of the map, e.g. 3; save the database; -> The Editor List references the array and not its members (which is not correct)!!!
	Concatenated #define for initialisation of ASCII parameter array (or non-calibration data array) cannot be entered and maintained in DDS Editor.	Such concatenated #define statements for initialisation are typically stable during a project's life, and mostly applied for instrumentation-related definitions (i.e. SAM consistency checks, see entities like sam_name, sam_ref_nr, ...). Normally the DDS support defines and maintains such definitions. Only the contents of the #define entities are intended to be changed by the user. When looking at the INIT page of these ASCII parameter arrays of non-calibration data arrays, only the first element of the concatenation is visible (can be confusing).
9909-09	Case dependent assignment of formats (%g / %G) and physical units (t / T) not possible. Same for identifier references (online→axis→map)...	Selection lists show attributes, which differ only in case (capital/small), but differentiation after assignment is not possible. Relevant in case of format (semantic difference of %g and %G). Effect related to MS-Windows restriction.
	Static default file (as C file, no header file) is not used.	The source file defaults in the active project settings allow to specify a static file, which is supposed to be assigned in

Defect Tracking Number	Title	Comment / Explanation
		<p>case of "persistence=static" (default is 'extern'). In the static case, no header file is allowed, and the "static" file should be used instead of the C file.</p> <p>This does not work. The regular behavior for 'extern' is applied.</p>
	<p>An explicitly assigned configuration unit must be saved within the 'Contain' panel, otherwise the configuration unit of the default rules is saved.</p>	<p>The default configuration unit assignment from the active project settings is used for any new definition, unless the user chooses another configuration unit explicitly.</p> <p>Restriction: the selection of another configuration unit must be saved (locally, button on right side of browser) before leaving the "contain" panel or the DDS Editor. If the user changes the panel, e.g. going to "calib" to enter another conversion or to "init" to enter initial values, before saving the first time, the default configuration unit will be used (false) instead of the user-selected configuration unit.</p> <p>This problem does not occur for the change of the configuration unit assignment with existing definitions.</p>

1.3 Data Pool Library (DpLib)

1.) Limited Warranty:

1a.) Any operation with **Split units** and **Update Units** is **excluded from the warranted DDS product features**, although the "Import Configuration Unit (GRL)" filter has some command line options supporting split units and update units, which are hidden in the user interface.

Such operations may or may not work. Use them on your own risk, but please avoid them in a productive environment. Any database with Split Units and Update Units is excluded from regular support and maintenance.

The Data Pool Library is not tested with Split Units and Update Units, and there are even some known defects (included in the list below). The reliable support of Split Units and Update Units would be a major implementation and testing task.

1b.) Any operation with **References to Read-Only Configuration Units** is **excluded from the warranted DDS product features**. Attempts to write to a definition, which is declared or referenced in a read-only configuration unit, fail and the DDS Editor may react in an instable manner.

Read-Only configuration units are safe when its definitions are not referenced in other parts of the DDS data pool and the purpose of "read-only" is to prevent the address import filters (I3E-695, ELF) from overwriting a fixed address.

Read-Only configuration units with global attributes only (conversion, physical units formats) are safe when the user does not try to write these global attributes. They may be referenced in other (writable) configuration units.

Defect Tracking Number	Title	Comment / Explanation
	Rename and delete of structure type elements –corruption in rare and complex situations cannot be excluded. Please control the result of the operation.	The "easy cases" of reproducible defect situations have been thoroughly analyzed and have been fixed. Please operate with care when renaming or deleting definitions, in particular elements of structure types with instances having individual attributes.
9908-10	SAVE AS/ Rename with referenced read-only obj.	Rare operation, but corruption occurs. Originally classified as editor problem.
9910-05	"change status" is sometimes incorrect after filter run	For "reading filters" like SAM export, it can happen that the "data base change indication (SAVE disk button)" is active, although nothing has been changed.
9911-21	Individual attributes of elements of instances of structure types remain even after their deletion. (no corruption, but confusing). Note: Individual attributes of array elements are immediately deleted when reducing the array size.	This could be interpreted as a "feature" (attributes are restored when their declaration returns), but in most cases it makes the data base more complex and error-prone. The DDS Check mode 'repair / advanced' option removes such unwanted attributes from the data pool.
	UPDATE units: any operation is unsafe	DPLIB not tested with the concept of update units. (update units are not a DDS V2.2 product feature). The feature has been removed from the user interface.
	SPLIT units: any operation is unsafe	DPLIB not tested with the concept of split units. (Split units are not a DDS V2.2 product feature). The feature has been removed from the user interface.

Defect Tracking Number	Title	Comment / Explanation
1559	Deletion of update units : corruption	DPLIB: If there is an entity in a configuration unit and a corresponding update unit is in effect, the removal of this entity results with an entity with all attributes removed instead of a removed entity. That means that entity is emptied but not deleted.
1530	Read Only Status incompatible with update units	DPLIB: If in an update unit an attribute with embedded entities is changed a 'unit is read only' exception is thrown saying that the original unit is read only (that is always the case for update units)

Remark: Defect symptoms of the Data Pool Library (DpLib) sometimes seem to be related to the DDS Editor. Therefore, some DpLib problems are registered as DDS Editor problems.

1.4 DDS Check and Semantic Layer

Defect Tracking Number	Title	Comment / Explanation
EHI 39146	Update of entities when referenced Entities are modified	When changing the array-dimension of the map-type, it will be inherited to the instance -> OK. However, it is not possible to edit/modify the array-elements. Furthermore, the source-export will not work properly in this case. Since working with map-/axis-/parameter-/online-types was never used in the past (it is even disabled per default), there might be further problems. Consequently (if these types should be really fully supported), it could be a major task and should therefore be planned in the next version.
EHI 39276	No attribute inheritance for (REF-)functions	Attribute inheritance is only supported for 'simple' attributes but not for lists. This means e.g., that although an online structure may has a REF-Function, all its members will not be added to this REF-function, since there is no attribute-inheritance. The user has to assign each member to the REF-function (It is possible to copy the REF-functions of the structure-instance to all its structure-members via CTRL-C and CTRL-V) This inconvenience also exists for arrays!
	'Token String' defines are not evaluated	Defines of the type <i>'Token String'</i> are not evaluated/checked in DDS. Thus, it may happen that the source export filter generates an invalid C-code when using 'Token String' defines.
TIS 354	It might happen that the database is temporary semantically inconsistent.	After some definition-related database operations like 'save_as', 'rename', 'delete' etc. the database might be temporary semantically inconsistent. These semantically inconsistencies can be fixed by starting the DDS check (in 'Advanced' mode)
0003-19	DDS Check reports 'No conversion allowed with FLOAT data type', but ASAP2 Import creates it and ASAP2 Export filters supports it. '-HideMessage L2230' recommended (on top of dds.gpp – then valid for all operations).	No conversion with FLOAT data type is supported with SAM Export. ASAP2 Export supports it. DDS Check reports an error for every FLOAT definition with a conversion (L2230) in order to indicate the missing support in SAM. This may be confusing for ASAP2 focused usage. In this case, it is recommended to hide message L2230 (enter as first line of dds.gpp – then this entry is not filter specific but valid for the complete operation of DDS).

Remark: Defects of the Semantic Layer can be also described with DDS filters, as the defect symptoms are often related to a particular DDS filter (Source Export, Address Imports, Description file exports).

1.5 ASAP2 Import

Defect Tracking Number	Title	Comment / Explanation
	Import of IF_DATA blocks and AML sections	There are some restrictions when importing IF_DATA blocks and AML sections: It is not possible to semantically import the values of a CCP IF_DATA definition into the DDS internal 'ccp' object. When importing a CCP IF_DATA definition, the values are stored in the normal ifData definitions but not in the specific 'ccp' definition.
	No default values for overwritten definitions	If the filter option "IgnoreProjectSettings" is set to false, an imported definition is initialized with the default values in the project settings. However, this is not the case when the definition already exists in the database and the filter option "Merge Mode" is set to "Overwrite". In this case, the default values are not taken into account.
	For FLOAT definitions COEFF conversion imported but not supported in DDS.	In ASAP2 the COMPU_METHOD is a mandatory attribute, also for definitions of type float. Typically, COEFF (Rat_Func) – Compu_Method is used with float definitions in ASAP2 and is also imported to DDS. This is incompatible with the semantic check which currently does not allow any conversion in conjunction with 'float' data type.
	Bit (re-locatable "bit", C166 kind) only supported as byte with fix bit mask. Problems may occur at I3E-695 Import.	ASAP2 cannot express the data type "bit". Existing bits of DDS are exported to ASAP2 as bytes with byte address and bit mask. ASAP2 Import will generate a definition of type "Byte" with a fixed bit mask. In case that the corresponding software object is a re-locatable bit, a subsequent I3E-695 import will not apply the hard-coded rules of the C166 bit addressable area and will not overwrite the fixed bitmask. A type mismatch error should occur.
9912-04	Unsigned data type for virtual axes not always appropriate.	The ASAP2 standard does not have any data types with virtual axis (FIX_AXIS), but the DDS data model requires a data type. Therefore, there is room for interpretation. In case of negative FIX_AXIS values of a conversion with negative "ECU-internal" table values, a signed data type would be more appropriate.
9905-16	min / max values of COEFF conversion not set when conversion is unrelated in ASAP2.	Only if the COMPU-METHOD in the ASAP2 file is not used by definitions, default min/max values are imported which may not be appropriate for the future use of the conversion in DDS. When the COMPU-METHOD is used in the ASAP2 file, then its valid range is determined by the calibration limit values of the definitions using it.

	<p>ASAP2 header info PROJECT_NO and VERSION overwrites the Project Info (visible in DDS Editor), but no change of the referenced string #define, which contains the real information.</p>	<p>Typically, the projectInfo attributes 'ProjectNumber' and 'versionStructure' in @info.grl (also visible in DDS Editor as 'Project Info / Number' and 'Version Info / Structure') contain a reference to string #define (typically PRJ_NUM and VER_NO), which are also used to generate the DES_VERSION (identification check string for SAM. The ASAP2 Import overwrites these references to string #define by ASAP2 file attributes (PROJECT_NO, VERSION in ASAP2 header).</p> <p>Requirement: Optionally, the ASAP2 Import shall write to the referenced string #define statements, to make the ASAP2 identification usable for the DES_VERSION.</p> <p>Note: MCS Migration already shows this behavior (writes directly to string #define).</p>
--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

1.6 Address Import (I3E-695, ELF)

Defect Tracking Number	Title	Comment / Explanation
EHI 37291 (TIS 299)	I3E-695 / Alternate_Struct / False Messages	In case of embedded axes in structures and unions, a misleading error message occurs. The axis labels are virtual, they do not exist in software. Therefore, they cannot be found, of course.
	ELF - Hitachi tool chain compiler support: It is o.k. that addresses > 'signed long' are stored as negative values in GRL files.	High addresses for online data (> 'signed long') are stored as NEGATIVE values (equivalent of two's complement). They are casted to the correct HEX value when exported to description files (SAM, ASAP2). The SH7055 microcontroller design has a hard-coded RAM area at 0xFFFF:6000 – 0xFFFF:E000 (this is NOT a negative value, no two's complement!) Note: Addresses > 'signed long' in a DES file are an issue for SAM V1.6 evaluation.
	For struct element size mismatch: no support of individual struct type elements. If no address can be imported for an element because of size mismatch, the instance address is not set. Partial structure declaration (more elements in code) and also declaration of elements not existing in code is allowed, but the declared and existing structure type elements must have a correct size.	If one element of a structure instance must be suppressed (no address imported because data base size greater than size in object debug file, suppressed in subsequent description export to SAM or ASAP2), all elements of the structure are suppressed. In this case, no base address is imported for the structure instance. In fact, the DDS data model allows to store only one address for the instance of a structure (which is either set or not set). In addition, individual address offsets are stored with each element of a structure type. The order of the elements of the structure type in DDS has no relevance for the result of the address import. Subsequent description export filters (for SAM or ASAP2) check the existence of an instance address and export every struct element, which has an offset, individually. Structure type elements, which do not exist in code, do not receive an offset and are not be exported. Partial structure declaration is supported. Elements of a structure type may exist only in the software. In this case, the structure type elements in DDS get an address offset, the structure instance gets a base address and a size mismatch warning for the instance occurs (total size in DDS greater than in the software, because of the not declared elements).

	<p>For array dimension mismatch: no support for individual array elements. Arrayed instances have only one instance address.</p> <p>When 'AllowArraySizeMismatch' of option 'SizeMismatchMode' is chosen, even array elements not existing in ECU software are exported to the description file.</p> <p>Recommendation: HIDDEN function could be assigned to the upper array elements. Also, consider a #define as array index to adapt the array dimension of multiple arrayed instances in the data pool to the software, avoid choice 'AllowArraySizeMismatch'.</p>	<p>In case of an array size mismatch, all array elements are either suppressed (no address imported, suppressed in subsequent description export to SAM or ASAP2) or all array elements get an address (visible in the DDS Editor) and get exported.</p> <p>In fact, the DDS data model allows to store only one address for an arrayed instance (which is either set or not set). The 'byteSize' attribute of an arrayed instance stores the byte size of one array element.</p> <p>Although the address-import stores the "real array dimension in the software" as object array size, subsequent description file export filters do not use this information. Either all array elements are exported to the description file for SAM or ASAP2 (each as single instance), or none of the elements is exported.</p> <p>In conjunction with the 'AllowArraySizeMismatch' choice of the 'SizeMismatchMode' option (since DDS V2.28.1beta), an address of an arrayed instance is also imported when the array dimension in the data base is bigger than in the software.</p> <p>In some use cases, array dimension mismatch is intentional (i.e. 4 - 6 cylinder based variant handling, which is expressed by array dimensions). DDS contains always the array dimension 6, but the software can contain array dimension 4 or 6. Address will be always imported. The upper array elements '5' and '6' are always exported, even in the "dimension 4" case.</p> <p>In order to avoid this effect, it is recommended to assign the elements '5' and '6' to a separate function, which can be HIDDEN for 'dimension 4'.</p> <p>Recommended process improvement: Use #define statements are array index, avoid array size mismatch. Avoid the usage of option 'AllowArraySizeMismatch'.</p>
	<p>Tasking compiler (I3E) - For bits "unknown type" warning (I0905) with user dialog occurs, when the Tasking C166 compiler option <i>c166 -gb</i> is set. No problem for <i>c166 -g</i></p>	<p>The data type information "bit" in the "debug section" of I3E-695 files is sometimes intentionally suppressed on the compiler level (c166.exe) by the compiler setting "<i>c166 -gb</i>", instead of "<i>c166 -g</i>".</p> <p>Note: Option "<i>-g</i>" is mandatory for DDS operation in order to "enable symbolic debug information"; Option <i>-gb</i> "omits bit type and pointer behaviour description", it is set to be compatible with 3rd party tool). DDS looks for the data type in the debug part of I3E-695 and complains if it cannot be found, a user dialog occurs. When the user chooses "import anyway - for all", then the bit addresses of all C-defined "flying bits" are correctly imported.</p> <p>Note: The option concept of DDS allows to answer this question in advance, recommended setting is: HideMessage I0905 ("do not display")</p>

		<p>ResumeAction I0905 (“answer YES in advance”) But this setting can lead to false bitmasks 0x1 in case of “bitfield elements”. A separate error message is issued for “unknown type” in conjunction with bitfield.</p>
	<p>Tasking compiler (I3E) – For bitfield elements, a false bit position ‘0’ is stored and a false bitmask ‘0x1’ is exported to SAM and ASAP2, when the Tasking C166 compiler option <i>c166 -gb</i> is set and the user dialog warning about “unknown type (I0905)” is either switched off or ignored. No problem for <i>c166 -g</i></p>	<p>When the Tasking C166 compiler option <i>c166 -gb</i> is set, the information on bit position of bitfield elements is omitted from the generated I3E-695 file (set to zero). This is blocking for the bitfield processing of DDS. In case of a bitfield in DDS, the I3E-695 import looks for the ‘bit’ data type in the debug part of I3E-695 and complains that it cannot be found (I0905). A user dialog occurs (if not answered in advance by ‘ResumeAction’). ATTENTION: When the user chooses “import anyway”, then the relative bit position of all “bitfield elements” of the bitfield is corrupted. The false bitposition ‘0’ is stored for all and subsequently leads to a false bitmask 0x1 with all elements of the instance of the bitfield structure type in SAM and in ASAP2. The separate error message I1000 is issued for “unknown type” in conjunction with bitfield.</p>
	<p>Tasking compiler (I3E) - for odd number of bytes as struct type members, a size mismatch occurs (due to compiler-specific false interpretation)</p> <p>Same problem with map/axis representation models having an implicit structure type (isArray=false).</p>	<p>The I3E-695 file from C166 Tasking compiler provides always an even address as byte size of a structure instance (rounding, +1 byte for odd number of bytes). If the effective byte size is i.e 3 (three bytes in one structure type, byte alignment=1), the I3E-file reports 4 as byte size. This leads to inevitable but “confusing” mismatch warnings for such byte structures with an odd byte size. There is no impact on the imported address.</p> <p>Before DDS V2.28.1beta, the size check also assumed an even address, which has led to blocking false size mismatch errors (no address imported) with other compilers, i.e. DIAB/DATA.</p>

1.7 Automotive Data Dictionary Import (ADD-Import)

Defect Tracking Number	Title	Comment / Explanation
	Missing @Lock after discarding the ADD-Import changes	<p>When performing an ADD Import within the DDS Editor, the user has the choice to either 'commit' or to 'discard' the changes done by the import. When discarding the changes:</p> <ul style="list-style-type: none"> - no @Lock file will exist after the import that means, that the database could be opened a second time. - no 'default-responsible' exists

1.8 CVX Import (Calibration Values)

Defect Tracking Number	Title	Comment / Explanation
EHI 37319 (TIS 363)	Identifiers in the CVX file are handled case-sensitively.	The CVX standard explicitly defines identifiers as case-insensitive, as some tools /procedures may automatically map definitions to uppercase. This is not taken into account in DDS. The CVX Import filter handles identifiers case-sensitively.

1.9 DDX Import

Defect Tracking Number	Title	Comment / Explanation
	Limited support of DDX elements	<p>The content of the DDX file is defined in the XML Schema file "DDX.xsd" which is located in the DDS 'Tools' directory.</p> <p>The DDX Import doesn't yet fully support all elements of the DDX file but has the following restrictions:</p> <ul style="list-style-type: none"> - DDS attribute "responsible" not supported - Import of memory regions not fully supported - DDS attribute "adaptive" not supported - DDS attribute "comparisonQuantity" not supported - Enum's are not supported - Import of addresses not supported

1.10 Source Export

Defect Tracking Number	Title	Comment / Explanation
	Structure types and structure instances must not have the same name	In the C programming language, it is possible to use the same name for structure types and structure instances since structure types have their own name space, e.g. struct abc {...}; struct abc abc; Although it is possible in DDS to declare structure types and structure instances with the same name, the source export doesn't support this feature.
	Unrelated axes are exported.	The Genesys ME V1.0I tool does not export axis, which are not referenced by any map. DDS does export such axis, as some customers apply processes in which even unrelated axes are needed.
	Online data is inside a #pragma section when a top subfile with a #pragma is embedded in the main file. (Normally online data is exported before the segmentation #pragma. This only works when no top subfile is used).	This special case has no practical meaning. Typically, C files (in particular such with segmentation #pragma statements) contain only calibration data (in ROM) or only online data (in RAM). In case of a #pragma with the C file and mixed online data and calibration data, the rule in case of option "OrderMode = Genesys ME" is to export the online data first, then the calibration data according to special rules. This does not work when a top subfile is assigned, which also contains calibration data and has a #pragma. In this case the online data is placed after the calibration data of the subfile, which has most likely no impact on the compiler but on the calculation of the segmentation size for Tasking specific #pragma.
00-0605	Option "SectionPragmaSyntax=StartNumWithFirst" does not work when the #pragma is used in the top subfile.	The option "StartNumWithFirst" shall assign the postfix (_1, _2, ...) to a segmentation #pragma, when the #pragma occurs more than once (default is "no postfix for first occurrence, _1 for second occurrence etc.). Restriction: The view of "more than once" is limited to one C file in the DDS data base. In case of subfiles (e.g. top subfile with few definitions is embedded in host file, both files use the same segmentation #pragma), the first occurrence of the segmentation #pragma is the subfile will be typically without postfix, although the second #pragma follows in the host file. Nevertheless, the segmentation size will be correct. For the constellation of a top subfile with few entries and a host file with the same #pragma, the result will be the same as for the default setting of option SectionPragmaSyntax.

	<p>Alternate map & axis with “anonymous structure” representation model unusable inside a structure type.</p>	<p>In case of an “alternate anonymous structure” map (with embedded axis) used as element inside a structure type, an incorrect syntax is generated. Problem occurs with representation model 70: extern const _map struct alternate_test_map_STRUCT { signed char x; signed char y; } alternate_test_map[3]; const _map struct alternate_test_map_STRUCT alternate_test_map[3] = {{0x06,0x20}, {0x08,0x00}, {0x45,0x56}};</p> <p>This representation model is based on an implicit anonymous structure with each map instance: Using this model inside an explicit DDS structure type does not work. As an alternative, use model “7_alternate_array” inside a DDS structure type.</p>
	<p>#define of type ‘Token string’ as initial value for ASCII parameter array can lead to invalid C Source Export.</p>	<p>The DDS Editor allows to assign to #define types ‘string’ and ‘token string’ as initialization of a parameter array with representation model ‘ASCII’. There is no semantic check in case of token string, neither in DDS Editor nor in Source Export. It is possible to assign a token string, which is invalid in the context of an initialization. This is exported to C Source and will lead to a compiler error. /* Example for invalid C code: */ #define token test[56].tz const _param unsigned char param1[6] = {token};</p>

1.11 ASAP2 Export

Defect Tracking Number	Title	Comment / Explanation
	Automatic generation of AML sections	The automatic generation of AML sections does not work: <ul style="list-style-type: none"> – when the ifDataTemplate has no 'global' block – when the 'global' block is not the first block within the ifDataTemplate – when the AML description contains the same sub-Block at multiple locations (e.g. partly done in XCP protocol)
EHI 37292 (TIS 300)	Non-linear conversion not recognized / false linearization for ASAM-Export	Sometimes, non-linear conversions are not recognized from the ASAP2 export or a wrong linearization is done during the export
	ASCII / VAL_BLK parameter array only supported for one array dimension.	The ASAP2 Export has been extended to support parameter arrays of representation model "ASCII" and "Value Block", which are exported as one ASAP2 object of type ASCII / VAL_BLK. For more than one array dimension, DDS Editor and Source Export interpret the last array dimension as "string length" and the previous array dimension(s) as "number of strings". The ASAP2 Export does NOT support this interpretation but multiplies the dimensions (event logged with an error message).
	EPK Identification Check: must be related with array of representation model ASCII and initialised with a string.	Only a parameter array or a non-calibration-data array (representation model ASCII) have an effect on the EPK statement in the ASAP2 file, when assigned to the CalibrationSystemProtocol statement in @ecu.grl. The string initialization (directly in INIT page, or indirectly by a string #define) of the parameter / non.calib. array is exported the ASAP2 EPK, and also to the C Source (HEX file). The EPK export to ASAP2 cannot work when other definitions are assigned, like: Regular parameter arrays (decimal initialisation) Structure instances with several elements Note: If the element of a structure is an ASCII array, it can be references as <instance>.<element name>.
	Map with three axis (cuboid) is not supported.	DDS Editor, data model and source export support CUBOIDS, but not the Export filters.
1574	Default function not suppressed if it has no member.	EXPASAP2: The default function __DDS_EXPA2_DEFAULT_FUNC__ is always created at the ASAP2 file, even if no definition is referenced. Note: The VECTOR CANape system seems to have problems with that because there the ESL analogue list is built based on the function grouping of the ASAP2 file.

Defect Tracking Number	Title	Comment / Explanation
	EDICmobil V2.2 Export mode: No check of bitmask restrictions.	EDICmobil V2.2 has only a restricted support of bitmasks (i.e. no bitmasks allowed with data type 'long', only coherent bitmasks allowed for other data types (for details, please see the documentation of Softing GmbH). Bitmask restrictions for EDICmobil V2.2 are not validated, therefore problems in EDICmobil cannot be excluded.
	<p>Table conversion with "unsigned long" values is exported, although invalid ASAP2 syntax (see also in section "ASAP2 Import").</p> <p>Example for invalid conversion:</p> <pre> /begin COMPU_METHOD CM_FIXED_UL_00 "" TAB_INTP "%13.1" "" COMPU_TAB_REF CT_FIXED_UL_00 /end COMPU_METHOD /begin COMPU_TAB CT_FIXED_UL_00 "" TAB_INTP 2 0 0 4294967295 4294967295 /end COMPU_TAB </pre>	<p>The DDS data model allows table conversions (linear conv.) with "signed / unsigned long" table values. ASAP2 only allows the range of "signed long".</p> <p>DDS exports table values with an invalid conversion in such cases (see example on left side), but generates an error message (new in DDS V2.26.2).</p> <p>Note: 2-point linear table conversions (COMPU_TAB with two value pairs) can be exported as COEFF / RAT_FUNC conversions (when the ASAP2 Export option "Map2PointLinearConv2RatFunc" is set). This conversion type has no restriction of limits.</p> <p>This can be applied in order to avoid the risk of incompatibility in ASAP2 reading tools.</p>
ADDS-12886	Cyclic references of unit definitions are not recognized by export	If the DDS project contains unit definitions that reference to itself as a base-unit directly or indirectly in a cycle, are not checked during export and therefore may lead the filter to fail. A check must be done explicitly by running the project check feature beforehand.

1.12 DDX Export

Defect Tracking Number	Title	Comment / Explanation
	Limited support of DDX elements	<p>The content of the DDX file is defined in the XML Schema file "DDX.xsd" which is located in the DDS 'Tools' directory.</p> <p>The DDX Export doesn't yet fully support all elements of the DDX file but has the following restrictions:</p> <ul style="list-style-type: none"> – Enum's are not fully supported (e.g. not supported as array- and axis-size)

1.13 Ascet Export

Defect Tracking Number	Title	Comment / Explanation
A-001-00	Project with Dependent SystemConstants	Export of an ADD Project containing "Dependent SystemConstants" to ASCEt leads to several errors; not yet supported.

1.14 Ascet Import

Defect Tracking Number	Title	Comment / Explanation
A-002-00	Dependent SystemConstants Condition Import	"Dependent SystemConstant" Conditions get lost when importing Ascet data into ADD (Offline).

2 'Not supported' or obsolete filters and components

2.1 Not Supported

'Not supported' DDS filters/components are no longer actively improved and maintained. Nevertheless they are still present after the DDS installation due to legacy reasons.

2.2 Obsolete

Obsolete DDS filters/components are no longer present in the current installation disks. If such a filter/component is needed for some reason, an older version of DDS has to be used temporary to get something done.

2.2.1 Import Configuration Unit (GRL) filter

Note: Since DDS V5.0.0, this filter is obsolete!

Known Defects and Restrictions of last version

1.) Attention with Import Configuration Unit (GRL) filter

Importing or merging external configuration units to a DDS data pool only works under the following conditions: Global attributes (data type, conversions, physical units, formats) referenced in the imported configuration unit (GRL file) should already exist.

Array sizes, Structure types and structure instances in the imported configuration unit and in the data pool should be consistent. Importing a structure type with modified structure leads to corruption when already existing instances of this structure type have individual attributes.

Defect Tracking Number	Title	Comment / Explanation
	Message "Undefined and not used" occurs at deletion, when using ImpConf.Unit(GRL) operation.	When definitions are deleted because of an operation of the Import Configuraiton Unit (GRL) filter, i.e. when an Import_Replace replaces a GRL file by another one with fewer definitions, then the deleted ones are persistently reported by the SEML check as "undefined and not used". Please note that this check is also automatically invoked at the end of the ImpConf.Unit (GRL) operation.
	"C/H file list reference to <deleted data>" is reported after deletion by 'Import Conf. Unit (GRL)' operation. These messages are confusing but not relevant.	The GRL objects of Source Files (C, H) contain reverse reference lists to the definitions, which describe the order for Source Export. These lists are updated automatically when adding of removing definitions with C/H file assignment. This also happens for ImpConf.Unit(GRL) operation, but confusing irrelevant error messages (L2037, L2041) may occur, which report an inconsistency. After saving the result to disk, these inconsistencies are fixed and the C/H file lists are updated.
	'Delete Configuration Unit (GRL)' not available in command line interface.	The functionality of removing configuration units (GRL files) from the DDS data pool is available in two ways: Import Conf. Unit (GRL) with option '-Delete' (in command line: impgirl.exe)

Defect Tracking Number	Title	Comment / Explanation
		Delete Conf. Unit (GRL). Not available in command line interface. This "filter" is only an internal indirection of the DDS Editor. Please use 'impgirl.exe -Delete ...' to automate the removal of configuration units from the DDS data pool.

As the 'Import Configuration Unit (GRL)' filter is only an option handler using the Data Pool Library, functional defect symptoms visible in Import Conf.Unit (GRL) are most likely originated from the Data Pool Library.

2.2.2 Report Generator

Note: Since DDS V5.0.0, this filter is obsolete!

Known Defects and Restrictions of last version

Defect Tracking Number	Title	Comment / Explanation
EHI 37314 (TIS 326)	Missing (init-) value for calibration data	There is no possibility to export (init-) values

2.2.3 Export SAM2000 filter

Note: Since DDS V8.2, this filter is obsolete!

3 Supported Compilers (Address-Import)

In general, DDS supports any compiler which generates ELF or I3E-695 files. However due to the different compiler dialects it might be that the DDS import filters must be slightly adapted to some compiler specifics. The following compilers are already evaluated and fully supported.

3.1 IEEE-695

Compiler	Description	Version
Tasking C166	16-bit compiler for Infineon C166 processor	v50r0 v60r4, v60r5 v75r0
Tasking TriCore	32-bit compiler for Infineon TriCore processor Limited support: DDS supports the ANSI-C data types plus the TriCore specific '_bit' data type. DDS does not support additional TriCore specific extensions like the data types '_fract', '_sfract' and '_accum', the packed data types, the modifier '_sat' and circular buffers. DDS doesn't fully support bitfields. Only simple bitfields where all members have the same data type and which do not exceed the overall amount of 32 bits are supported. DDS does not support enum types with a size different to the size of the data type <i>int</i> . (See <i>pragma intenum</i>).	v1.4r1

3.2 ELF – DWARF 1.0

Compiler	Description	Version
Hitachi	Hitachi tool chain compiler for Hitachi SH7050, SH7055 microcontroller.	S32HEWMCSSH version 5.1
DiabData	DiabData Compiler Suite C for Motorola black oak	4.3f 4.4 5.2.1

3.3 ELF – DWARF 2.0

Compiler	Description	Version
Hitachi	Hitachi tool chain compiler for Hitachi SH7050, SH7055 microcontroller.	S32HEWMCSSH version 7.0
Renesas	Renesas compiler for Hitachi SH2 (SH7055) microcontroller.	V9.01
Greenhills	Compiler for PowerPC microcontroller	V2.1

Compiler	Description	Version
Greenhills	Compiler for NEC V800 microcontroller	GHS C 2013.5.4 [dual] GHS C 2013.5.5 [dual] GHS C 2021.1.4 [dual]
Tasking TriCoreVX	32 bit compiler for Infineon TriCore processor Limited support: DDS supports the ANSI-C data types according to C90. DDS does not support new data types defined in C99 and additional TriCore specific extensions like the data types ‘__bit’, ‘__fract’, ‘__sfract’ and ‘__laccum’, the packed data types and circular buffers. DDS doesn’t fully support bitfields. Only simple bitfields where all members have the same data type and which do not exceed the overall amount of 32 bits are supported. DDS does not support enum types with a size different to the size of the data type <i>int</i> . (See switch <i>--integer-enumeration</i>).	v2.0r1 v2.1 v2.2r3 v2.2r3p1 v3.2.R3 v3.3.R1 v3.4 v3.5 v4.0 (AURIX) v4.1.r2(AURIX)
METROWERKS HC12	Notes on the support for the METROWERKS compiler V1.2 for HC12: The alignment attributes within the MEMORYLAYOUT entity (usually located in the file @ecu.grl) should be set to 1 for all types. Pointer data types are not verified yet; they should not be used within structures to avoid problems with offset calculation within such data types. Attention: When a variable is not used within your project, the variable will nevertheless be present within the ELF file but with the address 0; as the address 0 is also a valid address, such situations will not be detected by the address import or ASAP2 export.	V1.2
GNUSH	GNUSH v0603 from KPIT Cummins Infosystems Limited, a cross compiler toolchain for Renesas (formerly Hitachi and Mitsubishi) SH series of micro controllers.	V0603
Softune FUJITSU MB91F469G	Compiler for FUJITSU MB91F469G (MB91460 family) processor	V60L06
HighTec V3.4.5	HighTec GNU development tool for Infineon's TriCore family	V3.4.5.1
DiabData	DiabData Compiler Suite C for Motorola black oak	5.3.1

Compiler	Description	Version
		5.4.0 5.6.1 5.8.0
MW EABI PPC C-Compiler	MW EABI PPC C-Compiler	
IAR Embedded Workbench	IAR ANSI C/C++ Compiler	V7.50.2.10312/W32 for ARM

3.4 ELF – DWARF 3.0

Compiler	Description	Version
Tasking VX for C166	<p>TASKING VX-Toolset for C166</p> <p>Limited support:</p> <ul style="list-style-type: none"> ASAP2Toolkit doesn't fully support bitfields. Only simple bitfields where all members have the same data type and which do not exceed the overall amount of 32 bits are supported. ASAP2Toolkit does not support enum types with a size different to the size of the data type <i>int</i>. (See switch <i>--integer-enumeration</i>). Additional base types for C (as revised for 1999) are not supported Java is not supported No namespace support for C++ An optional section for global type names (similar to the global section for objects and functions) is not supported Adopt UTF-8 as the preferred representation of program name strings not supported 	v2.3

3.5 ELF – DWARF 4.0

Compiler	Description	Version
IAR Embedded Workbench	IAR ANSI C/C++ Compiler	V8.30.1.114/W32 for ARM