TargetLink Data Dictionary Manager

Manager Reference

For TargetLink Data Dictionary 5.1

Release 2020-B – November 2020

dSPACE

How to Contact dSPACE

Mail:	dSPACE GmbH Rathenaustraße 26 33102 Paderborn Germany
Tel.:	+49 5251 1638-0
Fax:	+49 5251 16198-0
E-mail:	info@dspace.de
Web:	http://www.dspace.com

How to Contact dSPACE Support

If you encounter a problem when using dSPACE products, contact your local dSPACE representative:

- Local dSPACE companies and distributors: http://www.dspace.com/go/locations
- For countries not listed, contact dSPACE GmbH in Paderborn, Germany.
  Tel.: +49 5251 1638-941 or e-mail: support@dspace.de

You can also use the support request form: http://www.dspace.com/go/supportrequest. If you are logged on to mydSPACE, you are automatically identified and do not need to add your contact details manually.

If possible, always provide the relevant dSPACE License ID or the serial number of the CmContainer in your support request.

Software Updates and Patches

dSPACE strongly recommends that you download and install the most recent patches for your current dSPACE installation. Visit http://www.dspace.com/go/patches for software updates and patches.

Important Notice

Contents

About This Reference

Contents

This reference provides detailed information on the menus, context menus and dialogs of the Data Dictionary Manager.

Symbols

dSPACE user documentation uses the following symbols:

Symbol	Description
DANGER	Indicates a hazardous situation that, if not avoided, will result in death or serious injury.
WARNING	Indicates a hazardous situation that, if not avoided, could result in death or serious injury.
CAUTION	Indicates a hazardous situation that, if not avoided, could result in minor or moderate injury.
NOTICE	Indicates a hazard that, if not avoided, could result in property damage.
Note	Indicates important information that you should take into account to avoid malfunctions.
Tip	Indicates tips that can make your work easier.
	Indicates a link that refers to a definition in the glossary, which you can find at the end of the document unless stated otherwise.
	Precedes the document title in a link that refers to another document.

Naming conventions

dSPACE user documentation uses the following naming conventions:

%name% Names enclosed in percent signs refer to environment variables for file and path names.

< > Angle brackets contain wildcard characters or placeholders for variable file and path names, etc.

Special folders

Some software products use the following special folders:

Common Program Data folder A standard folder for application-specific configuration data that is used by all users.

`%PROGRAMDATA%\dSPACE\<InstallationGUID>\<ProductName>`

or

`%PROGRAMDATA%\dSPACE\<ProductName>\<VersionNumber>`

Documents folder A standard folder for user-specific documents.

`%USERPROFILE%\Documents\dSPACE\<ProductName>\<VersionNumber>`

Local Program Data folder A standard folder for application-specific configuration data that is used by the current, non-roaming user.

`%USERPROFILE%\AppData\Local\dSPACE\<InstallationGUID>\<ProductName>`

Accessing dSPACE Help and PDF Files

After you install and decrypt dSPACE software, the documentation for the installed products is available in dSPACE Help and as Adobe® PDF files.

dSPACE Help (local) You can open your local installation of dSPACE Help:

- On its home page via Windows Start Menu
- On specific content using context-sensitive help via **F1**

dSPACE Help (Web) You can access the Web version of dSPACE Help at www.dspace.com.

To access the Web version, you must have a *mydSPACE* account.

PDF files You can access PDF files via the icon in dSPACE Help. The PDF opens on the first page.

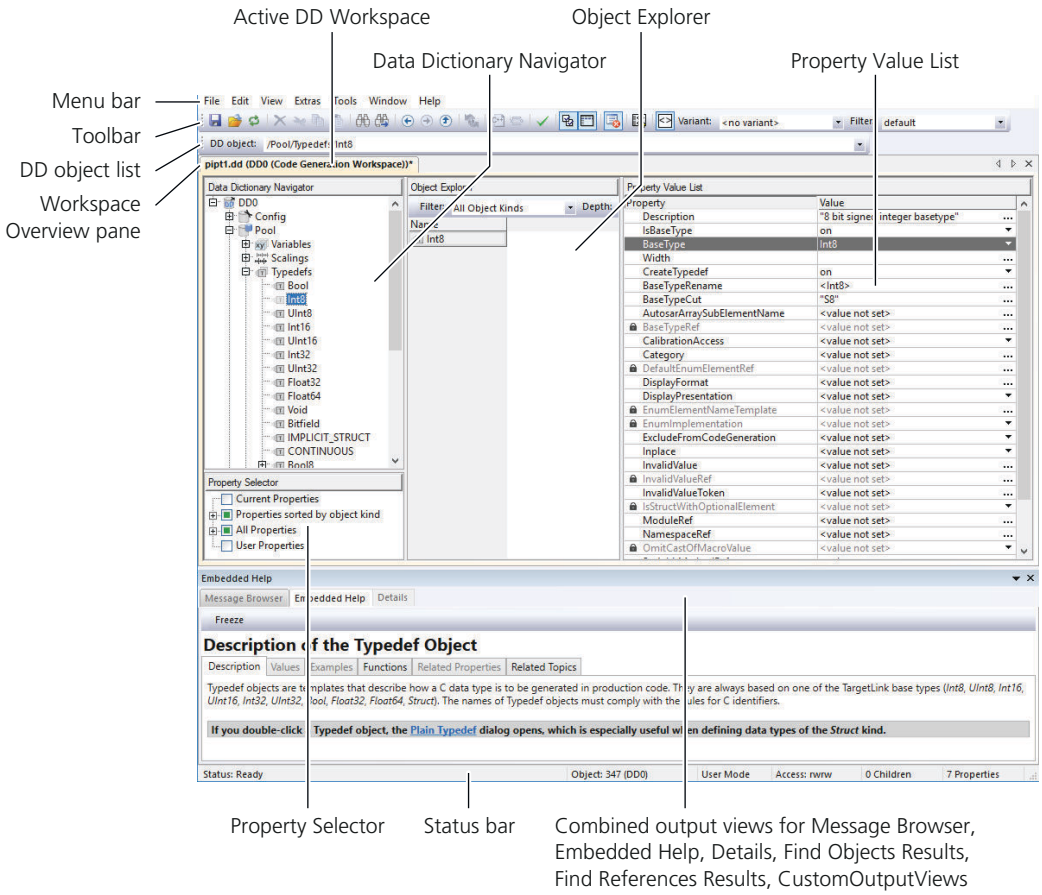
Panes of the GUI

Where to go from here

Information in this section

TargetLink Data Dictionary Manager Description

GUI



Purpose

To display DD objects in a tree structure, and to handle DD objects and their properties in multiple DD workspaces for an iterative development process.

Access

You can access this tool via:

Toolbar	Data Dictionary Manager toolbar button in the TargetLink Main Dialog, any TargetLink block dialog, any plot window (see TargetLink Data Dictionary Manager (TargetLink Tool and Utility Reference)).

API function	▪ **dsddman ('Open','<ddfile>')** where **<ddfile>** stands for the DD project file to be opened (the extension ".dd" can be omitted). ▪ **dsddman** to inspect the currently loaded DD workspace.
Windows **Start** menu	**Start** – **Programs** – **dSPACE Tools** – **Data Dictionary Manager** starts the Data Dictionary Manager in stand-alone mode.
File Explorer	Double-click on *.dd file.

Stand-alone mode

The **TargetLink Data Dictionary Manager** can be started in stand-alone mode (that means independently from MATLAB).

Some utilities that are called from the **TargetLink Data Dictionary Manager** are written as MATLAB M scripts. The following utilities cannot be used with the **TargetLink Data Dictionary Manager** in stand-alone mode:

- Opening the code file associated with a DD object.
- Showing the block associated with a DD object in a Simulink model.
- Creating a custom output view.
- Creating an HTML Comparison report.
- Menu Extensions specified via XML files.
- The following commands from the **Tools** menu:
  - **Manage Build**
  - **Import MATLAB Variable Objects**
  - **Export DD Variable Objects**
- Some **AUTOSAR** functionalities.

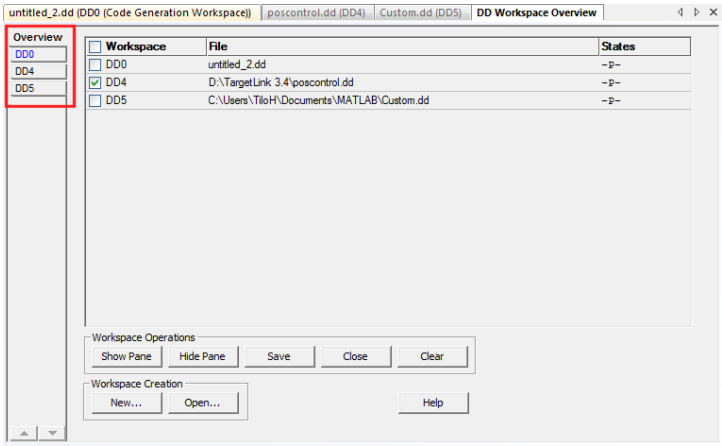
Related topics

References

<Custom Output View>....24
Data Dictionary Navigator....16
Find Object Results....25
Find References Results....26
Message Browser....21
Object Explorer....19
Property Selector....20
Property Value List....18

DD Workspace Overview

Pane



Purpose

To manage DD workspaces.

Description

The DD Workspace Overview displays an Overview of the DD workspaces in the Data Dictionary. You can select a DD workspace to inspect its properties and to perform operations on them.

Overview operations

Show pane To show the selected workspaces.

Hide pane To hide the selected workspaces.

Save To save the selected DD workspaces. If the DD workspace is untitled, the Save As dialog opens.

Close To close the selected DD workspace.

Clear To delete all DD objects from the selected DD workspaces. The DD workspaces are re-initialized to contain the minimum number of required DD objects.

New To create a new DD workspace.

Note

The DD1 ... DD3 workspaces are used internally by TargetLink. If you create a new DD workspace, it is named DD4, DD5, etc.

Open To open a DD project file in a new DD workspace.

Workspace operations

Show Pane To show the selected DD workspace.

Hide Pane To hide the selected DD workspace pane.

Save To save the selected DD workspace. If the DD workspace is **untitled**, the **Save As** dialog opens.

Save As To save the selected DD workspace with a new file name and/or in a different folder.

Save and Close To save and close the selected DD workspace.

Swap with DD0 To make a secondary DD workspace the primary DD workspace.

Export To export the selected DD workspace in another file format.

Import To import a file to the selected DD workspace.

Related topics

Basics

Basics on the Data Dictionary Manager (📖 TargetLink Data Dictionary Basic Concepts Guide)

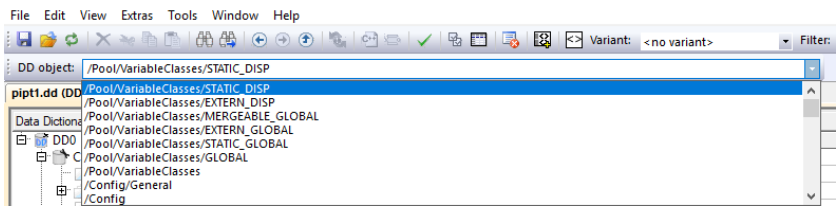
HowTos

How to Swap a Secondary DD Workspace with the Primary DD Workspace (📖 TargetLink Data Dictionary Basic Concepts Guide)

References

DD Object List

Pane



Purpose

To display the DD path of the selected DD object and to show a history of the previously selected DD objects.

Description

The **DD object** list displays the DD path of the currently selected DD object. Like the address bar in the File Explorer, you can access DD objects by typing or pasting a path (case-sensitive). The **DD object** list contains a history with the last selected DD objects.

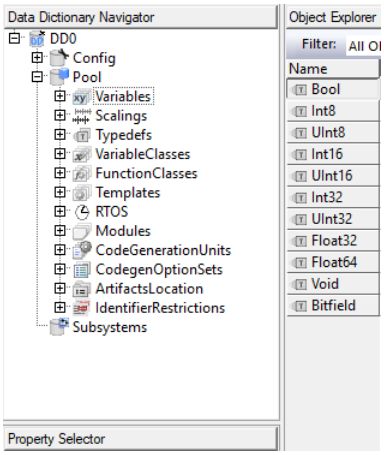
Instead of a path, you can also enter a DD handle (unsigned 32-bit number). You can enter DD handles for any DD objects in any DD workspaces (not necessarily the currently selected). If entered, the referenced DD object in its DD workspace is automatically selected.

Related topics

References

Data Dictionary Navigator

Pane



Purpose

To display the DD objects of the open DD workspace in the DD object tree.

Description

After you open the TargetLink Data Dictionary Manager, the Data Dictionary Navigator shows the topmost level of the DD object tree of workspace DD0. By default, the additional branches are hidden. Click the plus or minus sign to expand or collapse the particular branch.

Click a DD object to select it. The properties of the selected DD object are displayed and can be edited in the property value list (refer to Property Value List on page 18).

Right-click a DD object to open the context menu of the Data Dictionary Navigator, which offers various commands to work with the selected object.

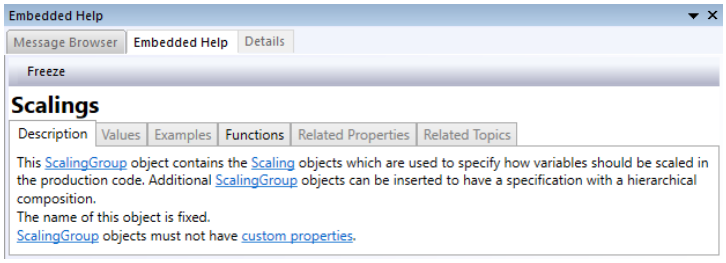
Related topics

Basics

Basics on the Data Dictionary Manager (TargetLink Data Dictionary Basic Concepts Guide)
Basics on User Modes and Access Rights (TargetLink Data Dictionary Basic Concepts Guide)

Embedded Help

Pane



Purpose

To provide detailed information on selected DD objects and properties.

Description

The Embedded Help pane contains the following tabs:

- Description
  Displays a detailed description of the selected DD object or property.
- Values
  Displays the values of the selected DD object or property.
- Examples
  Displays useful examples related to the selected DD object or property.
- Functions
  Displays API functions related to the selected DD object or property.

For information on Data Dictionary MATLAB API functions, refer to Getting Started with the DD MATLAB® API (TargetLink Data Dictionary Basic Concepts Guide).

- Related Properties

  Displays related properties to the selected DD object or property.
- Related Topics

  Displays help topics related to the selected DD object or property.

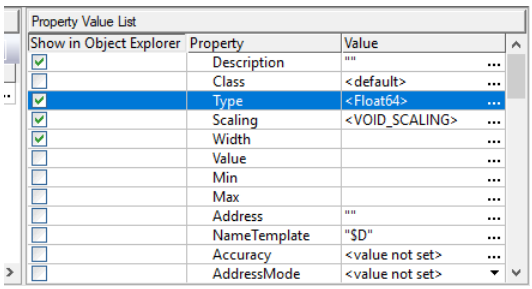
Related topics

Basics

Getting Started with the DD MATLAB® API (TargetLink Data Dictionary Basic Concepts Guide)

Property Value List

Pane



Purpose

To display and edit the properties of the selected DD object.

Description

You can set properties of DD objects as follows:

- You can use the ... button or the drop-down list.
- You can edit the value directly in the line or double-click to open its dialog.
- You can access various commands on the context menu of the selected property.

Note

Properties with a symbol are read-only. Refer to Basics on User Modes and Access Rights (TargetLink Data Dictionary Basic Concepts Guide).

Related panes for objects and properties

The Property Value List interacts with the Property Selector and the Object Explorer. The Show in Object Explorer checkboxes let you filter the properties shown in the Object Explorer.

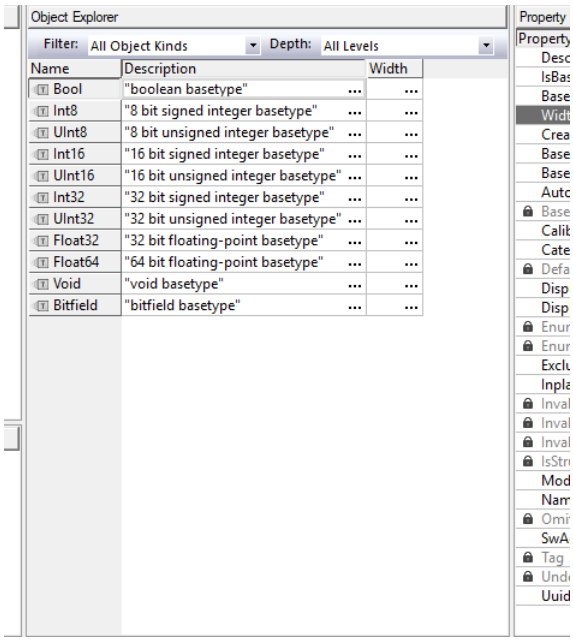
Related topics

Basics

Basics on the Data Dictionary Manager (TargetLink Data Dictionary Basic Concepts Guide)
Basics on User Modes and Access Rights (TargetLink Data Dictionary Basic Concepts Guide)

Object Explorer

Pane



Purpose

To display and edit the properties of multiple DD objects.

Description

DD objects and their properties are displayed in a table. You can influence the table contents to be displayed via the Property Selector, the Object Explorer - Filter on page 180, and the Object Explorer - Depth on page 179 lists. The settings in the Filter specify which object types are included in the list. The settings in the Depth specify the depth of the hierarchy of DD objects to be

displayed. The settings in the **Property Selector** determine which properties (columns) are shown.

Property values can be edited by clicking on the current value. If a limited set of values exists, you can select a new value from a list which opens; otherwise, you can enter a new value in a dialog which opens or an edit field directly.

You can sort the property values in ascending/descending order by clicking the column header or via the associated sort command (refer to How to Sort DD Objects by Property (📖 TargetLink Data Dictionary Basic Concepts Guide)).

If the field of a property value is gray, then this property is not editable for the object, for example, Class for a **Typedef** object.

Related panes for objects and properties

The **Object Explorer** interacts with the **Property Selector** and the **Property Value List**. The **Show in Object Explorer** checkboxes in the **Property Value List** let you filter the properties shown in the **Object Explorer**.

Related topics

HowTos

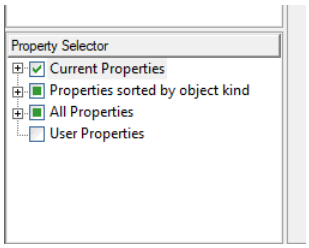
How to Set the Properties of Multiple Data Dictionary Objects (📖 TargetLink Data Dictionary Basic Concepts Guide)
How to Sort DD Objects by Property (📖 TargetLink Data Dictionary Basic Concepts Guide)
How to Specify Objects Shown in the Object Explorer (📖 TargetLink Data Dictionary Basic Concepts Guide)

References

Remove All Columns 182
Remove Column 182
Show Object Explorer 183

Property Selector

Pane



Purpose

To select or clear the properties of the DD objects to be displayed in the Object Explorer.

Description

The Property Selector displays the tree of the properties available for the DD objects. You can select an individual property at a deeper hierarchy level if you expand the node and select the individual property.

Current properties Shows the currently displayed properties (columns). Lets you clear these properties.

Properties sorted by object kind Lists all properties sorted by the object kind which they belong to. Lets you select or clear the properties to be displayed in the Object Explorer.

All properties Lists all properties in alphabetical order. Lets you select or clear the properties to be displayed in the Object Explorer.

User properties Indicates whether user-defined properties are displayed in the Object Explorer.

Related panes for objects and properties

The Property Selector interacts with the Object Explorer and the Property Value List. The selected properties are displayed in the Object Explorer.

Related topics

HowTos

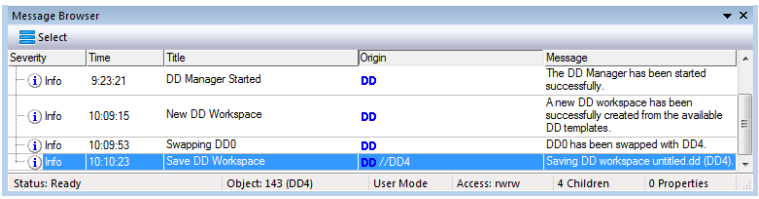
How to Specify Properties (Columns) Shown in the Object Explorer (TargetLink Data Dictionary Basic Concepts Guide)

References

Show Property Selector..... 184

Message Browser

Pane



Purpose

To display the messages of the Data Dictionary Manager.

Description

The Message Browser provides a history of all error, warning, and information messages that occur when you work with the Data Dictionary Manager. This helps you check the system state.

Hierarchical display of messages The messages are shown hierarchically in the Message Browser. The root messages are always on the top level. They give information on the actions that you performed in the Data Dictionary Manager, for example, renamed a DD object or edited a property value. Messages on lower levels belong to their parent level.

Severity, time, title, origin, and text of a message Each message consists of five parts: severity, time, title, origin and message text.

Part	Description
Severity	There are three types of messages from Data Dictionary Manager according to severity level. Each message has a symbol that indicates the message type. Refer to Icons.
Time	The time when the message occurred.
Title	The title of the message.
Origin	The DD object that the message comes from.
Message	The content of the message.

Access to DD object, and help related to a message If you click the Title of a error or warning message, the online help opens, if a related topic exists. In a hierarchy of messages, the subordinate messages provide access to the respective online help topics.

If you click the Origin of a message, the related DD object is selected in the Data Dictionary Navigator if it exists in the currently open DD workspace. If it does not exist, for example, because it was deleted or renamed after the selected message occurred, the previously selected DD object remains selected.

Icons

The icons indicate the severity of messages.

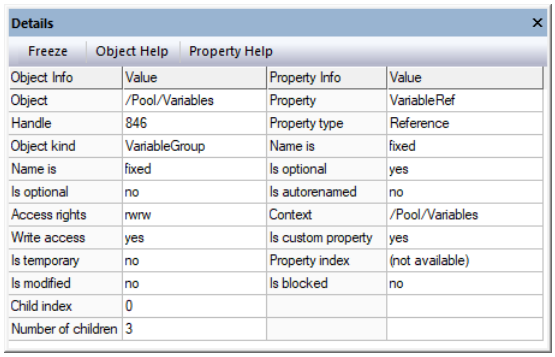
Icon	Description
	Identifies an error.
	Identifies a warning.
	Identifies a piece of information.

Tip

If you have multiple elements in this pane, you can select/multiselect them via left click and then click the Select button. If the selected elements refer to DD objects, these objects are then automatically displayed in the Data Dictionary Navigator *and* in the Object Explorer.

Details

Pane



Object Info	Value	Property Info	Value
Object	/Pool/Variables	Property	VariableRef
Handle	846	Property type	Reference
Object kind	VariableGroup	Name is	fixed
Name is	fixed	Is optional	yes
Is optional	no	Is autorenamed	no
Access rights	rwrw	Context	/Pool/Variables
Write access	yes	Is custom property	yes
Is temporary	no	Property index	(not available)
Is modified	no	Is blocked	no
Child index	0		
Number of children	3		

Purpose

To display detailed information on the selected DD object and, if selected, its associated properties.

Description

The Details pane provides the following buttons and tables:

Freeze Keeps the content of the pane, even if you selected another DD object or property.

Object Help Opens dSPACE Help for the selected DD object.

Property Help Opens dSPACE Help for the selected property.

The Details pane provides the following DD object information (read-only) :

Object Info	Description
Object	Displays the path of the selected DD object.
Handle	Displays the handle of the DD object.
Object kind	Displays the kind of the DD object.
Name is	Indicates whether the name of the DD object is arbitrary, fixed, or a C identifier.
Is optional	Indicates whether the DD object is optional.
Access rights	Displays the access rights.
Write access	Indicates whether you have write access.
Is temporary	Indicates whether the DD object is temporary.
Is modified	Indicates whether the DD object is modified. 'Yes' is displayed if you modified the DD object after you created a new DD or after you saved the DD.
Child index	Displays the child index, which is the position of the DD object below the parent DD object.
Number of children	Displays the number of children.

The Details pane provides the following property information (read-only):

Property Info	Description
Property	Displays the name of the property.
Property type	Displays the property type.
Name is	Indicates whether the name of the property is arbitrary, fixed or a Cidentifier.
Is optional	Indicates whether the property is optional.
Is autorenamed	Indicates whether the property is automatically renamed.
Context	Returns the path of the object that defines the context for the specified property.
Is custom property	Indicates whether it is a custom property.
Property index	Displays the property index, which is the position of the property in the DD object.
Is blocked	Indicates whether and why the property is blocked.

<Custom Output View>

Pane

Purpose

To display custom messages issued by MATLAB functions in a custom output view of the TargetLink Data Dictionary Manager.

Description

A custom output view displays custom messages that can be issued when you work with the TargetLink Data Dictionary Manager.

ID, tag, origin, and text of a message Each message consists of four parts: id, tag, origin and message text.

Part	Description
ID	ID of the message.
Tag	Tag (may be used to introduce message categories, i.e., 'Info').
Origin	DD object that is associated with message.
Message	The content of the message.

Access to DD object If you click the Origin of a message, the related DD object is selected in the Data Dictionary Navigator if it exists in the currently open

DD project file. If it does not exist, for example, because it was deleted or renamed after the selected message occurred, the previously selected DD object remains selected.

A custom output view is created with an individual name. Therefore, the pane's label depends on the name that the custom output view is created with.

Tip

If you have multiple elements in this pane, you can select/multiselect them via left click and then click the Select button. If the selected elements refer to DD objects, these objects are then automatically displayed in the Data Dictionary Navigator *and* in the Object Explorer.

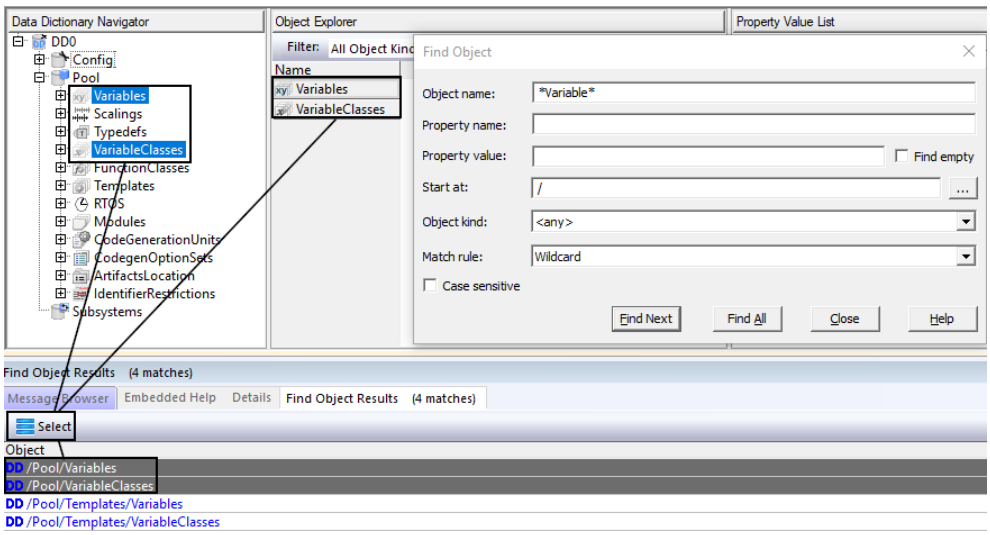
Related topics

HowTos

How to Create Custom Output Views (TargetLink Data Dictionary Basic Concepts Guide)
How to Display Custom Messages (TargetLink Data Dictionary Basic Concepts Guide)

Find Object Results

Pane



Purpose

To display all DD objects that match the search criteria.

Description

The Find Object Results pane provides information on the DD objects that match the search criteria.

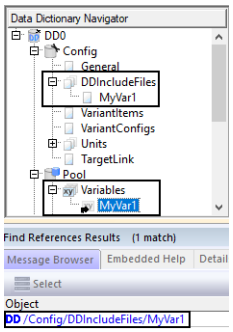
Object, and value Each entry consists of two parts: object, and value.

Part	Description
Object	Name and path to the DD object.
Value	Value of the found object.

Access to DD object If you have one or more search results in the Find Object Results pane, you can select/multiselect them via left click and then click the Select button. The selected search results are then automatically displayed in the Data Dictionary Navigator *and* in the Object Explorer.

Find References Results

Pane



Purpose

To find DD objects that reference the selected DD object.

Description

The Find References Results pane provides information on the DD objects that reference the selected DD object.

Object, type, property, and library Each entry consists of four parts: object, type, property, and library.

Part	Description
Object	Name and path to the object in the TargetLink Data Dictionary (DD object) or model (Simulink block or Stateflow object). Refer to Icons.
Type	There are three types of objects that may reference DD objects: other DD objects (ddobject), Simulink blocks (slblock) or Stateflow objects (sfobject).

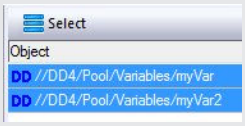
Part	Description
Property	Referencing object property that references the referenced object.
Library	Name and path to the object in a library that the referencing object is based on.

Access to DD object If you click the Object of an entry, the related object is selected. If the object is a:

- DD object, it is selected in the Data Dictionary Navigator if it exists in the currently open DD workspace. If it does not exist, for example, because it was deleted or renamed after the Find References command was executed, the previously selected DD object remains selected.
- Simulink block and the model is open, the Simulink Editor is activated and the block is highlighted.
- Stateflow object and the model is open, Simulink's Model Explorer opens.

Tip

If you have one or more DD object search results in the Find References Results pane, you can select/multiselect them via left click and then click the Select button. The selected DD object search results are then automatically displayed in the Data Dictionary Navigator *and* in the Object Explorer.



Icons

The icons indicate the object types.

Icon	Description
DD	Identifies a DD object.
	Identifies a Simulink block.
	Identifies a Stateflow object.

Basic Dialogs

Where to go from here

Information in this section

Property Dialogs

Where to go from here

Information in this section

Advanced Edit String List

Purpose

To easily add, delete, change, or reorganize a list of values, e.g., to add a list of values from external sources in one step or to edit the values in your favorite editor.

Access

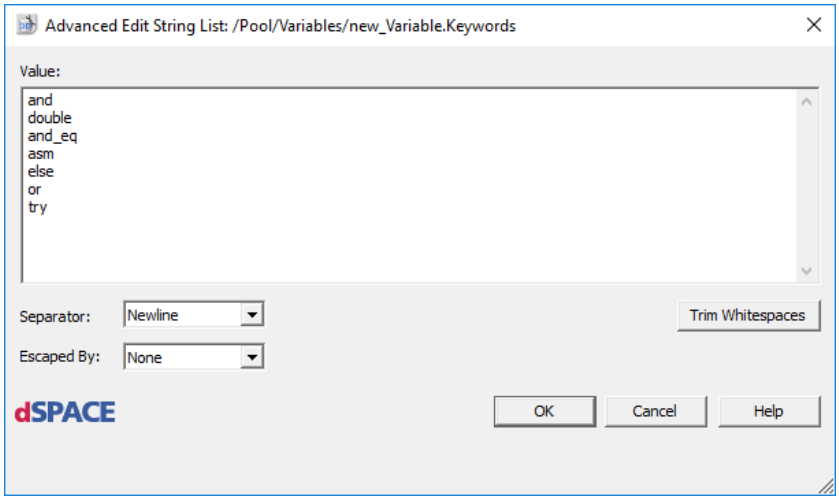
You can access this dialog by clicking the edit button  in the **Edit String list** dialog.

Result

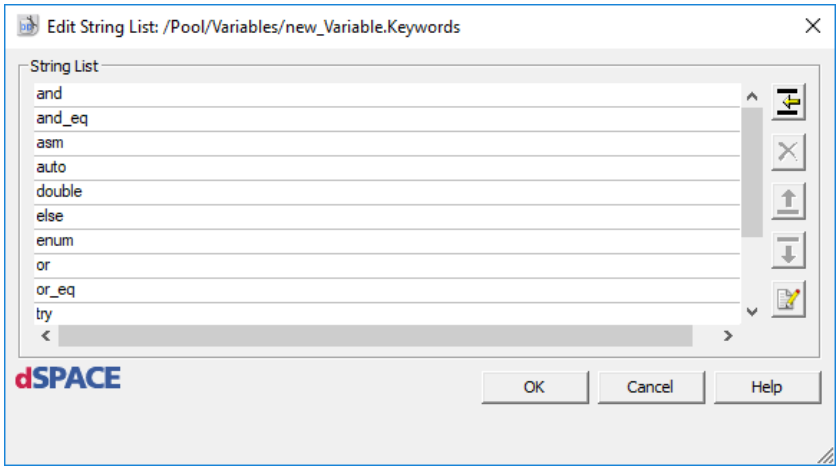
The list of values is changed and applied to the string list in the **Edit String List** dialog.

Description

You can extend the list of values in the Edit String List by opening the Advanced Edit String List dialog via the edit button.



If you want to keep existing values, copy the displayed values into your favorite editor. Add new values and copy all values from the editor back into the Advanced Edit String List dialog. Click OK. In the Edit String List dialog the list of values is extended.



Dialog settings

The Advanced Edit String List dialog provides the following options:

- Use a specific separator.
- Use quotes.
- Remove whitespaces.

Note

If the access right of the related object is read-only, the Advanced Edit String List dialog is not available.

Related topics

References

Edit StringList 35

Declaration Statement

Purpose

To set the declaration statements of a DD VariableClass or DD FunctionClass object.

Access

You can access this dialog in the following ways:

- In the Property Value List or Object Explorer,click the ··· button or double-click the Value field.
- On the context menu of the selected property, click the Edit button .
- In various dialogs, such as Plain Variable dialog or Plain Typedef dialog, click the ··· button.

Result

The declaration statements of the DD VariableClass or DD FunctionClass object are set.

Description

The declaration statements are not written to the DD VariableClass or DD FunctionClass object unless you have corrected all invalid settings.

Dialog settings

Keyword Lets you select a declaration statement keyword. Invalid declaration statement keywords are shown in red.

Delete button Lets you delete an invalid keyword (displayed in red) or a keyword that you do not need from the list of keywords.

Value Lets you enter a declaration statement for the selected keyword.

Status bar Displays invalid settings.

Related topics

HowTos

How to Specify Code Decorations in the Data Dictionary (📖 TargetLink Customization and Optimization Guide)

Examples

Example of Mapping Variables to Memory Sections via Code Decorations (📖 TargetLink Customization and Optimization Guide)

Edit Bitfield

Purpose

To select one or more bits.

Access

You can access this dialog in the following ways:

- In the **Property Value List** or **Object Explorer**,click the ··· button or double-click the **Value** field.
- On the context menu of the selected property, click the **Edit** button .
- In various dialogs, such as **Plain Variable** dialog or **Plain Typedef** dialog, click the ··· button.

Result

The selected bits are set.

Note

Some bits cannot be combined with others. Such bits are grouped (indicated by small lines between the value groups). You can select only one bit from each group.

Note

The status bar informs you about invalid entries. As long as entries are invalid, the **OK** button is not available. For more information on valid entries, refer to the embedded help.

Dialog settings

Select none Selects none of the bits.

Select all Selects all of the bits. The Select all button is available only if all bits can be set at once (no groups).

To select a bit, select its checkbox.

Edit Matrix

Purpose

To set values that can be a vector or a matrix.

Access

You can access this dialog in the following ways:

- In the Property Value List or Object Explorer,click the ··· button or double-click the Value field.
- On the context menu of the selected property, click the Edit button .
- In various dialogs, such as Plain Variable dialog or Plain Typedef dialog, click the ··· button.

Result

The new vector or matrix is set.

Note

The status bar informs you about invalid entries. As long as entries are invalid, the OK button is not available. For more information on valid entries, refer to the embedded help.

Dialog settings

Number of rows Lets you specify the number of rows (only for matrices).

Number of columns Lets you specify the number of columns.

Adjust Table Adjusts the column width to the contents.

Status bar Displays invalid settings.

Tip

To navigate within the matrix of the Edit Matrix dialog, use the keys and key combinations known from spreadsheet applications, such as (Shift+) Enter, (Shift+) Tab, and the arrow keys.

Edit String

Purpose	To edit properties of string type.
Access	You can access this dialog in the following ways: ▪ In the **Property Value List** or **Object Explorer**,click the ··· button or double-click the **Value** field. ▪ On the context menu of the selected property, click the **Edit** button . ▪ In various dialogs, such as **Plain Variable** dialog or **Plain Typedef** dialog, click the ··· button.
Result	Properties of string type are edited.

Note

The status bar informs you about invalid entries. As long as entries are invalid, the **OK** button is not available. For more information on valid entries, refer to the embedded help.

Edit StringList

Purpose	To edit properties of the StringList or StringArray type and properties containing paths.
Access	You can access this dialog in the following ways: ▪ In the **Property Value List** or **Object Explorer**,click the ··· button or double-click the **Value** field. ▪ On the context menu of the selected property, click the **Edit** button . ▪ In various dialogs, such as **Plain Variable** dialog or **Plain Typedef** dialog, click the ··· button.
Result	Properties of the StringList or StringArray type and properties containing paths are edited.

Dialog settings

Add

Adds a new value to the list of strings.

Delete

Deletes the selected value from the list of strings.

Move up

Moves the selected value up one position in the list of strings.

Move down

Moves the selected value down one position in the list of strings.

Edit

Opens the Advanced Edit String List dialog. For more information, refer to Advanced Edit String List on page 30.

Related topics

References

Advanced Edit String List..... 30

Edit Value

Purpose

To set a numerical value of a property.

Access

You can access this dialog in the following ways:

- In the Property Value List or Object Explorer,click the ··· button or double-click the Value field.
- On the context menu of the selected property, click the Edit button .
- In various dialogs, such as Plain Variable dialog or Plain Typedef dialog, click the ··· button.

Result

The new numerical value of the property is set.

Note

The status bar informs you about invalid entries. As long as entries are invalid, the OK button is not available. For more information on valid entries, refer to the embedded help.

Related topics

Basics

Basics on Handling Multiple DD Objects with the Object Explorer (📖 TargetLink Data Dictionary Basic Concepts Guide)

Select Object

Purpose

To select a DD reference object.

Access

You can access this dialog:

- by clicking the Edit button ··· in the property value list or the Object Explorer.
- by clicking the Browse button in dialogs like Manage Build, Import Variable Objects or Export Variable Objects.

Result

The new DD reference object is selected.

Dialog settings

Reference Selection Lets you select the DD reference object. The name of the selected DD reference object is displayed in the filter edit field.

Use default as variable/function class Indicates whether the default variable class is selected. The checkbox is available only if you access the dialog for selecting a variable class or function class object.

Find Next Lets you search for the next DD reference object which matches the specified string.

The button's drop-down list provides additional commands:

Command	Description
Find Next	Searches for the next DD reference object which matches the specified string.

Command	Description
Find First	Searches for the top-most DD reference object in the object tree which matches the specified string.
Find Previous	Searches for the previous DD reference object which matches the specified string.

Apply Filter Lets you apply the filter entered in the filter edit field to the DD reference objects displayed in the Reference Selection window.

If the filter is applied, the label on the button turns blue.

The button's drop-down list provides additional commands:

Command	Description
Apply Filter	Applies the filter to the displayed DD reference objects.
Show All	Disables the filter. All DD reference objects are displayed.

Filter edit field Lets you enter a string to filter and/or search the displayed DD reference objects.

The dialog creates a history list of the selected DD reference objects. You can select a previously selected object from the field's drop-down list and use it as filter and/or search string.

According to Microsoft Windows standard, the field's context menu provides commands that let you edit the filter string.

Description field Shows you a description of the currently selected object.

Status bar Displays invalid selections.

Related topics

HowTos

How to Reference Data Dictionary Objects (TargetLink Data Dictionary Basic Concepts Guide)

References

Addfile Block (TargetLink Model Element Reference)
DD Reference Selection Dialog (Property Manager) (TargetLink Tool and Utility Reference)

Object Dialogs

Where to go from here

Information in this section

DDIncludeFile Object Dialog

Purpose

To edit the properties of a **DDIncludeFile** object.

Access

You can access this dialog via:

Menu bar	None
Context menu of	Data Dictionary Navigator
Shortcut key	None
Toolbar button	None
Mouse	Double-click on a **DDIncludeFile** object

Result

The **DDIncludeFile** object is edited.

Description

This dialog opens if the DD object is a **DDIncludeFile** object.

DDIncludeFile options

Name Displays the name of the selected DD object.

DDIncludeFile object Displays the path of the DDIncludeFile object that contains the inclusion point data. If not yet created, <new> is displayed.

Point of inclusion object kind Displays the object kind of the object you selected to make a Point of Inclusion.

Include file Lets you specify the path of the partial DD file to be included.

Include file path specification Lets you specify how the file path is stored in the DDIncludeFile. The following options are available:

- Absolute path
- File name only
- Path relative to current working directory
- Path relative to the DDINCLUDE
- Path relative to main DD directory (using the MainDDDir macro)

Load include file with main DD Lets you specify whether the included DD file is automatically loaded when loading the main DD file.

Save include file with main DD Lets you specify the saving behavior of the included DD file (autoload). The following options are available:

- Off
- On
- OnlyIfModified
- PromptIfmodified

Description Lets you enter a description for the DDIncludeFile object that contains the inclusion point data.

Load include file on OK Lets you specify whether the partial DD file is loaded (in overwrite mode) into the Point of Inclusion after clicking OK.

Save include file on OK Lets you specify whether the partial DD file is saved after clicking OK.

Include file information (read-only) Provides information about the included DD file:

- Absolute path of the included DD file
- File root object name
- File root object kind
- Parent object
- Data Model Revision

Related topics

HowTos

How to Include Partial Data Dictionary Files (📖 TargetLink Data Dictionary Basic Concepts Guide)

References

Point of Inclusion..... 109

Enum Typedef

Purpose

To edit the properties of a enum Typedef object.

Access

You can access this dialog via:

Menu bar	None
Context menu of	Data Dictionary Navigator
Shortcut key	None
Toolbar button	None
Mouse	Double-click on a enum Typedef object

Result

The (new) enum Typedef object is edited. The Typedef object is valid according to the data model. Obsolete properties and child objects are deleted.

Description

This dialog opens if the Typedef object is a enum base type.

The properties are not written to the Typedef object unless you have corrected all invalid settings.

Dialog settings

BaseType Lets you select a standard base type. If you select a plain base type, the Plain Typedef dialog opens.

Implementation Lets you select how enumerations are implemented in the generated code (enumeration versus macro). An empty field evaluates to Enumeration during code generation.

Tag Lets you enter the tag name of the C enum.

Create typedef statement Indicates whether a typedef statement will be integrated in your code.

Module Lets you specify the file name of the module where the typedef is declared (without extension).

Name Template Lets you specify the name template for the enumeration elements, e.g., CHARTMODE_$V.

Default Lets you specify the default enumeration element. An empty field evaluates to the first/top enumeration element during code generation.

Data Type Lets you specify the integral type of the enumeration data type (especially when generated as a C code macro), e.g., Int8.

Omit Cast of Macro Lets you specify whether type casts are generated into the macros representing the enumeration elements.

Name Lets you enter the name of the enumeration element, e.g., Blue.

Value Lets you enter the integer value of the enumeration element, e.g., 0.

Description (Enum Elements group box) Lets you enter a description of the enumeration element.

Add component

Adds a new component to the list of components.

Delete

Deletes the selected component from the list of components.

Move up

Moves the selected component up one position in the list of components.

Move down

Moves the selected component down one position in the list of components.

Description (Documentation group box) Lets you enter a description of the typedef.

Status bar Displays invalid properties.

Related topics

HowTos

How to Explicitly Specify Structured Data Types (TargetLink Customization and Optimization Guide)

Examples

Example of Creating New Typedefs (TargetLink Data Dictionary Basic Concepts Guide)

References

Plain Typedef 53

Interpolation Block Object Dialog

Purpose

To edit the properties of a DD Interpolation Block object.

Access

You can access this dialog via:

Menu bar	None
Context menu of	Data Dictionary Navigator
Shortcut key	None
Toolbar button	None
Mouse	Double-click on a DD Interpolation Block object

Result

The DD Interpolation Block object is edited.

Description

This dialog opens if the DD Block object is a DD Interpolation Block object.

Look-up options

Number of table dimensions Lets you choose the table dimension. The maximum is two.

Apply interpolation Indicates whether interpolation is applied in the production code. This increases the execution time and the table accuracy.

Distances between table entries less than half of implemented range Indicates whether differences between adjacent table entries fit into the associated signed data type. If you use this setting to reduce code memory

consumption, you should be careful because look-up tables are often subject to calibration. This setting also applies to all vectors and matrices of a block. For details, refer to the example in Basics on Applying Output Calculation Methods (Lookup Methods) (📖 TargetLink Preparation and Simulation Guide).

Related Variable objects

Table Lets you select a DD Variable object.

NumAxisPoints(#2) Lets you select a DD Variable object.
Grayed out, unless Number of table dimensions is set to 2.
Specify the DD Variable object as shown in the following table:

Property	Value
Class	MERGEABLE_GLOBAL[1]
Type	Unsigned integer
Value	The width of the corresponding axis as scalar.[2]

[1] This lets you use the variable at different look-up table objects.
[2] Must match the value of the axis variable's Width property.

InputIndex Lets you select a DD Variable object.

InputIndex(#2) Lets you select a DD Variable object.
Grayed out, unless Number of table dimensions is set to 2.

InputFraction Lets you select a DD Variable object.

InputFraction(#2) Lets you select a DD Variable object.
Grayed out, unless Number of table dimensions is set to 2.

Command buttons

OK Lets you close the dialog, confirming all changes.

Cancel Lets you close the dialog, canceling all changes.

Help Lets you open this document.

Related topics

Basics

Basics on Controlling the Representation of Look-Up-Table-Related Variables in the A2L File (📖 TargetLink Interoperation and Exchange Guide)
Basics on Specifying the Variables to Export to A2L Files (📖 TargetLink Interoperation and Exchange Guide)
CG_FROM_DD (📖 TargetLink Demo Models)

HowTos

How to Create DD Look-Up Table Objects (📖 TargetLink Preparation and Simulation Guide)
How to Specify DD Look-Up Table Objects (📖 TargetLink Preparation and Simulation Guide)

Look-up Table (1-D) Block Object Dialog

Purpose

To edit the properties of a DD Look-up Table (1-D) block object.

Access

You can access this dialog via:

Menu bar	None
Context menu of	Data Dictionary Navigator
Shortcut key	None
Toolbar button	None
Mouse	Double-click on a DD Look-up Table (1-D) block object

Result

The DD Look-up Table (1-D) block object is edited.

Description

This dialog opens if the DD Block object is a DD Look-up Table (1-D) block object.

Look-up options

Look-up method Lets you select the look-up method you want to use:

Look-Up Method	Description	Out-Of-Range Handling
Interpolation – Extrapolation	Uses the two elements of the argument vector that wrap the input value. A linear interpolation is applied to compute the output value.	Extrapolation based on the two adjacent data points 
Interpolation – Use End Values	Uses the two elements of the argument vector that wrap the input value. A linear interpolation is applied to compute the output value.	Saturation to the adjacent data point 
Use Input Nearest	Uses the element of the argument vector that is nearest to the input value	Saturation to the adjacent data point 
Use Input Below	Uses the element of the argument vector that is below the input value.	Saturation to the adjacent data point 
Use Input Above	Uses the element of the argument vector that is above the input value	Saturation to the adjacent data point 

Distances between table entries less than half of implemented range Indicates whether differences between adjacent table entries fit into the associated signed data type. If you use this setting to reduce code memory consumption, you should be careful because look-up tables are often subject to calibration. This setting also applies to all vectors and matrices of a block. For details, refer to the example in Basics on Applying Output Calculation Methods (Lookup Methods) (🕮 TargetLink Preparation and Simulation Guide).

Generate map struct Indicates whether a map structure is generated for the table.

Axis - Search method Lets you select the search method you want to use:

Search Method	Result
Linear search, start low	A linear search is implemented from the lowest end of the abscissa. The algorithm will go through the abscissa entries until a value higher than the input value is found. This search is simple and fast if the table is not too large.
Linear search, start high	A linear search is implemented from the highest end of the abscissa. The algorithm will go through the abscissa entries until a value lower than the input value is found. This search is simple and fast if the table is not too large.
Local search	The search starts at the index where the previous table value was found. Local searches can be slightly slower than linear searches due to the more complex code. This is recommended when the table inputs are smooth and do not change abruptly.
Binary search	The search area is divided into two halves, depending on whether the input value is in the upper or lower part. The iterations are repeated until the adjacent entries of the specified input value are found. This search is fast, especially in large tables, but the code is more complex.
Equidistant implementation	Only the abscissa parameters (Distance, From, and To) and the corresponding ordinate entries have to be generated in the production code, which results in less memory consumption.

For details on the search method, refer to Details on Argument Vectors (🕮 TargetLink Preparation and Simulation Guide).

Axis - Use boundary points (For integer data types only). Indicates whether boundary points are added in the production code. These are table entries inserted at the implemented range boundaries of the abscissa. You can reduce the code size and speed up the table search by using boundary points.

Axis - Arbitrary factor (For integer data types only). Indicates whether the distance between the two points is an arbitrary LSB multiple. When cleared, the distance is a power-of-two LSB multiple.

Grayed out, unless **Equidistant implementation** is selected as look-up method.

Related Variable objects

Table Lets you select a DD Variable object.

Axis (input) Lets you select a DD Variable object.

Grayed out, if the **Equidistant implementation** look-up method is selected.

Number of points axis (input) Lets you select a DD Variable object.

Specify the DD Variable object as shown in the following table:

Property	Value
Class	MERGEABLE_GLOBAL[1)]
Type	Unsigned integer
Value	The width of the corresponding axis as scalar.[2)]

1) This lets you use the variable at different look-up table objects.
2) Must match the value of the axis variable's Width property.

Distance of axis (input) Lets you select a DD Variable object.

Grayed out, unless the Equidistant implementation look-up method is selected.

Start point of axis (input) Lets you select a DD Variable object.

Grayed out, unless the Equidistant implementation look-up method is selected.

Output Lets you select a DD Variable object.

Map struct Lets you select a DD Variable object.

Grayed out, unless the Generate map struct checkbox is selected.

Command buttons

OK Lets you close the dialog, confirming all changes.

Cancel Lets you close the dialog, canceling all changes.

Help Lets you open this document.

Related topics

Basics

Basics on Controlling the Representation of Look-Up-Table-Related Variables in the A2L File (TargetLink Interoperation and Exchange Guide)
Basics on Specifying the Variables to Export to A2L Files (TargetLink Interoperation and Exchange Guide)
CG_FROM_DD (TargetLink Demo Models)

HowTos

How to Create DD Look-Up Table Objects (TargetLink Preparation and Simulation Guide)
How to Specify DD Look-Up Table Objects (TargetLink Preparation and Simulation Guide)

Look-up Table (2-D) Block Object Dialog

Purpose

To edit the properties of a DD Look-up Table (2-D) block object.

Access

You can access this dialog via:

Menu bar	None
Context menu of	Data Dictionary Navigator
Shortcut key	None
Toolbar button	None
Mouse	Double-click on a DD Look-up Table (2-D) block object.

Result

The DD Look-up Table (2-D) block object is edited.

Description

This dialog opens if the DD Block object is a DD Look-up Table (2-D) block object.

Look-up options

Look-up method Lets you select the look-up method you want to use:

Look-Up Method	Description	Out-Of-Range Handling
Interpolation – Extrapolation	Uses the two elements of the argument vector that wrap the input value. A linear interpolation is applied to compute the output value.	Extrapolation based on the two adjacent data points 
Interpolation – Use End Values	Uses the two elements of the argument vector that wrap the input value. A linear interpolation is applied to compute the output value.	Saturation to the adjacent data point 
Use Input Nearest	Uses the element of the argument vector that is nearest to the input value	Saturation to the adjacent data point 
Use Input Below	Uses the element of the argument vector that is below the input value.	Saturation to the adjacent data point 
Use Input Above	Uses the element of the argument vector that is above the input value	Saturation to the adjacent data point 

Distances between table entries less than half of implemented range Indicates whether differences between adjacent table entries fit into the associated signed data type. If you use this setting to reduce code memory consumption, you should be careful because look-up tables are often subject to calibration. This setting also applies to all vectors and matrices of a block. For details, refer to the example in Basics on Applying Output Calculation Methods (Lookup Methods) (🕮 TargetLink Preparation and Simulation Guide).

Generate map struct Indicates whether a map structure is generated for the table.

Axis - Search method Lets you select the search method you want to use:

Search Method	Result
Linear search, start low	A linear search is implemented from the lowest end of the abscissa. The algorithm will go through the abscissa entries until a value higher than the input value is found. This search is simple and fast if the table is not too large.
Linear search, start high	A linear search is implemented from the highest end of the abscissa. The algorithm will go through the abscissa entries until a value lower than the input value is found. This search is simple and fast if the table is not too large.
Local search	The search starts at the index where the previous table value was found. Local searches can be slightly slower than linear searches due to the more complex code. This is recommended when the table inputs are smooth and do not change abruptly.
Binary search	The search area is divided into two halves, depending on whether the input value is in the upper or lower part. The iterations are repeated until the adjacent entries of the specified input value are found. This search is fast, especially in large tables, but the code is more complex.
Equidistant implementation	Only the abscissa parameters (Distance, From, and To) and the corresponding ordinate entries have to be generated in the production code, which results in less memory consumption.

For details on the search method, refer to Details on Argument Vectors (🕮 TargetLink Preparation and Simulation Guide).

Axis - Use boundary points (For integer data types only). Indicates whether boundary points are added in the production code. These are table entries inserted at the implemented range boundaries of the abscissa. You can reduce the code size and speed up the table search by using boundary points.

Axis - Arbitrary factor (For integer data types only). Indicates whether the distance between the two points is an arbitrary LSB multiple. When cleared, the distance is a power-of-two LSB multiple.

Grayed out, unless **Equidistant implementation** is selected as look-up method.

Related Variable objects

Table Lets you select a DD Variable object.

Axis#1 (row) Lets you select a DD Variable object.

Grayed out, if the **Equidistant implementation** look-up method is selected.

Axis#2 (column) Lets you select a DD Variable object.
Grayed out, if the Equidistant implementation look-up method is selected.

Number of points axis#1 (row) Lets you select a DD Variable object.
Specify the DD Variable object as shown in the following table:

Property	Value
Class	MERGEABLE_GLOBAL[1)]
Type	Unsigned integer
Value	The width of the corresponding axis as scalar.[2)]

1) This lets you use the variable at different look-up table objects.
2) Must match the value of the axis variable's Width property.

Start point of axis#2 (column) Lets you select a DD Variable object.
Grayed out, unless the Equidistant implementation look-up method is selected.

Distance of axis#1 (row) Lets you select a DD Variable object.
Grayed out, unless the Equidistant implementation look-up method is selected.

Distance of axis#2 (column) Lets you select a DD Variable object.
Grayed out, unless the Equidistant implementation look-up method is selected.

Start point of axis#1 (row) Lets you select a DD Variable object.
Grayed out, unless the Equidistant implementation look-up method is selected.

Start point of axis#2 (column) Lets you select a DD Variable object.
Grayed out, unless the Equidistant implementation look-up method is selected.

Max. of axis#1 (row) Lets you select a DD Variable object.
Grayed out, unless the Equidistant implementation look-up method is selected.

Max. of axis#2 (column) Lets you select a DD Variable object.
Grayed out, unless the Equidistant implementation look-up method is selected.

Output Lets you select a DD Variable object.

Map struct Lets you select a DD Variable object.
Grayed out, unless the Generate map struct checkbox is selected.

Command buttons

OK Lets you close the dialog, confirming all changes.

Cancel Lets you close the dialog, canceling all changes.

Help Lets you open this document.

Related topics

Basics

Basics on Controlling the Representation of Look-Up-Table-Related Variables in the A2L File (📖 TargetLink Interoperation and Exchange Guide)
Basics on Specifying the Variables to Export to A2L Files (📖 TargetLink Interoperation and Exchange Guide)
CG_FROM_DD (📖 TargetLink Demo Models)

HowTos

How to Create DD Look-Up Table Objects (📖 TargetLink Preparation and Simulation Guide)
How to Specify DD Look-Up Table Objects (📖 TargetLink Preparation and Simulation Guide)

Manage Messages

Purpose

To edit the properties of an RTOS Message object.

Access

You can access this dialog via:

Menu bar	None
Context menu of	Data Dictionary Navigator
Shortcut key	None
Toolbar button	None
Mouse	Double-click on an RTOS Message object

Result

The RTOS Message object is saved with the new properties.

Description

The properties are not written to the message object unless you have corrected all invalid settings.

Dialog settings (Message group box)

Message Lets you select a message object beneath /Pool/RTOS/Messages in the TargetLink Data Dictionary.

Rename Lets you rename the message object selected in the Message list.

Add Lets you add a new message object to the TargetLink Data Dictionary.

Delete Lets you delete the message you selected in the Message list.

Description Lets you delete the message object selected in the Message list.

Message Kind Lets you select one of the following message kinds:

Message Kind	Description
GLOBAL_BUFFER	TargetLink copies the calculated data from a local copy to a global buffer. The global buffer can have one of the following shapes: ▪ Each signal is assigned to a separate variable. ▪ The global buffer is a structure variable and each signal is assigned to a separate component of the variable.
OSEK_MESSAGE	TargetLink uses the OSEK API function SendMessage (<Msg>, <AccessNameRef>) to send the data from the sender's local copy to a separate buffer. The receiver reads the data from the buffer using the OSEK API function ReceiveMessage (<Msg>, <AccessNameRef>).
SWITCHED_BUFFER	TargetLink does not support switched buffer.

Advanced options Indicates that specific edit fields in this dialog are enabled. If this checkbox is not selected, the Name, Class, and Protection Mechanism edit fields in the SendAccessor and the ReceiveAccessor group boxes are disabled. The edit fields in the Message Variables group box are also disabled.

Dialog settings (Message Variables group box)

Message Variable Lets you select a message variable object beneath the current message object in /Pool/RTOS/Messages in the TargetLink Data Dictionary. TargetLink uses this variable to store the message data.

Add Lets you add a message variable object.

Delete Lets you delete the message variable object selected in the Message Variable list.

Variable Lets you select a C variable from the TargetLink Data Dictionary for mapping to the current message variable object.

Note

The contents of the Name, Class, and Type edit fields are inherited from the Variable property. If you specify the Variable property, these fields are disabled. To specify the name, class, and type independently, you have to clear the Variable edit field.

Name Lets you select a pattern to specify the message variable name.

Class Lets you specify the variable class of the C variable.

Type Lets you specify the data type of the C variable.

Dialog settings (Send Accessor group box)

Local Copy Lets you specify to use a local copy at the subsystem sending a message. If you do, TargetLink creates a local copy of the message variable at the sender. The sender uses the local copy to calculate its signals. After the calculation is completed, the variables are copied to the message variable.

Protection Mechanism Lets you specify a mechanism to protect a message from being accessed during transfer.

Name Lets you specify the name of the message's local copy used at the sender.

Class Lets you specify the variable class of the message's local copy used at the sender.

Dialog settings (Receive Accessor group box)

Receiver Lets you specify the receiving subsystem of a message by selecting a ReceiveAccessor object beneath the current message variable object in the TargetLink Data Dictionary. If the DisplayName property of a ReceiveAccessor object contains a name, the Receiver field displays that name. Otherwise, the name of the ReceiveAccessor object is shown.

Local Copy Lets you specify to use a local copy at the subsystem receiving a message. If you do, TargetLink creates a local copy of the message variable at the receiver. The receiver uses the local copy to calculate its signals.

Protection Mechanism Lets you specify a mechanism to protect a message from being accessed during transfer.

Name Lets you specify the name of the message's local copy used at the receiver.

Class Lets you specify the variable class of the message's local copy used at the receiver.

Plain Typedef

Purpose

To edit the properties of a plain Typedef object.

Access

You can access this dialog via:

Menu bar	None
Context menu of	Data Dictionary Navigator
Shortcut key	None
Toolbar button	None
Mouse	Double-click on a plain Typedef object

Result	The (new) plain Typedef object is edited. The Typedef object is valid according to the data model. Obsolete properties and child objects are deleted.
Description	This dialog opens if the Typedef object is a plain Typedef object. The dialog looks different depending on whether you edit a base type or a user-defined Typedef object. The properties are not written to the Typedef object unless you have corrected all invalid properties.
Dialog settings (base type)	**BaseType** Lets you select a standard base type. If you select a struct base type, the Struct Typedef dialog opens. **Rename** Lets you select the Typedef object that represents the renamed base type. **Cut** Lets you enter a shortcut for the type name. **Description** Lets you enter a description of the Typedef object.
Dialog settings (user-defined typedef)	**BaseType** Lets you select a standard base type. If you select a struct base type, the Struct Typedef dialog opens. **Module** Lets you specify the name of the module where the Typedef object is declared (without extension). **Create typedef statement** Indicates whether a typedef statement will be integrated in your code. **Scaling** Lets you select a scaling reference for the constraints node. **Max** Lets you enter a maximum value for the constraints node when a Scaling object is selected. **Min** Lets you enter a minimum value for the constraints node when a Scaling object is selected. **Description** Lets you enter a description of the Typedef object. **Status bar** Displays invalid properties.

Related topics

HowTos

How to Create User-Defined Data Types (📖 TargetLink Customization and Optimization Guide)

Examples

Example of Creating New Typedefs (📖 TargetLink Data Dictionary Basic Concepts Guide)

References

Struct Typedef..... 62

Plain Variable

Purpose

To edit the properties of a plain **Variable** object.

Access

You can access this dialog via:

Menu bar	None
Context menu of	Data Dictionary Navigator
Shortcut key	None
Toolbar button	None
Mouse	Double-click on a **Variable** object

Result

A plain **Variable** object is edited.

Description

The properties are written to the Variable object only if the settings are valid according to the data model.

Note

The dialog provides different user interfaces, depending on the features you use:

- Data variants (Basics on Coding Styles for Data Variants (🕮 TargetLink Customization and Optimization Guide))
- Code variants (How to Specify Code Variants (🕮 TargetLink Customization and Optimization Guide))
- Variable vector width (Basics on Variable Vector Widths (🕮 TargetLink Customization and Optimization Guide))

Value Options

Data Variant Lets you select a variant configuration.

Value Displays the specified value. Depending on the context, the dialog lets you change it.

Variable vector width Lets you enter the variable vector width.

Variant Displays the specified variant. Depending on the context, the dialog lets you change it.

Width Displays the specified width. Depending on the context, the dialog lets you change it.

Code Options

Class Lets you select the VariableClass object from the DD object tree.

Max Lets you enter a maximum value for the variable.

Min Lets you enter a minimum value for the variable.

Module Lets you specify in which file a variable is defined during code generation.

Name Lets you enter the name of the variable in the generated code. You can enter any string that complies with the conventions for C identifiers. The string can also contain name macros.Refer to Basics on Using Name Macros (🕮 TargetLink Customization and Optimization Guide).

Type Lets you select a Typedef object for the variable from the DD object tree. If you select a struct or implicit struct typedef, the Struct Variable dialog opens.

Scaling Options

LSB Lets you enter the least significant bit.

Offset Lets you enter an offset value for the variable.

Scaling Lets you select a Scaling object for the variable from the DD object tree.

Scaling Range Displays the implemented range of the variables.

Unit Lets you specify a physical unit.

Use local scaling Indicates whether the formula used for scaling is defined locally at the Variable object.

Documentation

Description Lets you enter a description of the variable.

Related topics

Basics

Basics on Using Name Macros (📖 TargetLink Customization and Optimization Guide)

HowTos

How to Set Properties of Data Dictionary Objects (📖 TargetLink Data Dictionary Basic Concepts Guide)

References

Struct or Implicit Struct Variable..... 59

PreLook-Up Index Search Block Object Dialog

Purpose

To edit the properties of a DD PreLook-Up Index Search Block object.

Access

You can access this dialog via:

Menu bar	None
Context menu of	Data Dictionary Navigator
Shortcut key	None
Toolbar button	None
Mouse	Double-click on a DD PreLook-Up Index Search Block object

Result The DD PreLook-Up Index Search Block object is edited.

Description This dialog opens if the DD Block object is a DD PreLook-Up Index Search Block object.

Look-up options

Output only the index Indicates whether only the index is output.

Axis - Search method Lets you select the search method you want to use:

Search Method	Result
Linear search, start low	A linear search is implemented from the lowest end of the abscissa. The algorithm will go through the abscissa entries until a value higher than the input value is found. This search is simple and fast if the table is not too large.
Linear search, start high	A linear search is implemented from the highest end of the abscissa. The algorithm will go through the abscissa entries until a value lower than the input value is found. This search is simple and fast if the table is not too large.
Local search	The search starts at the index where the previous table value was found. Local searches can be slightly slower than linear searches due to the more complex code. This is recommended when the table inputs are smooth and do not change abruptly.
Binary search	The search area is divided into two halves, depending on whether the input value is in the upper or lower part. The iterations are repeated until the adjacent entries of the specified input value are found. This search is fast, especially in large tables, but the code is more complex.

For details on the search method, refer to Details on Argument Vectors (📖 TargetLink Preparation and Simulation Guide).

Related Variable objects

Axis (breakpoint data) Lets you select a DD Variable object.

Number of points axis (breakpoint) Lets you select a DD Variable object.

Specify the DD Variable object as shown in the following table:

Property	Value
Class	MERGEABLE_GLOBAL [1)]
Type	Unsigned integer
Value	The width of the corresponding axis as scalar.[2)]

1) This lets you use the variable at different look-up table objects.
2) Must match the value of the axis variable's Width property.

Fraction Lets you select a DD Variable object.

Not available, if the Output only the index on page 58 checkbox is selected.

Index Lets you select a DD Variable object.

Command buttons

OK Lets you close the dialog, confirming all changes.

Cancel Lets you close the dialog, canceling all changes.

Help Lets you open this document.

Related topics

Basics

Basics on Controlling the Representation of Look-Up-Table-Related Variables in the A2L File (📖 TargetLink Interoperation and Exchange Guide)
Basics on Specifying the Variables to Export to A2L Files (📖 TargetLink Interoperation and Exchange Guide)
CG_FROM_DD (📖 TargetLink Demo Models)

HowTos

How to Create DD Look-Up Table Objects (📖 TargetLink Preparation and Simulation Guide)
How to Specify DD Look-Up Table Objects (📖 TargetLink Preparation and Simulation Guide)

Struct or Implicit Struct Variable

Purpose

To edit the properties of a struct or implicit struct **Variable** object. In addition, you can automatically create variable components according to the components in the struct data type that the variable is based on.

Access

You can access these dialogs via:

Menu bar	None
Context menu of	Data Dictionary Navigator
Shortcut key	None
Toolbar button	None
Mouse	Double-click on a **Variable** object

Result

A struct or implicit struct **Variable** object is edited. The **Variable** object is valid according to the data model. Obsolete properties and child objects are deleted.

Description

If you edit a struct **Variable** object, the edit fields and buttons for editing components of implicit struct **Variable** objects are disabled. The properties are not written to the **Variable** object unless you have corrected all invalid settings.

If you assign a struct data type to a Variable object, TargetLink automatically references all its struct components and creates corresponding variable components when you click OK to close the dialog.

Dialog settings

Name (Code Options group box) Lets you specify the name of the variable in the generated code. You can enter any string that complies with the conventions for C identifiers. The string can also contain name macros. Refer to Basics on Using Name Macros (🕮 TargetLink Customization and Optimization Guide).

Module Lets you specify in which file a variable is defined.

Class (Code Options group box) Lets you select a VariableClass object from the DD object tree to specify the variable's C attributes.

Type (Code Options group box) Lets you select a Typedef object of the variable from the DD object tree. If you select a plain typedef, the Plain Variable dialog opens.

Show LSB/Offset Indicates that the LSB and Offset columns are displayed in the Components list.

Show Min/Max Indicates that the Min and Max columns are displayed in the Components list.

Name (Components group box) Displays/lets you enter the name of the component.

Class (Components group box) Lets you select the VariableClass object of the component.

Type (Components group box) Displays/lets you enter the type of the component.

Scaling Lets you select the Scaling object of the component.

LSB Displays/lets you enter the least significant bit of the component.

Offset Displays/lets you enter the offset value of the component.

Value Lets you enter the value of the component.

Width Displays/lets you enter the width of the component.

Min Lets you enter a minimum value for the component.

Max Lets you enter a maximum value for the component.

SortBy Lets you enter the criteria for sorting dynamic components.

Description (Components group box) Lets you enter a description of the component.

Add component

Adds a new component to the list of components.

Add dynamic component

Adds a new dynamic component to the list of components.

Delete

Deletes the selected component from the list of components.

Move up

Moves the selected component up one position in the list of components.

Move down

Moves the selected component down one position in the list of components.

Description (Documentation group box) Lets you enter a description of the variable.

Status bar Displays invalid properties.

Related topics

Basics

Basics on Arranging Data in Structured Variables (TargetLink Customization and Optimization Guide)
Basics on Using Name Macros (TargetLink Customization and Optimization Guide)

HowTos

How to Create Explicitly Specified Structured Variables (TargetLink Customization and Optimization Guide)
How to Create Implicitly Specified Structured Variables (TargetLink Customization and Optimization Guide)

Examples

Example of Collecting Gain Parameters via Dynamic Components (TargetLink Customization and Optimization Guide)
Example of Creating New Variables (TargetLink Data Dictionary Basic Concepts Guide)

References

Plain Variable..... 55

Struct Typedef

Purpose

To edit the properties of a struct Typedef object.

Access

You can access this dialog via:

Menu bar	None
Context menu of	Data Dictionary Navigator
Shortcut key	None
Toolbar button	None
Mouse	Double-click on a struct Typedef object

Result

The (new) struct Typedef object is edited. The Typedef object is valid according to the data model. Obsolete properties and child objects are deleted.

Description

This dialog opens if the Typedef object is a struct base type.

The properties are not written to the Typedef object unless you have corrected all invalid settings.

Dialog settings

BaseType Lets you select a standard base type. If you select a plain base type, the Plain Typedef dialog opens.

Tag Lets you enter the internal tag name of the structure.

Create typedef statement Indicates whether a typedef statement will be integrated in your code.

Module Lets you specify the file name of the module where the typedef is declared (without extension).

Prefix Lets you enter a prefix to be added to a variable definition.

Qualifier Lets you enter the type qualifier.

Type Lets you select a typedef.

Name Lets you enter the name of the property.

Width Lets you enter the desired width of the property.

Description (Components group box) Lets you enter a description of the component.

Add component

Adds a new component to the list of components.

Delete

Deletes the selected component from the list of components.

Move up

Moves the selected component up one position in the list of components.

Move down

Moves the selected component down one position in the list of components.

Description (Documentation group box) Lets you enter a description of the typedef.

Status bar Displays invalid properties.

Related topics

HowTos

How to Explicitly Specify Structured Data Types (📖 TargetLink Customization and Optimization Guide)

Examples

Example of Creating New Typedefs (📖 TargetLink Data Dictionary Basic Concepts Guide)

References

Plain Typedef..... 53

Commands

Where to go from here

Information in this section

AUTOSAR-Related Commands

Where to go from here

Information in this section

Adjust to Typedef

Purpose

To synchronize DD **Variable** objects or DD objects that specify AUTOSAR data prototypes with their referenced DD **Typedef** object.

Access

You can access this command via:

Menu bar	None
Context menu of	Variable, CalPrmElement, CalPrmInterface, ClientServerInterface, DataElement, InterRunnableVariable, ModeSwitchInterface, NvDataElement, NvDataInterface, OperationArgument objects
Shortcut key	None
Toolbar button	None

MATLAB API functions

Refer to the corresponding MATLAB API functions:

DD Object	API Function
CalPrmElement	`[errorCode,processedObjectInfo] = dsdd('SyncType',<objectIdentifier>);`
DataElement	
InterRunnableVariable	
NvDataElement	
OperationArgument	
Variable	`[errorCode,processedObjectInfo] = dsdd('SyncType',<objectIdentifier>[,<variantID>]);`
ClientServerInterface	`[bSuccess,processedObjectInfo] = dsdd('SyncTypes',<objectIdentifier>[,<attributeName>,<attributeValue>]);`
CalPrmInterface	
ModeSwitchInterface	
NvDataInterface	

Result

One or more DD objects are synchronized with the DD Typedef object that they reference.

ApplicationDataType Creation Wizard Command

Purpose

To create application data types (ADTs), reference them at data prototypes, and map the application data types to implementation data types (IDTs).

Access

You can access this command via:

Menu bar	None
Context menu of	DD SoftwareComponent objects - AUTOSAR Tools
Shortcut key	None
Toolbar button	None

Result

TargetLink performs the following:

- Creating the specified application data types.
- Referencing the application data types at the respective data prototypes.
- Creating data type maps between the types that are referenced at the data prototypes.

Description

The ApplicationDataType Creation Wizard lets you step through data prototypes of the selected software component that reference no application data types.

The wizard allows you to review data prototype and typedef (implementation data type) settings and proposes settings for an application data type. You can specify application data types and let the wizard create and reference the application data types.

ApplicationDataType Creation Wizard

Software component Displays the currently selected DD SoftwareComponent object.

Application data type group Displays the DD ApplicationDataTypeGroup object that DD ApplicationDataType objects will be stored in.

TargetLink proposes a default group named `ADTGroup_<SoftwareComponent>`. Edit it as required.

Data type mapping set Displays the DD DataTypeMappingSet object referenced at the DD SoftwareComponent object.

If no data type mapping set object is referenced, TargetLink creates a default data type mapping set object named `DTMS_<SoftwareComponent>.`

Data type mapping set group Displays the DD DataTypeMappingSetGroup object that contains the DD DataTypeMappingSet object referenced at the DD SoftwareComponent object.

If no data type mapping set object is referenced, TargetLink creates a default data type mapping set group object named `<SoftwareComponent>_Group`, containing a default data type mapping set object named `DTMS_<SoftwareComponent>.`

DD Object Kind of selected data prototype Displays the DD Object Kind of the currently selected data prototype.

Progress bar Visualizes the position of the currently selected data prototype in the list.

<PosInList> of <ItemsInList> Displays the position of the currently selected data prototype in the list and the total number of data prototypes contained in this list.

Note

The ApplicationDataType Creation Wizard lets you step through only the following data prototypes referenced at the software component:

- Data prototypes that do not reference an application data type
- Data prototypes that you did not exclude from the list by clicking the Exclude button

It is best practice to create all ADTs of one software component in one session.

DataPrototype

Name Lets you select data prototypes to create application data types for.

Scaling Displays the value of the property Scaling contained in the selected data prototype's Property Value List.

Max Displays the value of the property Max contained in the selected data prototype's Property Value List.

Min Displays the value of the property Min contained in the selected data prototype's Property Value List.

Width Displays the value of the property Width contained in the selected data prototype's Property Value List.

Typedef

Name Displays the DD Typedef object currently referenced at the selected data prototype

Scaling Displays the value of the property Scaling contained in the Property Value List of the typedef currently referenced at the selected data prototype.

Max Displays the value of the property Max contained in the Property Value List of the typedef currently referenced at the selected data prototype.

Min Displays the value of the property Min contained in the Property Value List of the typedef currently referenced at the selected data prototype.

Width Displays the value of the property Width contained in the Property Value List of the typedef currently referenced at the selected data prototype.

ApplicationDataType (ADT)

Name Lets you specify the name of the DD ApplicationDataType object to be created. TargetLink proposes a name that is generated following the schema `ADT_<Typedef>` that you can change as required.

Scaling Displays the proposed scaling of the DD ApplicationDataType object to be created. Change it as required.

Max Displays the proposed max value of the DD ApplicationDataType object to be created. Change it as required.

Min Displays the proposed min value of the DD ApplicationDataType object to be created. Change it as required.

Width Displays the proposed width of the DD ApplicationDataType object to be created.

Command buttons

Back/Next Lets you select the previous/next data prototype.

Exclude Lets you exclude the currently selected data prototype from the list. TargetLink creates application data types only for data prototypes of the list.

Finish Lets TargetLink perform the following:

- Creating the application data types for all the data prototypes in the list including application data types for composite data prototypes.
- Referencing the application data types at the respective data prototypes.
- Creating data type maps between the types that are referenced at the data prototypes.
- Closing the dialog.

Cancel Lets you cancel all changes and close the wizard.

Help Lets you open this document.

Related topics

HowTos

How to Create Application Data Types for Existing Implementation Data Types (Classic AUTOSAR) (TargetLink Classic AUTOSAR Modeling Guide)

AUTOSAR

Purpose

To specify Variable objects and Typedef objects with AUTOSAR-compliant settings.

Access

You can access this command via:

Menu bar	Edit
Context menu of	DD Variable + DD Typedef objects
Shortcut key	None
Toolbar button	None

Commands for Variable objects

Create PIM Creates Per Instance Memories variables.

For details, refer to Basics on Per Instance Memories (TargetLink Classic AUTOSAR Modeling Guide).

Create SharedCalPrm Creates shared calprm variables.

For details, refer to How to Model Shared Parameters for Calibration (TargetLink Classic AUTOSAR Modeling Guide).

Create composite SharedCalPrm Creates composite shared calprm variables.

For details, refer to How to Model Shared Parameters for Calibration (TargetLink Classic AUTOSAR Modeling Guide).

Create PerInstanceCalPrm Creates instance-specific calibration parameters.

For details, refer to How to Model Per Instance Parameters for Calibration (📖 TargetLink Classic AUTOSAR Modeling Guide).

Command for Typedef objects

Create AUTOSAR Array Type Creates new AUTOSAR array type typedef, that contains the following properties:

- AutosarArrayWidth
- AutosarArrayElementRef
- ModuleRef

Convert into AUTOSAR Array Type Converts an existing typedef into an AUTOSAR array type by adding the following properties:

- AutosarArrayWidth
- AutosarArrayElementRef
- ModuleRef

Result

You have specified Variable objects and Typedef objects with AUTOSAR-compliant settings.

Create DefinedArguments

Purpose

To add the PortDefinedArguments object to a ServerPort object's subtree.

Access

You can access this command via:

Menu bar	None
Context menu of	ServerPort objects
Shortcut key	None
Toolbar button	None

MATLAB API function

Refer to the corresponding MATLAB API function:

```
[hDDObject,errorCode] =
dsdd('CreatePortDefinedArguments',<objectIdentifier>);
```

Result

The PortDefinedArguments object is added to the ServerPort object's subtree.

Generate Frame Model

Purpose

To generate an AUTOSAR frame model for software components.

Access

You can access this command via:

Menu bar	None
Context menu of	DD SoftwareComponent objects - AUTOSAR Tools DD SoftwareComponentGroup objects - AUTOSAR Tools
Shortcut key	None
Toolbar button	None

Result

If started from the AUTOSAR Tools menu on the context menu of a

- DD SoftwareComponent object, TargetLink generates a frame model for the selected software component(s)
- DD SoftwareComponentGroup object, TargetLink generates a frame model for all software components contained in the selected group(s)

Related topics

Basics

Basics on Generating/Updating a Frame Model from Classic AUTOSAR Data (TargetLink Classic AUTOSAR Modeling Guide)
Comprehensive Modeling of RTE Code Pattern (TargetLink Classic AUTOSAR Modeling Guide)
Modeling Communication According to Classic AUTOSAR (TargetLink Classic AUTOSAR Modeling Guide)

References

tl_pre_add_comspecblock_hook (TargetLink File Reference)

ImplementationDataType Creation Wizard Command

Purpose

To create implementation data types (IDTs) via DD Typedef objects, reference them at data prototypes, and map the implementation data types to application data types (ADTs).

Access

You can access this command via:

Menu bar	None
Context menu of	DD SoftwareComponent objects - AUTOSAR Tools
Shortcut key	None
Toolbar button	None

Result

TargetLink performs the following:

- Creates the DD Typedef object for the implementation data type.
- References the typedef at the data prototypes that reference the selected application data types.
- Creates data type maps between the types that are referenced at the data prototypes.

Description

The wizard lets you select application data types that are referenced at a data prototype and not already mapped to a typedef.

The wizard lets you multiselect application data types by their kind (Primitive, Array, Record, Composite) and a filter expression. This is helpful when you create one typedef for several application data types.

The wizard allows you to specify typedef properties and create the DD Typedef object. It also references the typedef at the respective data prototypes and specifies the data type maps.

ImplementationDataType Creation Wizard

Software component Displays the selected DD SoftwareComponent object.

Typedef group Displays the DD TypeDefGroup object in which the DD Typedef objects will be stored.

TargetLink proposes a default group named `IDTGroup_<SoftwareComponent>`. Edit it as required.

Data type mapping set Displays the DD DataTypeMappingSet object referenced at the DD SoftwareComponent object.

If no data type mapping set object is referenced, TargetLink creates a default data type mapping set object named `DTMS_<SoftwareComponent>.`

Data type mapping set group Displays the DD DataTypeMappingSetGroup object that contains the DD DataTypeMappingSet object referenced at the DD SoftwareComponent object.

If no data type mapping set object is referenced, TargetLink creates a default data type mapping set group object named `<SoftwareComponent>_Group`, containing a default data type mapping set object named `DTMS_<SoftwareComponent>.`

Application data type

List Lets you select one or several application data types for which to create and map a DD Typedef object.

The letters enclosed in parantheses identify the kind of the application data types:

- (A) for array
- (C) for component
- (P) for primitive
- (R) for record
- (X) for composite

Kind Lets you filter the list of application data types by their kind, i.e., Array, Primitive, Record, or Composite.

Note

When specifying typedefs for composite application data types of the Record type proceed as follows:

1. Specify a typedef for the composite application data type. The application data type will be removed from the list.
2. Specify a typedef for each component of the composite application data type.

Map the composite application data type *and all of its components* in one session. Components are listed together with their parent application data type, even if the components are of the Primitive kind.

Filter Lets you enter a search string to filter the list of application data types. The wizard only checks whether the search string is contained in the names of the application data types. Wildcards are not supported.

Typedef

Name Lets you specify a DD Typedef object to be mapped to the selected application data types. TargetLink proposes a name following the schema `IDT_<ApplicationDataType>` that you can change as required.

Base type Lets you specify the base type of the DD Typedef object to be created and mapped to the selected application data types.

Note

Considerations for specifying base types:

- For arrays, choose the same base type as for the components of the array.
- For primitives, choose a suitable base type from the data types of the Primitive kind.
- For records, choose the Struct base type.

Scaling Displays the proposed scaling of the DD Typedef object to be created and mapped to the selected application data types.

Max Displays the proposed max value of the DD Typedef object to be created and mapped to the selected application data types.

Min Displays the proposed min value of the DD Typedef object to be created and mapped to the selected application data types.

Width Displays the proposed width of the DD Typedef object to be created and mapped to the selected application data types .

Can be edited for composite application data types.

Generate map struct If selected, TargetLink creates a DD Typedef object for each composite application data type. The Typedef object's Type property is set to `Struct`.

Generate # of points component If selected, TargetLink creates a component for each axis to store the number of axis points at the DD Typedef object created for the map struct of the selected composite application data type. Only applicable if Generate map struct is selected.

Command buttons

Exclude Lets you exclude the selected data types from the list without creating a DD Typedef object.

Cancel Lets you cancel all changes and close the wizard.

Help Lets you open this document.

Apply TargetLink performs the following:

- Creates the DD Typedef object for the implementation data type.
- References the typedef at the data prototypes that reference the selected application data types.
- Creates data type maps between the types that are referenced at the data prototypes.

Related topics

HowTos

How to Create Implementation Data Types for Existing Application Data Types (Classic AUTOSAR) (TargetLink Classic AUTOSAR Modeling Guide)

Synchronize Activation Reason Settings

Purpose

To synchronize the activation reason settings (Typedef objects and Variable objects).

Access

You can access this command via:

Menu bar	None
Context menu of	DD Runnable objects

Shortcut key	None
Toolbar button	None

MATLAB API function

Refer to the corresponding MATLAB API function:

```
[bSuccess,processedObjectInfo] =
dsdd('CreateActivationReasonCodeElements',<objectIdentifier>);
```

Description

This command creates all the DD Typedef objects and DD Variable objects required for the DD Runnable object's activation reasons.

Result

The activation reason settings (Typedef objects and Variable objects) are automatically synchronized.

Related topics

Basics

Basics on Activation Reasons (📖 TargetLink Classic AUTOSAR Modeling Guide)

Synchronize Application Data Type Settings

Purpose

To synchronize ApplicationDatatype objects (DataPrototype objects, ApplicationDatatype objects and Typedef objects) during an import of AUTOSAR data into the Data Dictionary.

Access

You can access this command via:

Menu bar	None
Context menu of	`/Pool/Autosar` subtree and all SoftwareComponent objects
Shortcut key	None
Toolbar button	None

MATLAB API function

Refer to the corresponding MATLAB API function *SyncAutosar* in the 📖 TargetLink Data Dictionary Reference.

Description

By default, TargetLink executes the Synchronize Application Data Type Settings command during the import of AUTOSAR data to the DD. You can

change this behavior by setting the SynchronizeApplicationDataTypeSettings property of the `/Pool/Autosar/Config/ImportExport` object to `off`.

Result

The ApplicationDatatype settings (DataPrototype objects, ApplicationDatatype objects and Typedef objects) are synchronized.

Synchronize Interface Settings

Purpose

To synchronize the Interface Settings (Element objects, Variable objects and ComSpec objects).

Access

You can access this command via:

Menu bar	None
Context menu of	NvReceiverPort, NvSenderReceiverPort, ProvideCalPrmPort, or RequireCalPrmPort objects
Shortcut key	None
Toolbar button	None

Related DD MATLAB API function

```
[bSuccess,processedObjectInfo] = dsdd('CreateVariableObjects',<objectIdentifier>[,<bDeleteObsoleteInterfaceElements>,<bVerbose>]);
```

Description

By default, TargetLink executes the command during the import of AUTOSAR data to the DD. You can change this behavior by setting the SynchronizeCalPrmInterfaceSettings/SynchronizeNvDataInterfaceSettings property of the `/Pool/Autosar/Config/ImportExport` object to `off`.

Result

The Interface Settings (Element objects, Variable objects and ComSpec objects) are synchronized.

Synchronize Mode Settings

Purpose

To synchronize the Mode Settings (ModeDeclaration Group, Variable objects and Typedef objects) during an import of AUTOSAR data into the Data Dictionary.

Access

You can access this command via:

Menu bar	None
Context menu of	DD `ModeDeclarationGroup` objects
Shortcut key	None
Toolbar button	None

MATLAB API function

Refer to the corresponding MATLAB API function *CreateRteModes* in the 🕮 TargetLink Data Dictionary Reference.

Description

By default, TargetLink executes the **Synchronize Mode Settings** command during the import of AUTOSAR data to the DD. You can change this behavior by setting the **SynchronizeModeSettings** property of the `/Pool/Autosar/Config/ImportExport` object to `off`.

Result

The Mode Settings (ModeDeclaration Group, Variable objects and Typedef objects) are automatically synchronized.

Update AUTOSAR Model

Purpose

To update an existing AUTOSAR frame model.

Precondition

An existing frame model must be open. If no model is open, TargetLink creates a new frame model instead.

Access

You can access this command via:

Menu bar	None
Context menu of	DD **SoftwareComponent** objects - **AUTOSAR Tools** DD **SoftwareComponentGroup** objects - **AUTOSAR Tools**

Shortcut key	None
Toolbar button	None

Result

If started from the **AUTOSAR Tools** menu on the context menu of a

- DD **SoftwareComponent** object, TargetLink updates an existing frame model for the selected software component(s)
- DD **SoftwareComponentGroup** object, TargetLink updates an existing frame model for all software components contained in the selected group(s)

An Update report is displayed in the MATLAB Web browser.

Description

TargetLink updates an existing frame model according to the following rules:

Supported AUTOSAR Element in DD	Block Exists in Model	Action
yes	no[1), 2)]	TargetLink adds a corresponding block to the model and connects it appropriately. Model blocks referencing renamed AUTOSAR elements/DD objects are treated as to be deleted.
no	yes	TargetLink writes a warning to the report. You have to delete the block.
yes	yes	none
no	no[1), 2)]	none

1) Referenced models are ignored. Blocks contained in referenced models are treated as not existing in the model.
2) Renamed AUTOSAR elements/DD objects are treated as not existing in the model.

For details on the mapping between supported AUTOSAR elements and DD objects, refer to Generate Frame Model on page 72.

Update report

TargetLink generates an update report and displays it in the MATLAB Web browser.

The report contains:

- Hyperlinks to DD objects and TargetLink blocks.
- Information on changes made by TargetLink.
- Information on actions required by the user.

Color codes in the update report The update report can contain the following colors:

Color Code	Description:
Green	Blocks have been added to the frame model.
Yellow	Blocks have been changed in the frame model.
Red	User action is required as described in the report.
White	No changes have been made.

Related topics

Basics

Basics on Generating/Updating a Frame Model from Classic AUTOSAR Data
(📖 TargetLink Classic AUTOSAR Modeling Guide)

References

Generate Frame Model..... 72

Code-Generation-Related Commands

Where to go from here

Information in this section

Copy Code Generator Options

Purpose

To either export Code Generator options to the model or to import them from the model into a DD object. This command overwrites the existing options.

Access

You can access this command via:

Menu bar	None
Context menu of	DD objects
Shortcut key	None
Toolbar button	None

Result

You have imported/exported Code Generator options.

Description

For detailed description of Code Generator options, refer to Edit Code Generator Options on page 82.

Related topics

References

Edit Code Generator Options

Purpose

To access/specify all Code Generator options.

Access

You can access this command via:

Menu bar	None
Context menu of	**CodegenOptionSet** object
Shortcut key	None
Toolbar button	None
Mouse	Double-click on **CodegenOptionSet** object

Result

All Code Generator options are specified.

Description

The **Code Generator Options** dialog lists all the options available for the Code Generator at one central point. The option settings are saved to a **CodegenOptionSet** object. In the TargetLink Data Dictionary several **CodegenOptionSet** objects can be defined which can be used to generate production code for specific DD objects. For details, refer to Basics on Configuring the Code Generator for Production Code Generation (🕮 TargetLink Customization and Optimization Guide).

If you use the command with newly created **CodegenOptionSet** object that does not contain any Code Generator options and you leave the **Code Generator Options** dialog via **OK**, a complete set of options is created with their default values.

For a complete list of **Code Generator options**, refer to Code Generator Options (🕮 TargetLink Model Element Reference).

Exchanging Code Generator options between TargetLink and the DD

Code Generator options you configured for TargetLink subsystems, subsystems for incremental code generation, or referenced models via the **TargetLink Main Dialog** do not affect **OptionSet** objects stored in the Data Dictionary and vice versa. TargetLink provides API functions to exchange Code Generator options between TargetLink and the Data Dictionary, and to compare an **OptionSet** object stored in the Data Dictionary with Code Generator options you configured via the **TargetLink Main Dialog**. The following API functions are available:

- `dsdd_export_optionset` configures the Code Generator options of a TargetLink subsystem, a subsystem for incremental code generation, or a referenced model according to an **OptionSet** object stored in the Data Dictionary.

- `dsdd_import_optionset` reads the Code Generator options of a TargetLink subsystem, a subsystem for incremental code generation, or a referenced model and stores them as an **OptionSet** object in the Data Dictionary.
- `dsdd_compare_optionset` compares the Code Generator options of a TargetLink subsystem, a subsystem for incremental code generation, or a referenced model with an **OptionSet** object stored in the Data Dictionary, and lists the differences found.

Related topics

Basics

Basics on Configuring the Code Generator for Production Code Generation (📖 TargetLink Customization and Optimization Guide)

References

dsdd_compare_optionset (📖 TargetLink API Reference)
dsdd_export_optionset (📖 TargetLink API Reference)
dsdd_import_optionset (📖 TargetLink API Reference)
Export Options to Model <Model> 83
Import Options from Model <Model> 85

Export Options to Model <Model>

Purpose

To export the Code Generator options stored in the DD **CodegenOptionSet** object to the <Model>.

Access

You can access this command via:

Menu bar	None
Context menu of	**CodegenOptionSet** objects - **Copy Code Generator Options**
Shortcut key	None
Toolbar button	None

Related MATLAB API function

Refer to the corresponding MATLAB API function: `dsdd_export_optionset`

Related topics

Basics

Basics on Configuring the Code Generator for Production Code Generation
(📖 TargetLink Customization and Optimization Guide)

References

dsdd_export_optionset (📖 TargetLink API Reference)
Edit Code Generator Options 82
Import Options from Model <Model> 85

Generate Code

Purpose

To generate production code for specific DD objects, such as CodeGenerationUnit or Module objects.

Access

You can access this command via:

Menu bar	None
Context menu of	▪ CodeGenerationUnit objects ▪ CodeGenerationUnitGroup objects ▪ Module objects ▪ ModuleGroup objects
Shortcut key	None
Toolbar button	None

Result

Production code is generated for the selected DD object.

Description

You can generate production code for variable and type definitions, as well as switch and access functions straight from the TargetLink Data Dictionary. You can start the code generation either from DD Module objects, DD CodeGenerationUnit objects that structure Module objects or the corresponding group objects.

Note

This command is not available in the stand-alone mode of the TargetLink Data Dictionary Manager.

Related topics

Basics

Basics on Generating Code from the Data Dictionary (📖 TargetLink Customization and Optimization Guide)

Import Options from Model <Model>

Purpose

To import the Code Generator options from <Model> into the CodegenOptionSet object.

Access

You can access this command via:

Menu bar	None
Context menu of	CodegenOptionSet objects - Copy Code Generator Options
Shortcut key	None
Toolbar button	None

Related MATLAB API function

Refer to the corresponding MATLAB API function: `dsdd_import_optionset`

Related topics

Basics

Basics on Configuring the Code Generator for Production Code Generation (📖 TargetLink Customization and Optimization Guide)

References

dsdd_import_optionset (📖 TargetLink API Reference)
Edit Code Generator Options 82
Export Options to Model <Model> 83

Comparison-Related Commands

Where to go from here

Information in this section

Comfort Copy Middle

Purpose

To merge the selected DD object with its properties and all descendants into the merged DD workspace. Conflicting elements are not merged.

Access

You can access this command via:

Menu bar	None
Context menu of	DD three-way merge
Shortcut key	None
Toolbar button	None

Result

The selected DD object with its properties and all descendants are merged into the merged DD workspace.

Description

The **Comfort Copy Middle** command is useful if a DD three-way merge is performed when automerge is set to off because it lets you merge the selected DD object with its properties and all descendants into the merged DD workspace at the same time.

Related topics

Basics

Basics on Comparing and Merging DD Files via DD Three-Way Merge
(TargetLink Data Dictionary Basic Concepts Guide)

Compare Tool Options

Purpose

The **TargetLink Data Dictionary Compare Tool** lets you compare selected DD objects within a DD workspace, DD objects in different DD workspaces, or different DD workspaces in their entirety. You can specify, load and save further options for the comparison in this dialog.

Access

You can access this command via:

Menu bar	Tools
Context menu of	None
Shortcut key	None
Toolbar button	None

Result

You have specified the settings of the report file to be generated.

For information on how to start a comparison, refer to Select Left Side to Compare on page 97 or Select Left Side to Compare on page 97.

Description

Ignore order of objects To ignore the order of objects (Default setting).

Ignore the Access attribute To ignore differences in the Access attribute.

Ignore the Temporary attribute To ignore the Temporary attribute. The Temporary attribute defines if a sub tree is saved or discarded.

Ignore differences near machine precision To ignore floating-point inaccuracies (Default setting).

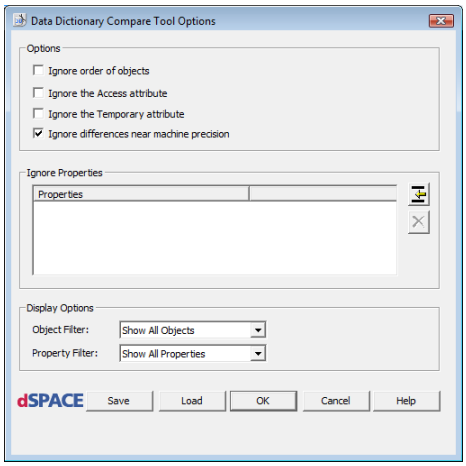
Ignore Properties To enter properties the comparison should ignore.

Object Filter Lets you select whether all objects are shown or objects with differences only.

Property Filter Lets you select whether all properties are shown or properties with differences only.

Save Lets you save an XML file with the settings you have selected in the Compare Tool Options dialog.

Load Lets you load an XML file with compare tool options settings.



Note

All settings are automatically saved and remain for the next start of the TargetLink Data Dictionary Manager.

Related topics

HowTos

How to Compare DD Objects Using the DD Two-Way Comparison (TargetLink Data Dictionary Basic Concepts Guide)

References

Compare with <DD Object> 88
Select Left Side to Compare 97

Compare with <DD Object>

Purpose

You can compare DD objects in the following ways:

- Compare selected DD objects in one DD workspace.
- Compare selected DD objects in different DD workspaces.

You can also compare entire DD workspaces. The **Compare with <DD object>** command starts the DD two-way comparison. Refer to Basics on Comparing and Merging DD Objects and Workspaces Using the DD Two-Way Comparison (📖 TargetLink Data Dictionary Basic Concepts Guide).

Precondition

You selected a DD object with the **Select Left Side to Compare** command before. Otherwise, the **Compare with <DD object>** command is not displayed in the context menu of the DD object. Refer to Select Left Side to Compare on page 97.

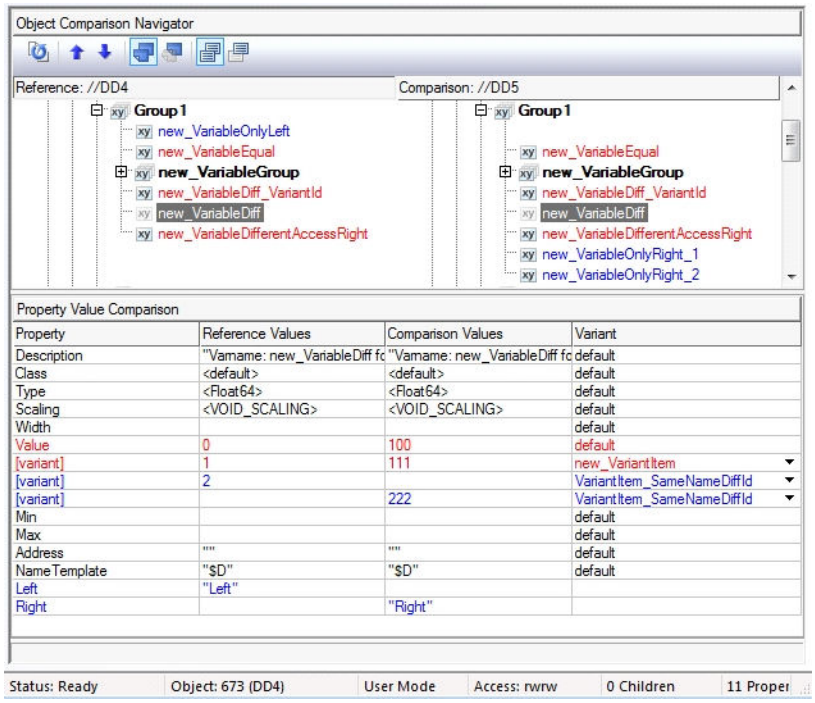
Access

You can access this command via:

Menu bar	None
Context menu of	DD objects
Shortcut key	None
Toolbar button	None

Result

The DD two-way comparison pane opens.



Refer to How to Compare DD Objects Using the DD Two-Way Comparison (📖 TargetLink Data Dictionary Basic Concepts Guide).

Navigation

Button	Name	Description
	Recompare Root	To instantly compare the DD object trees again.
	Previous / Next Difference	To jump to the next/previous difference in the DD object tree. If you synchronized the **Reference: <root path>** and the **Comparison: <root path>** in the DD two-way comparison pane, the next/previous difference is highlighted in both hierarchies. Otherwise, only the **Reference: <root path>** is highlighted.
	Show All Objects	To show all objects (default setting).
	Show Objects with Differences	To show only objects containing differences.
	Show All Properties	To show all properties in the **Property Value List** (default setting).
	Show Properties with Differences	To show only properties containing differences in the **Property Value List**.

Related topics

Basics

Basics on Comparing and Merging DD Files via DD Three-Way Merge (📖 TargetLink Data Dictionary Basic Concepts Guide)
Basics on Comparing and Merging DD Objects and Workspaces Using the DD Two-Way Comparison (📖 TargetLink Data Dictionary Basic Concepts Guide)

References

Select Left Side to Compare 97

Delete

Purpose

To delete the selected DD object. You can only delete single DD objects without corresponding mirror objects in the DD two-way comparison pane or DD three-way merge.

Access

You can access this command via:

Menu bar	Edit
Context menu of	DD two-way comparison DD three-way merge
Shortcut key	`Del`
Toolbar button	

Result

The selected DD object is deleted.

Description

If a DD object that is referenced elsewhere is deleted, the reference becomes invalid. If you delete a parent DD object, all DD child objects are also automatically deleted. Once deleted, the DD object cannot be restored.

Related topics

HowTos

How to Delete Data Dictionary Objects (📖 TargetLink Data Dictionary Basic Concepts Guide)

Generate Report (HTML/XML)

Purpose

To generate a comparison report in HTML or XML format.

Note

An HTML report can only be generated for the DD two-way comparison, and it can only be generated in MATLAB mode.

Access

You can access this command via:

Menu bar	None
Context menu of	DD two-way comparison DD three-way merge (only XML)
Shortcut key	None
Toolbar button	None

Result

A comparison report is generated in HTML or XML format.

Related topics

HowTos

How to Generate a Comparison Report Using the DD Three-Way Merge (📖 TargetLink Data Dictionary Basic Concepts Guide)
How to Generate a Comparison Report Using the DD Two-Way Comparison (📖 TargetLink Data Dictionary Basic Concepts Guide)

Merge <Left/Right/Middle>

Purpose

To merge a DD object or property value.

Access

You can access this command via:

Menu bar	None
Context menu of	DD two-way comparison (only left/right) DD three-way merge (only middle)
Shortcut key	None
Toolbar button	None

Result

The DD object or property value is merged.

Description

The merge command results in the following:

- In the DD two-way comparison pane, the selected DD object or property value is merged into the corresponding reference (left) or comparison (right) DD workspace.
- In the DD three-way merge pane, the selected DD object or property value is merged into the merged (middle) DD workspace.

Related topics

Basics

Basics on Comparing and Merging DD Files via DD Three-Way Merge (📖 TargetLink Data Dictionary Basic Concepts Guide)
Basics on Comparing and Merging DD Objects and Workspaces Using the DD Two-Way Comparison (📖 TargetLink Data Dictionary Basic Concepts Guide)

Merge <Left/Right/Middle> Without Overwrite

Purpose

To merge DD objects without overwriting DD child objects or property values.

Access

You can access this command via:

Menu bar	None
Context menu of	DD two-way comparison (only left/right) DD three-way merge (only middle)
Shortcut key	None
Toolbar button	None

Result

The DD objects are merged without overwriting DD child objects or property values.

Related topics

Basics

Basics on Comparing and Merging DD Files via DD Three-Way Merge (TargetLink Data Dictionary Basic Concepts Guide)
Basics on Comparing and Merging DD Objects and Workspaces Using the DD Two-Way Comparison (TargetLink Data Dictionary Basic Concepts Guide)
Merge <Left/Right/Middle> 92

Merge <Left/Right/Middle>, and Overwrite Objects

Purpose

To merge DD objects. DD child objects and their property values are overwritten.

Access

You can access this command via:

Menu bar	None
Context menu of	DD two-way comparison (only left/right) DD three-way merge (only middle)
Shortcut key	None
Toolbar button	None

Result

The DD objects are merged. DD child objects are overwritten.

Related topics

Basics

Basics on Comparing and Merging DD Files via DD Three-Way Merge (📖 TargetLink Data Dictionary Basic Concepts Guide)
Basics on Comparing and Merging DD Objects and Workspaces Using the DD Two-Way Comparison (📖 TargetLink Data Dictionary Basic Concepts Guide)
Merge <Left/Right/Middle>..... 92

Merge <Left/Right/Middle>, and Overwrite Properties

Purpose

To merge DD objects without overwriting DD child objects. If properties differ, the target properties are overwritten.

Access

You can access this command via:

Menu bar	None
Context menu of	DD two-way comparison (only left/right) DD three-way merge (only middle)
Shortcut key	None
Toolbar button	None

Result

The DD objects are merged. Property values are overwritten.

Related topics

Basics

Basics on Comparing and Merging DD Files via DD Three-Way Merge (📖 TargetLink Data Dictionary Basic Concepts Guide)
Basics on Comparing and Merging DD Objects and Workspaces Using the DD Two-Way Comparison (📖 TargetLink Data Dictionary Basic Concepts Guide)
Merge <Left/Right/Middle>..... 92

Merge and Replace

Purpose

To copy property values or DD child objects from one DD object to another. The DD objects can reside in different DD workspaces.

Access

You can access this command via:

Menu bar	None
Context menu of	DD objects
Shortcut key	None
Toolbar button	None

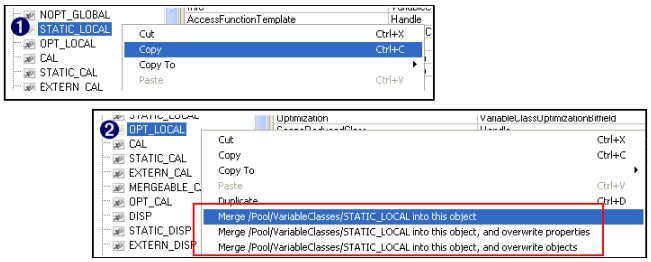
Description

Merging is possible in three ways:

- Merge without overwrite, or
- Merge and overwrite properties, or
- Merge and overwrite objects
- Replace objects

The command is available in the context menu of a DD object B only, if:

1. You already copied a DD object A to the Clipboard (Ctrl+C), and
2. Object B is of the same type as object A (for example, you can merge a **VariableGroup** with another **VariableGroup** but not with a **Variable**)



Result

The DD object is merged with another object.

Related topics

HowTos

How to Merge or Replace DD Objects (TargetLink Data Dictionary Basic Concepts Guide)

Recompare Root

Purpose

To refresh a comparison.

Access

You can access this command via:

Menu bar	None
Context menu of	DD two-way comparison DD three-way merge
Shortcut key	None
Toolbar button	

Result

The comparison is refreshed.

Replace <Left/Right/Middle>

Purpose

To replace a DD object or property value.

Access

You can access this command via:

Menu bar	None
Context menu of	DD two-way comparison (only left/right) DD three-way merge (only middle)
Shortcut key	None
Toolbar button	None

Result

The DD object or property value is replaced.

Description

The replace command results in the following:

- In the DD two-way comparison pane, the corresponding DD object or property value is replaced by the selected DD object or property value from the reference or comparison.
- In the DD three-way merge pane, the DD object or property value of the merged DD is replaced by the selected DD object or property value from reference or comparison.

Related topics

Basics

Basics on Comparing and Merging DD Files via DD Three-Way Merge (📖 TargetLink Data Dictionary Basic Concepts Guide)
Basics on Comparing and Merging DD Objects and Workspaces Using the DD Two-Way Comparison (📖 TargetLink Data Dictionary Basic Concepts Guide)

Select Left Side to Compare

Purpose

To compare and merge selected DD objects in one DD workspace, DD objects in different DD workspaces, or whole DD workspaces. You can initiate the comparison by selecting a DD object with this command.

Access

You can access this command via:

Menu bar	None
Context menu of	DD two-way comparison
Shortcut key	None
Toolbar button	None

Result

You have selected a DD object for a comparison.

Next Step

You must select another object with the **Compare with <DD Object>** command. Refer to Compare with <DD Object> on page 88 or Compare with <DD Object> on page 88.

Related topics

HowTos

How to Compare DD Objects Using the DD Two-Way Comparison (📖 TargetLink Data Dictionary Basic Concepts Guide)

References

Compare with <DD Object> 88

Customization-Related Commands

Where to go from here

Information in this section

Edit Menu Extension Specification

Purpose

To edit the file that contains the specifications of the menu extensions in the Data Dictionary Manager. Menu extension specifications are plug-in mechanisms to call user-specific M-files from menus and context menus of the Data Dictionary Manager.

Access

You can access this command via:

Menu bar	Extras
Context menu of	None
Shortcut key	None
Toolbar button	None

Result

The menu extension specification is edited.

For the modified specification to take effect, you have to reload it. For further information, refer to Reload Menu Extension Specification on page 102.

Description

The `DDManagerMenuExtension.xml` file in `%USERPROFILE %\AppData\Local\dSPACE\<Installation GUID>\TargetLink\DDMenuExtension`, where <GUID> stands for a globally unique identifier provided by the TargetLink installation, opens in the current Code Editor (by default MATLAB's built-in editor). The Code Editor can be changed via the `tl_pref` tool.

You can specify menu commands in the menu bar, or on the context menus of objects in the **DD Navigator** or properties in the property value list.

The command does not apply to menu extension specifications in separate XML files. To specify user-specific menu commands, you can create and/or open the

additional XML files via the editor's commands. You must place the XML files in `%USERPROFILE%\AppData\Local\dSPACE\<Installation GUID>\TargetLink\DDMenuExtension`.

The command is not available in stand-alone mode.

Syntax

Menu bar The following syntax is used to specify the menu commands in the menu bar:

```
<ToolsMenu ParentMenuName=<ParentMenuName>
    MenuName=<MenuName>
    Description=<Description>
    ShowDDMessages=<showDDMessages>
    Separator=<separator>>
   <Script ScriptName=<functionName>/>
   <Param Value=<value>/>
</ToolsMenu>
```

Object context menu The following syntax is used to specify the menu commands in an object's context menu in the DD Navigator:

```
<ObjectMenu ParentMenuName=<ParentMenuName>
    MenuName=<MenuName>
    Description=<Description>
    ShowDDMessages=<showDDMessages>
    Separator=<separator>>
    <ObjectKind Value=<objectKind>/>
    <Script ScriptName=<functionName>/>
</ObjectMenu>
```

Property context menu The following syntax is used to specify the menu commands in a property's context menu in the property value list:

```
<PropertyMenu ParentMenuName=<ParentMenuName>
    MenuName=<MenuName>
    Description=<Description>
    ShowDDMessages=<showDDMessages>
    Separator=<separator>>
    <ObjectKind Value=<objectKind>/>
    <PropertyName Value=<propertyName>/>
    <Script ScriptName=<functionName>/>
</PropertyMenu>
```

Syntax explanation

The following syntax is used to specify the menu commands:

Syntax	Description
`<ParentMenuName>`	Specifies where the new menu appears in the menu bar. Use | as a separator to insert submenus. Without specifying the `<ParentMenuName>` the new menu appears in the Tools menu.
`<MenuName>`	Label that appears in the menu (menu bar, object's or property's context menu).

Syntax	Description
<Description>	Descriptive text that appears in the Message Browser or a custom output view when the MATLAB function is called up. This attribute is optional.
<showDDMessages>	Specifies whether messages that are generated during command execution should appear in the Message Browser or a custom output view. If set to **off**, messages are not shown. This attribute is optional. Its default value is **on**.
<separator>	Specifies whether a separating line should be inserted before the menu command. If there is only one menu command, this argument is ignored. This attribute is optional. Its default value is **off**.
<objectKind>	Specifies the object kind to which the menu command applies. You can specify one or more object kinds for which the command shall apply. You may use this only in the specifications of an object's or a property's context menu.
<propertyName>	Specifies the name of the property to which the menu command applies. You can specify one or more property names for which the command shall apply. You may use this only in the specifications of a property's context menu.
<functionName>	Name of the MATLAB function to be called. The name must comply with rules for MATLAB identifiers.
<value>	Value of an additional constant string parameter. The number of these parameters is arbitrary (max. 64). You may use this attribute only in the specifications of the menu bar.
<Param Value="$MsgId"/>	The token $MsgID in a parameter value is replaced by the current message ID in the DD Manager Message Browser. Use it to add child messages to the parent message and to show them in the Message Browser of the DD Manager.
<Param Value="$MenuName"/>	The token $MenuName is replaced by the menu name. Use it to distinguish the function call source.

Related topics

Basics

Basics on Adding Custom Functionality to the Data Dictionary Manager (📖 TargetLink Data Dictionary Basic Concepts Guide)

HowTos

How to Add User-Specific Menu Extensions to Objects' Context Menus in the Data Dictionary Navigator (📖 TargetLink Data Dictionary Basic Concepts Guide)
How to Add User-Specific Menu Extensions to Properties' Context Menus in the Property Value List (📖 TargetLink Data Dictionary Basic Concepts Guide)
How to Add User-Specific Menu Extensions to the Menu Bar (📖 TargetLink Data Dictionary Basic Concepts Guide)

References

Reload Menu Extension Specification..... 102
tl_pref (📖 TargetLink API Reference)

Filter List (Filter Rule Sets)

Purpose

To select filter rule sets in order to hide specific DD objects and properties.

Access

You can access this command via:

Menu bar	None
Context menu of	None
Shortcut key	None
Toolbar button	

Result

A filter rule set hides specific DD objects and properties according to the contained filter rules.

Description

You can create your own filter rule sets or select one of the predefined filter rule sets:

- Admin - a filter rule set for administrators
- AR_User - A filter rule set for AUTOSAR users
- NonAR_NonRTOS_User - A filter rule set for both non-AUTOSAR and non-RTOS users

Refer to Basics on Filter Rule Sets for the Data Model (📖 TargetLink Data Dictionary Basic Concepts Guide) and How to Create Filter Rule Sets via DD Files (📖 TargetLink Data Dictionary Basic Concepts Guide).

Related topics

Basics

Basics on Filter Rule Sets for the Data Model (📖 TargetLink Data Dictionary Basic Concepts Guide)

HowTos

How to Create Filter Rule Sets via DD Files (📖 TargetLink Data Dictionary Basic Concepts Guide)

Examples

Examples of Filter Rule Sets for Different Use Cases (📖 TargetLink Data Dictionary Basic Concepts Guide)

Reload Menu Extension Specification

Purpose

To reload the files that contain the specifications of the menu extensions into the TargetLink Data Dictionary Manager. Menu extension specifications are plug-in mechanisms to call user-specific M-files from menus and context menus of the TargetLink Data Dictionary Manager.

Access

You can access this command via:

Menu bar	Extras
Context menu of	None
Shortcut key	None
Toolbar button	None

Result

By default, the `DDManagerMenuExtension.xml` file that contains the specifications of the menu extensions is reloaded. User-specific menu commands specified in separate XML files are reloaded, too. You must place the XML files in `%USERPROFILE%\AppData\Local\dSPACE\<Installation GUID>\TargetLink\DDMenuExtension`, where <GUID> stands for a globally unique identifier provided by the TargetLink installation. You can view the <GUID> with the `tl_env('GetApplicationSettingsPath')` MATLAB command.

Description

After the specifications of the menu extensions were edited, you have to reload the XML files for the modifications to the files to take effect.

The command is not available in stand-alone mode.

Related topics

Basics

Basics on Adding Custom Functionality to the Data Dictionary Manager
(📖 TargetLink Data Dictionary Basic Concepts Guide)

References

Edit Menu Extension Specification..... 98

Data-Model-Related Commands

Where to go from here

Information in this section

Upgrade Current DD

Purpose

To upgrade the DD workspace to the current version of the DD data model.

Access

You can access this command via:

Menu bar	Tools
Context menu of	None
Shortcut key	None
Toolbar button	None

Result

The DD workspace is upgraded.

Description

If the active DD workspace does not match the current version of the Data Model, it has to be upgraded.

For a detailed description on how to upgrade previously created DD project files, refer to Upgrading Models, Libraries, and Data Dictionaries (TargetLink New Features and Migration Guide).

Related topics

Basics

Upgrading Models, Libraries, and Data Dictionaries (TargetLink New Features and Migration Guide)

Validate

Purpose

To check if the DD objects contain valid data, such as cross-dependencies between DD objects.

Access

You can access this command via:

Menu bar	None
Context menu of	DD objects
Shortcut key	None
Toolbar button	

Result

The DD objects are validated (with level 4).

Description

For example, the Data Dictionary Manager checks whether a Scaling object exists that is referenced from a Variable object.

Related topics

Basics

Basics on Validating Data Dictionaries (TargetLink Data Dictionary Basic Concepts Guide)

HowTos

How to Validate DD Objects (TargetLink Data Dictionary Basic Concepts Guide)

File-Related Commands

Where to go from here

Information in this section

Close

Purpose

To close the active DD workspace.

Access

You can access this command via:

Menu bar	File
Context menu of	None
Shortcut key	`Ctrl` + `W`
Toolbar button	

Result

The active DD workspace is closed.

Note

You cannot close the DD0 workspace tab. The Close command works only if you have more than one DD workspace open.

Close All

Purpose

To close all DD workspaces.

Access

You can access this command via:

Menu bar	File
Context menu of	None
Shortcut key	None
Toolbar button	None

Result

The active DD workspace is closed.

Note

You cannot close the DD0 workspace tab. The Close All command works only if you have more than one DD workspace open.

Load

Purpose

To load a file to the currently selected DD object.

Access

You can access this command via:

Menu bar	None
Context menu of	DD objects
Shortcut key	None
Toolbar button	None

Result

The selected DD file is loaded into the Data Dictionary. Its contents are inserted into or below the selected DD object in the DD object tree.

Description

The DD files that were previously saved with the Save As command can be loaded. However, their data has to conform with the type of data at the insertion point. There is no reference between the loaded contents and the original DD file. Changes made in the loaded contents do not influence the original DD file.

Dialog settings

Look in Lets you select the path and folder from the file to be loaded.

File name Displays the name of the selected file.

Files of type Lets you select which type of file should be opened.

Load as child object Indicates that the DD file becomes a child of the specified DD object.

Mode Lets you specify how the DD file is loaded into the DD object tree:

Setting	Description
stop on conflict (default)	DD file loading is interrupted as soon as a conflict occurs. This ensures that existing objects and properties are not overwritten.
merge	Objects in the DD object tree which conflict with loaded objects are not deleted but augmented with child objects and properties from the DD file.
overwrite	Objects and properties in the DD object tree which conflict with loaded objects are deleted.
merge-overwrite	Objects in the DD object tree which conflict with loaded objects are not deleted but augmented with child objects. Conflicting properties are overwritten.

Related topics

HowTos

How to Load DD Files (📖 TargetLink Data Dictionary Basic Concepts Guide)

References

Save As..... 114

Load included Files

Purpose

To load/reload all the included files into the selected DD workspace. You are prompted if you overwrite modified points of inclusion.

Access

You can access this command via:

Menu bar	File
Context menu of	None

Shortcut key	None
Toolbar button	None

Result

All the included files are loaded into the selected DD workspace.

Point of Inclusion

Purpose

To create, edit or remove a Point of Inclusion. The Point of Inclusion dialog allows you to include DD files in a specific DD subtree. This method is primarily useful if you have to share centrally managed configuration and pool data.

Note

You can also edit DDIncludeFile objects in the `/Config/DDIncludeFiles` object tree. These objects are required to specify Points of Inclusions (e.g., the path of the partial DD file to be included). For more information, refer to DDIncludeFile Object Dialog on page 39.

Access

You can access this command via:

Menu bar	None
Context menu of	DD objects (except `Config`, `Config/General` and `Config/DDIncludeFiles`)
Shortcut key	None
Toolbar button	None

Result

You have created, edited, or removed Points of Inclusions.

Available context commands

Make Point of Inclusion Lets you create a new Point of Inclusion. Opens the Point of Inclusion dialog. For more information, refer to the Point of Inclusion dialog settings (Make and Edit Point of Inclusion) on page 110.

Edit Point of Inclusion Lets you edit an existing Point of Inclusion. Opens the Point of Inclusion dialog. Refer to the dialog description below.

Remove Point of Inclusion Lets you remove a Point of Inclusion

Load Included Files Available only if you select a DDIncludeFileGroup in the `/Config/DDIncludeFiles` object tree. Loads all the specified included files of the DDIncludeFile objects into the DD workspace.

Load Included File Available only if you select a DDIncludeFile object in the `/Config/DDIncludeFiles` object tree. Loads the specified included file of the DDIncludeFile object into the DD workspace.

Goto Point of Inclusion Available only if you select a DDIncludeFile object in the `/Config/DDIncludeFiles` object tree. Jumps to the specified object containing the Point of Inclusion.

Goto DDIncludeFile Object Available only if you select a DD object containing a point of inclusion. Jumps to the specified DDIncludeFile object.

Point of Inclusion dialog settings (Make and Edit Point of Inclusion)

Name Lets you specify the name of the DDIncludeFile object that contains the inclusion point data.

Description Lets you enter a description for the DDIncludeFile object that contains the inclusion point data.

DDIncludeFile object Displays the path of the DDIncludeFile object that contains the inclusion point data. If not yet created, `<new>` is displayed.

Point of inclusion object kind Displays the object kind of the object you selected to make a Point of Inclusion.

Include file Lets you specify the path of the partial DD file to be included.

Include file path specification Lets you specify how the file path is stored in the DDIncludeFile. The following options are available:

- Absolute path
- File name only
- Path relative to current working directory
- Path relative to main DD directory (using the MainDDDir macro)
- Path relative to the DDINCLUDE system environment variable (if specified)

Load include file with main DD Lets you specify whether the included DD file is automatically loaded when loading the main DD file.

Save include file with main DD Lets you specify the saving behavior of the included DD file (autoload). The following options are available:

- Off
- On
- OnlyIfModified
- PromptIfmodified

Load include file on OK Lets you specify whether the partial DD file is loaded (in overwrite mode) into the Point of Inclusion after clicking OK.

Save include file on OK Lets you specify whether the partial DD file is saved after clicking OK.

Include file information (read-only) Provides information about the included DD file:

- Absolute path of the included DD file
- File root object name
- File root object kind
- Parent object
- Data Model Revision

For details, refer to TargetLink Data Dictionary Reference.

Related topics

HowTos

How to Include Partial Data Dictionary Files (TargetLink Data Dictionary Basic Concepts Guide)

References

DDIncludeFile Object Dialog..... 39

Recent Files

Purpose

To open recently loaded DD project files. The Recent Files list contains DD project files that were recently loaded or saved in the TargetLink Data Dictionary Manager and DD project files from recently opened models. You can choose to open them from these locations:

- in the active DD workspace
- in a new DD workspace

Access

You can access this command via:

Menu bar	File
Context menu of	None
Shortcut key	None
Toolbar button	None

Result

You have loaded a recently opened DD project file.

Related topics

HowTos

How to Load DD Files (TargetLink Data Dictionary Basic Concepts Guide)

Save

Purpose

To save the currently open DD workspace.

Access

You can access this command via:

Menu bar	File
Context menu of	None
Shortcut key	**Ctrl** + **S**
Toolbar button	

Result

The currently open DD workspace is saved under the file path and with the name displayed in the title bar of the Data Dictionary Manager.

Description

If the file has already been saved, it is saved under its existing name. If the name is `untitled.dd`, you are prompted to save it with a specified file name.

Note

- You are not asked to confirm whether you want to overwrite an existing DD project file.
- If the DD project file is write-protected, a message is displayed, click **Overwrite** to overwrite the file in its current location or click **Save As**.
- If you changed DD objects that reside in included DD files, a dialog opens for you to save these DD objects, as well. Select the checkbox to use the selected option for all included DD files. Click **Help** for more information on included DD files.

Related topics

HowTos

How to Save DD Workspaces to DD Project Files (TargetLink Data Dictionary Basic Concepts Guide)

References

Save As..... 114

Save All

Purpose

To save all currently open DD workspaces.

Access

You can access this command via:

Menu bar	File
Context menu of	None
Shortcut key	None
Toolbar button	

Result

All the currently open DD workspaces are saved, even DD workspaces that are only opened but not displayed in a separate pane yet.

Description

The Save as dialog opens for all DD workspaces which are untitled. Refer to Save As on page 114.

Note

- If the DD project file is write-protected, a message is displayed, click Overwrite to overwrite the file in its current location or click Save As.
- If you changed DD objects that reside in included DD files, a dialog opens for you to save these DD objects, as well. Select the checkbox to use the selected option for all included DD files. Click Help for more information on included DD files.

Related topics

HowTos

How to Save DD Workspaces to DD Project Files (TargetLink Data Dictionary Basic Concepts Guide)

References

Save As..... 114

Save As

Purpose

To save the currently selected DD workspace or a single DD object under a new file name and/or in a different folder.

Access

You can access this command via:

Menu bar	File
Context menu of	DD objects
Shortcut key	**Ctrl** + **A**
Toolbar button	None

Result

The currently selected DD workspace or a single DD object is saved under a new file name and/or in a different folder.

Description

The new file name is displayed in the title bar of the Data Dictionary Manager.

Note

If you changed DD objects that reside in included DD files, a dialog opens for you to save these DD objects, too. Select the checkbox to use the selected option for all included DD files. Use **Help** to get more information on included DD files.

Dialog settings

Save in Lets you select the path and folder in which the file should be saved.

File name Lets you specify the name of the file.

Save as type Lets you select which type the file should be saved as.

Saving single DD objects to a DD project file

You can save single DD objects and its child elements in a DD project file. In the DD Navigator, select the DD object you want so save and use the Save As command from the context menu. For a detailed description, refer to How to Save a DD Subtree to a Partial DD File (📖 TargetLink Data Dictionary Basic Concepts Guide).

Related topics

HowTos

How to Save DD Workspaces to DD Project Files (📖 TargetLink Data Dictionary Basic Concepts Guide)

General Commands

Where to go from here

Information in this section

Collapse All

Purpose

To collapse all DD child objects of a selected DD object in the DD Navigator, DD two-way comparison pane or DD three-way merge pane, or to collapse the subordinate messages in the **Message Browser**.

Access

You can access the command via:

Menu bar	None
Context menu of	▪ DD Navigator ▪ DD two-way comparison ▪ DD three-way merge ▪ DD Message Browser
Shortcut key	None
Toolbar button	

Result

The displayed subordinate DD child objects or messages are hidden.

Description

If a high number of DD child objects is displayed that you are not working with, it can be useful to reduce the number of displayed objects. Some messages in the **Message Browser** contain a hierarchy of messages. If a high number of messages is displayed, it can be useful to hide subordinate messages to reduce complexity.

Related topics

References

Expand All

Purpose

To expand all DD child objects of a selected DD object in the DD Navigator, DD two-way comparison pane, or DD three-way merge pane, or to expand all the collapsed subordinate messages in the **Message Browser**.

Access

You can access the command via:

Menu bar	None
Context menu of	▪ DD Navigator ▪ DD two-way comparison ▪ DD three-way merge ▪ DD Message Browser
Shortcut key	None
Toolbar button	

Result

The hidden subordinate DD child objects or messages are displayed.

Related topics

References

Collapse All..... 116

Help-Related Commands

Where to go from here

Information in this section

About TargetLink

Purpose

To display information on which TargetLink version is currently installed on your system.

Access

You can access this command via:

Menu bar	Help
Context menu of	None
Shortcut key	None
Toolbar button	None

Result

The version number of your currently installed TargetLink is displayed.

dSPACE Help

Purpose

To open dSPACE Help for the Data Dictionary Manager.

Access

You can access this command via:

Menu bar	Help
Context menu of	None
Shortcut key	None
Toolbar button	

Result

dSPACE Help for the Data Dictionary Manager is opened.

New Features and Migration

Purpose

To open the online help of the New Features and Migration document.

Access

You can access this command via:

Menu bar	Help
Context menu of	None
Shortcut key	None
Toolbar button	

Result

The online help of the New Features and Migration document is opened.

Show Details

Purpose

To show or hide the Details pane providing information on the selected DD object and, if selected, its associated properties. By default, the pane is activated.

Access

You can access this command via:

Menu bar	Help
Context menu of	each DD object and property
Shortcut key	None
Toolbar button	None

Result

The Details pane is displayed or hidden. Refer to Details on page 23.

Related topics

References

Show Embedded Help

Purpose

To open or close the Embedded Help pane providing detailed information on selected DD objects and properties. By default, the pane is activated.

Access

You can access this command via:

Menu bar	Help
Context menu of	None
Shortcut key	None
Toolbar button	None

Description

The Embedded Help pane contains the following tabs:

- Description

  Displays a detailed description of the selected DD object or property.
- Values

  Displays the values of the selected DD object or property.
- Examples

  Displays useful examples related to the selected DD object or property.
- Functions

  Displays API functions related to the selected DD object or property.

  For information on Data Dictionary MATLAB API functions, refer to Getting Started with the DD MATLAB® API (TargetLink Data Dictionary Basic Concepts Guide).

- **Related Properties**
  Displays related properties to the selected DD object or property.
- **Related Topics**
  Displays help topics related to the selected DD object or property.

Result

You have opened or closed the Embedded Help pane.

Related topics

Basics

Getting Started with the DD MATLAB® API (📖 TargetLink Data Dictionary Basic Concepts Guide)

References

Show Output View 183

Useful Links

Purpose

Provides useful links, such as the TargetLink Product Support Center.

Access

You can access this command via:

Menu bar	Help
Context menu of	None
Shortcut key	None
Toolbar button	None

Result

Useful links are provided.

Using dSPACE Help

Purpose

To open the online help of dSPACE Help.

Access

You can access this command via:

Menu bar	Help
Context menu of	None
Shortcut key	None
Toolbar button	

Result

You have opened dSPACE Help.

Import-Export-Related Commands

Where to go from here

Information in this section

Export

Purpose

To export the active DD workspace in another file format.

Access

You can access this command via:

Menu bar	File
Context menu of	None
Shortcut key	None
Toolbar button	None

Result

The active DD workspace is exported in another file format.

Description

All supported file formats are listed in the **Export** submenu.

The export of file formats is described in the following documents:

File format	Refer to
A2L	Basics on the Workflow for Exporting A2L Files (📖 TargetLink Interoperation and Exchange Guide)
AUTOSAR	Export as AUTOSAR File on page 128
OIL	Basics of Exporting OIL Files (📖 TargetLink Interoperation and Exchange Guide)
Container	Export as Container on page 129
XML	How to Export XML Files via the TargetLink Data Dictionary Manager (📖 TargetLink Interoperation and Exchange Guide)

Note

A2L files, OIL files, and Containers can be exported only if DD0 is the active workspace.

Related topics

HowTos

How to Export Data Files from DD Workspaces (📖 TargetLink Data Dictionary Basic Concepts Guide)

Export as A2L File

Purpose

To export an A2L (ASAM MCD-2 MC) file from the **TargetLink Data Dictionary**.

Access

You can access this dialog via:

Menu bar	File – Export – as A2L File
Context menu of	None
Shortcut key	None
Toolbar button	None

Note

A2L files can be exported only if DD0 is the active workspace.

Result

The TargetLink Data Dictionary Manager exports data to an A2L (ASAM MCD-2 MC) file.

Description

The pane is docked to the right side in the Data Dictionary Manager. You can move it to change the docking side or to use it as a dialog. You can open the same pane several times to make different settings in each pane.

Dialog settings

File Lets you specify the name of the A2L file to be generated.

Build object Lets you specify the Build object containing information on the ECU application for which you want to generate an A2L file. The Build object you specify must correspond to the selected subsystem (see below).

Subsystems Lets you select the Subsystem object(s) for which an A2L file is to be generated. In the A2L file, each Subsystem object is exported as a `MODULE` element. To select several subsystems, open the Subsystem Selection dialog via the Browse button.

Stylesheet Lets you select the style sheet used to transform the intermediate XML file into the final A2L file. You can specify the `<TL_InstRoot>\Dsdd\A2L\StyleSheets\a2l_export_control.xsl` or `<TL_InstRoot>\Dsdd\A2L\StyleSheets\a2l_export_control_include_aml.xsl` file as the default style sheet.

Tip

You can create your own XSL style sheet as an alternative to the default style sheet. You can specify the path for your own style sheets in the TargetLink Preferences. Refer to Defining Search Paths for Customization Files, Demo Models,TSM Extension Packages, and Instruction Set Simulators (📖 TargetLink Customization and Optimization Guide).

ASAP1b IF Lets you specify one or more ASAP1b interfaces to be included in the generated A2L file. For details on the ASAP1b interfaces, refer to Basics on ASAP1b Interfaces (📖 TargetLink Interoperation and Exchange Guide).

Note

The names of the ASAP1b interfaces are case-sensitive and must be comma-separated if you specify more than one interface.

Name prefix Lets you specify a prefix for the names of `COMPU_METHOD` and `RECORD_LAYOUT` elements in the A2L file. TargetLink uses this prefix as follows:

`<Prefix>.<Name>`

Providing a prefix prevents naming conflicts when you merge incrementally generated A2L files.

Default: no prefix.

Phase Lets you specify the phase of the export process that the A2L export module will run.

- If you select CreateAll, the export module will generate the A2L file in one step.
- If you select ConvertToA2L (phase 1), properties required by calibration tools and the data to be written to the A2L file are added to a temporary DD A2L object in the DD3 workspace. You can modify this temporary DD object before transforming it into the final A2L file.
- If you select WriteA2Lfile (phase 2), the temporary DD A2L object is transformed into the final A2L file.

> **Note**
>
> Do not run phase 2 before running phase 1.

For details on the various export phases, refer to Basics on Customizing the A2L Export (📖 TargetLink Interoperation and Exchange Guide).

Generate project frame Lets you specify the generation of a `PROJECT` element in the A2L file to be exported.

- If the checkbox is selected, the export module generates a `PROJECT` element as a frame for the `MODULE` elements in the A2L file:

```
/begin Project
   /begin MODULE
   /end MODULE
   ...
   /begin MODULE
   /end MODULE
/end Project
```

  The export module derives the `PROJECT` name from the name of the A2L file to be exported. All other settings are derived from the properties of the `\<Application_name>/ApplicationInfo` object.
- If the checkbox is not selected, the export module will not generate a `PROJECT` element as a frame for the `MODULE` elements in the A2L file.

Merge A2L modules Lets you specify the generation of `MODULE` elements in the A2L file to be exported.

- If the checkbox is selected, the export module generates only one `MODULE` element for all the specified DD subsystems.
- If the checkbox is not selected, the export module will generate one `MODULE` element for each specified DD Subsystem object.

Overwrite calibration properties Lets you specify how to process properties that are specific to calibration purposes (calibration tool-specific properties).

- If the checkbox is selected, the export module overwrites calibration tool-specific properties with values derived from existing data.

- If the checkbox is not selected, the export module does not overwrite the properties of **Variable** objects. Calibration tool-specific properties that you specified manually in the Data Dictionary are included in the generated A2L file without being overwritten.

During A2L file export, calibration tool-specific properties are set in phase 1. For details on the various export phases and on which calibration tool-specific properties are set, refer to Basics on Customizing the A2L Export (📖 TargetLink Interoperation and Exchange Guide).

Use look-up structs Lets you specify the export of look-up table structs.

- If the checkbox is selected, `CHARACTERISTIC` elements for curves and maps are created with standard axes and a common `RECORD_LAYOUT` element for axis and table values.
- If the checkbox is not selected, `CHARACTERISTIC` elements for curves and maps are created with shared axes (`COM_AXIS`).

External variables Lets you include descriptions of variables that fulfill the following conditions in A2L files generated by TargetLink:

- They are defined in external code.
- They are used in TargetLink's production code.

Default: cleared.

Command buttons

Export Exports an XML file according to the settings made.

Close Closes the dialog.

Help Opens dSPACE Help.

Related topics

Basics

Basics on Customizing the A2L Export (📖 TargetLink Interoperation and Exchange Guide)
Defining Search Paths for Customization Files, Demo Models,TSM Extension Packages, and Instruction Set Simulators (📖 TargetLink Customization and Optimization Guide)
Specifying the ASAP1b Interface (📖 TargetLink Interoperation and Exchange Guide)
Specifying the Build Object (📖 TargetLink Interoperation and Exchange Guide)

HowTos

How to Export A2L files via the TargetLink Data Dictionary Manager (📖 TargetLink Interoperation and Exchange Guide)

Export as AUTOSAR File

Purpose

To export AUTOSAR elements from the TargetLink Data Dictionary.

Access

You can access this command via:

Menu bar	File – Export
Context menu of	None
Shortcut key	None
Toolbar button	None

Result

The selected AUTOSAR elements are exported from the TargetLink Data Dictionary.

Note

The dialog does not show all the available options. You can specify further options at the DD `/Pool/Autosar/Config/ImportExport` object.

Description

The pane is docked to the right side in the Data Dictionary Manager. You can move it to change the docking side or to use it as a dialog. You can open the same pane several times to make different settings in each pane.

Dialog settings

Strategy Lets you select the export strategy. For more information, refer to ExportStrategy. If you select Inherit from DD, the settings in the DD ImportExport object are used.

Output Lets you select the file or folder to which to save the AUTOSAR elements.

Fallback file If an AUTOSAR element does not have any package allocation, it is exported into the fallback file.

DD object Lets you specify the DD object whose AUTOSAR elements are exported to the AUTOSAR files via the Browse button. Specify a DD object, if you want to export interface descriptions, type definitions or scalings from the `/Pool` area.

Enabled only if no subsystem is selected.

Subsystems Lets you select the subsystems whose AUTOSAR elements are exported to the AUTOSAR files via the Browse button. Select a subsystem if you want to export a complete description of an SWC's internal behavior and implementation from the `/Subsystems` area.

Enabled only if both of the following conditions are met:

- No DD object is specified.
- At least one DD **Subsystem** object exists.

Build object Lets you specify a DD **Build** object to export software components as object code.

If you want to to export an AUTOSAR file of a software component whose object code you want to deliver, you have to set the DD **SoftwareComponent** object's **DeliveryFormat** property to **`ObjectCode`** *before* starting code generation. For more information on exchanging SWC object code, refer to Exchanging Software Component Code (TargetLink Interoperation and Exchange Guide).

Format Lets you select the format of the exported AUTOSAR files. You can select from the supported AUTOSAR Releases. If **Validate against AUTOSAR schema** is selected, the respective AUTOSAR XSD schema is used for validation.

Edit Displays the property value list of the DD ImportExport object.

Command buttons

Export Exports an AUTOSAR file or files according to the settings in this dialog and in the DD ImportExport object. If there is an error, you are informed that messages are issued in the Message Browser.

Close Closes the dialog.

Help Opens dSPACE Help.

Related topics

Basics

Basics on Exporting AUTOSAR Files (TargetLink Interoperation and Exchange Guide)

References

Import/Export Hook Scripts (AUTOSAR) (TargetLink File Reference)

Export as Container

Purpose

To export containers from the Data Dictionary.

Restriction

Exporting containers is only possible from DD0.

Access

You can access this command via:

Menu bar	File – Export
Context menu of	DD SoftwareComponent objects
Shortcut key	None
Toolbar button	None

Elements of the dialog

The dialog has the following columns for each file:

- Name: The file name
- Operation: Lets you select the file operation for synchronizing containers (Insert, Delete, Exclude, Replace).
- Category: Lets you select the file category, such as `Metadata.Component.Autosar`, `Code.Component`.
- Path: The file location

The following table shows the container files statuses indicated by the colors.

File Color	Description
Black	Source and target files are the same.
Red	Source file is newer than target file.
Gray	Target file is newer than source file.
Blue	File is available only in the *source* container.
Blue (italic)	File is available only in the *target* container.

The following table shows the meanings of the symbols.

File Symbol	Description
	Target file is read-only.
	Target file does not exist.

Catalog File Lets you select a catalog file (CTLG) that is owned by TargetLink. Click the Browse button to open the **Select Catalog** (refer to Select Catalog Dialog on page 146) dialog.

Container grid

Column Chooser (available from the context menu of column headers) Lets you add a column. The command opens a dialog displaying the columns that can be added. To add a column, drag it from the dialog to the grid header. To remove a column from the grid, drag its header below the grid.

Optimum Column Width/Best Fit (available from the context menu of column headers) Lets you optimize the width of the selected column.

Optimum Arrangement of Columns/Best Fit (all columns) (available from the context menu of column headers) Lets you optimize the widths of all columns to fit the width of the editor or browser.

Related topics

Basics

Basics on Exchanging Software Components (TargetLink Interoperation and Exchange Guide)
Basics on Exporting AUTOSAR Files (TargetLink Interoperation and Exchange Guide)
Customizing Container Exchange (TargetLink Interoperation and Exchange Guide)
Exchanging Software Component Code (TargetLink Interoperation and Exchange Guide)

Export as OIL File

Purpose

To export OIL files from the Data Dictionary.

Access

You can access this command via:

Menu bar	File – Export
Context menu of	None
Shortcut key	None
Toolbar button	None

Note

OIL files can be exported only if DD0 is the active DD workspace.

Result

The selected OIL file is exported from the Data Dictionary.

Description

The pane is docked to the right side in the Data Dictionary Manager. You can move it to change the docking side or to use it as a dialog. You can open the same pane several times to make different settings in each pane.

Dialog settings

File

A dialog opens for you to do the following:

- Select the folder to export the file to.
- Specify the name of the file.

Destination folder Displays the folder to save the file to.

Subsystems Lets you select the subsystems whose DD RTOS objects are to be exported to specify the OIL file.

If no subsystem is selected, only objects from the DD Pool area will be exported.

This edit field is inactive if DD0 does not contain DD Subsystem objects.

OIL version Select the OIL version that the exported file must comply with.

Create OS object if it does not exist Specifies to create an `OS` element in the exported OIL file. If export from the DD Pool area is enabled, the name and attributes of the newly created OS element are set according to the properties of the `/Pool/RTOS/RTOSInfo` object. If export from the Pool area is disabled, the name and the attributes of the newly created `OS` element are set according to the properties of the `/Subsystems/<subsystem>/RTOS/RTOSInfo` object, where `<subsystem>` denotes the first subsystem selected via the Subsystems edit field. If an OIL file was imported beforehand and if it contains the `OS` element, this option has no effect. Existing DDOS objects will be exported.

Export from Pool area Specifies to perform the export from the DD Pool area. If you clear this checkbox, only the DD RTOS objects in the selected subsystems will be exported. If no subsystems are selected, nothing will be exported.

Use imported file structure This checkbox can be activated only if an OIL file was imported beforehand.

Specifies to use the imported file structure.

If it is used, the imported OIL file will be exported in the same way it was imported. The file structure does not change. Changed attributes are given their new values. Deleted attributes or elements are not contained in the exported OIL file. New elements are saved in the first suitable OIL file.

To avoid overwriting the originally imported files, you can redirect the export to another folder (see Save in).

If this checkbox is disabled, one OIL file named as specified in the File name edit field is exported to the destination folder.

Command buttons

Export Exports an OIL file according to the settings.

Close Closes the dialog.

Help Opens dSPACE Help.

Export as XML File

Purpose

To export XML files from the TargetLink Data Dictionary.

Access

You can access this command via:

Menu bar	File – Export
Context menu of	None
Shortcut key	None
Toolbar button	None

Result

The selected XML file is exported from the TargetLink Data Dictionary.

Description

The pane is docked to the right side in the Data Dictionary Manager. You can move it to change the docking side or to use it as a dialog. You can open the same pane several times to make different settings in each pane.

Dialog settings

File A dialog opens for you to do the following:

- Select the folder to export the file to.
- Specify the name of the file to be exported.

Object to export Select the DD object to be exported.

Include DD object in XML export Specifies whether to include the selected DD object as a root object in the exported XML file or to export only its children.

Note

XML allows only one root element. Due to this restriction, if the Include DD object in XML export is cleared, only the first child can be exported.
To avoid this complication, select the Include DD object in XML export checkbox on the Export as XML File dialog.

Export XML declaration Specifies whether to export the XML declaration.

XML version Select the XML version.

Encoding Select the encoding.

Mode Select the XML format.

Value	Description
simple	The XML file is exported in the simple mode.
extended	The XML file is exported in the extended mode.

Stylesheet Lets you specify the XSL file used as a style sheet for XSL transformation.

Schema Lets you specify whether to validate the XML file. The following values specify how validation is carried out:

Value	Description
auto	If an external DTD/schema has been specified in the edit field next to the selection of the validation type, the XML file is validated against it. If an external DTD/schema has not been specified, but a DTD/schema is referenced in the XML file, the referenced one is used. Otherwise no validation is performed.
on	The behavior is identical to **auto**, but a warning is issued if no DTD/schema can be found.
off	The XML file is not validated at all.

Command buttons

Export Exports an XML file according to the settings.

Close Closes the dialog.

Help Opens dSPACE Help.

Related topics

Basics

Exporting XML Files (TargetLink Interoperation and Exchange Guide)
Importing XML Files (TargetLink Interoperation and Exchange Guide)

References

Import from XML File..... 141

Export DD Variable Objects

Purpose

To export **Variable** objects from the DD0 workspace into a MATLAB workspace or a separate MATLAB file.

Access

You can access this command via:

Menu bar	Tools
Context menu of	None

Shortcut key	None
Toolbar button	None

Result

The selected Variable objects are exported from the DD0 workspace either into a MATLAB workspace or into a MATLAB file.

Dialog settings

Source Lets you specify the DD VariableGroup to export Variable objects recursively, that is, including all the Variable objects of the existing subgroups.

Note

You can only export variables from the DD0 workspace.

Destination Lets you specify whether the DD Variable objects should be exported to the MATLAB Base Workspace, a model workspace, or a MATLAB file.

Model/Filename Lets you select a model workspace or a MATLAB file to export the DD Variable objects to. This text box is disabled if DD Variable objects are to be exported to the MATLAB Base Workspace.

Append to existing file Lets you specify to append the exported DD Variable objects to a specified file. Otherwise the specified file is overwritten after being saved as a `.bak` file. This checkbox is available only if you export Variable objects to a file.

Synchronize value properties only Lets you specify to only synchronize value properties instead of performing a full export.

Export controlled by Displays the file path of the MLIE API function that controls the export.

Note

If the Export dialog is closed by clicking OK, the entries made for Source, Destination, and Model/Filename are saved. These entries are also passed to the Import dialog as it is very likely that you will later want to import the same variables that were exported.

Related topics

HowTos

How to Export Variable Objects via the Data Dictionary Manager (📖 TargetLink Interoperation and Exchange Guide)

References

Import MATLAB Variable Objects..... 144

Import

Purpose

To import a file to the TargetLink Data Dictionary.

Access

You can access this command via:

Menu bar	File
Context menu of	None
Shortcut key	None
Toolbar button	None

Result

The file you selected is imported to the TargetLink Data Dictionary.

Description

All supported file formats are listed in the Import submenu.

The import of file formats is described in the following documents:

File format	Refer to
AUTOSAR	Basics on Importing AUTOSAR Files (TargetLink Interoperation and Exchange Guide)
OIL	Basics of Importing OIL Files (TargetLink Interoperation and Exchange Guide)
Container	Import from Container on page 138
XML	How to Import XML Files via the TargetLink Data Dictionary Manager (TargetLink Interoperation and Exchange Guide)

Note

OIL files can be imported only if DD0 is the active workspace.

Related topics

HowTos

How to Import Data Files to DD Workspaces (TargetLink Data Dictionary Basic Concepts Guide)

Import from AUTOSAR File

Purpose

To import AUTOSAR files into the TargetLink Data Dictionary.

Access

You can access this command via:

Menu bar	File – Import
Context menu of	None
Shortcut key	None
Toolbar button	None

Result

The selected AUTOSAR file is imported into the TargetLink Data Dictionary.

Note

The dialog does not show all the available options. You can specify further options at the DD `/Pool/Autosar/Config/ImportExport` object.

Description

The pane is docked to the right side in the Data Dictionary Manager. You can move it to change the docking side or to use it as a dialog. You can open the same pane several times to make different settings in each pane.

Dialog settings

File A dialog opens for you to select one or more AUTOSAR files to import from.

Edit Displays the property value list of the DD ImportExport object.

Command buttons

Import Imports an AUTOSAR file according to the settings.

Close Closes the dialog.

Help Opens dSPACE Help.

Related topics

Basics

Basics on Importing AUTOSAR Files (TargetLink Interoperation and Exchange Guide)

References

Import/Export Hook Scripts (AUTOSAR) (TargetLink File Reference)

Import from Container

Purpose

To import containers to the Data Dictionary.

Access

You can access this command via:

Menu bar	File – Import
Context menu of	None
Shortcut key	None
Toolbar button	None

Elements of the dialog

The dialog has the following columns for each file:

- Name: The file name
- Operation: Lets you select the file operation for synchronizing containers (Insert, Delete, Exclude, Replace).
- Category: Lets you select the file category, such as `Metadata.Component.Autosar`, `Code.Component`.
- Path: The file location

The following table shows the container files statuses indicated by the colors.

File Color	Description
Black	Source and target files are the same.
Red	Source file is newer than target file.
Gray	Target file is newer than source file.
Blue	File is available only in the *source* container.
Blue (italic)	File is available only in the *target* container.

The following table shows the meanings of the symbols.

File Symbol	Description
	Target file is read-only.
	Target file does not exist.

External catalog file

Lets you select a catalog file (CTLG) that is not owned by TargetLink (external catalog file). Click the Browse button to open the Select Catalog (refer to Select Catalog Dialog on page 146) dialog.

Local catalog file

Lets you select a catalog file (CTLG) that is owned by TargetLink (local catalog file). Click the Browse button to open the **Select Catalog Dialog** (📖 Container Management Manual) dialog.

Note

- Select both an external and a local catalog file to exchange data between TargetLink and SystemDesk.
- Local containers are synchronized with external containers on import, i.e., files are copied from the external container to the local container.

Import AUTOSAR Files

Lets you import AUTOSAR files from a container to the TargetLink Data Dictionary.

Container grid

Column Chooser (available from the context menu of column headers) Lets you add a column. The command opens a dialog displaying the columns that can be added. To add a column, drag it from the dialog to the grid header. To remove a column from the grid, drag its header below the grid.

Optimum Column Width/Best Fit (available from the context menu of column headers) Lets you optimize the width of the selected column.

Optimum Arrangement of Columns/Best Fit (all columns) (available from the context menu of column headers) Lets you optimize the widths of all columns to fit the width of the editor or browser.

Related topics

Basics

Basics on Exchanging Software Components (TargetLink Interoperation and Exchange Guide)
Basics on Importing AUTOSAR Files (TargetLink Interoperation and Exchange Guide)
Customizing Container Exchange (TargetLink Interoperation and Exchange Guide)
Exchanging Software Component Code (TargetLink Interoperation and Exchange Guide)

Import from OIL File

Purpose

To import OIL files into the TargetLink Data Dictionary.

Access

You can access this command via:

Menu bar	File – Import
Context menu of	None

Shortcut key	None
Toolbar button	None

Note

OIL files can be imported only if DD0 is the active workspace.

Result

The selected OIL file is imported into the TargetLink Data Dictionary.

Description

The pane is docked to the right side in the Data Dictionary Manager. You can move it to change the docking side or to use it as a dialog. You can open the same pane several times to make different settings in each pane.

Dialog settings

File Select the file to import from.

Import strategy Lets you specify the import strategy.

- Overwrite: All DD RTOS objects (e.g., tasks, events, etc.) that are not in the imported OIL file are deleted. If an element in the OIL file already exists as a DD RTOS object, all properties that are unspecified in the OIL element remain unchanged in the Data Dictionary.
- Merge: All DD RTOS objects in the Data Dictionary that do not exist in the imported OIL file remain unchanged. If an element that must be imported already exists in the Data Dictionary, all properties that are not specified in the OIL element remain unchanged in the DD RTOS object.

Command buttons

Import Imports an OIL file according to the settings.

Close Closes the dialog.

Help Opens dSPACE Help.

Related topics

Basics

Exporting OIL Files (TargetLink Interoperation and Exchange Guide)
Importing and Exporting OIL Files (TargetLink Multirate Modeling Guide)
Importing OIL Files (TargetLink Interoperation and Exchange Guide)

References

Export as OIL File 131
Export OIL File Dialog (TargetLink Tool and Utility Reference)
Import OIL File Dialog (TargetLink Tool and Utility Reference)

Import from XML File

Purpose

To import XML files into the TargetLink Data Dictionary.

Access

You can access this command via:

Menu bar	File – Import
Context menu of	None
Shortcut key	None
Toolbar button	None

Result

The selected XML file is imported into the TargetLink Data Dictionary.

Description

The pane is docked to the right side in the Data Dictionary Manager. You can move it to change the docking side or to use it as a dialog. You can open the same pane several times to make different settings in each pane.

Dialog settings

File Select the file to import from.

Import into Lets you select the DD object to save the imported file to.

Via the Browse button, you can select DD objects from the active workspace pane. To import into DD objects in other workspaces, specify the path in the Import into edit field.

You can create a new workspace by specifying a path beginning with the new workspace's DDRoot object `//DD<n>` (where `n > 3`):
e.g., `//DD6/Pool/Variables/myVariables`.

Import as child of DD object specified above Lets you specify to import the XML file as a child of the specified DD object.

Mode You can choose between two different import modes.

Value	Description
simple	The XML file is imported in the simple mode.
extended	The XML file is imported in the extended mode.

The extended import mode allows import of XML files describing DD objects. The parent or root DD object for extended mode XML import can be specified:

- If a DD parent object is specified, the XML file is imported into a newly created DD child object of this parent object.
- If a DD root object is specified, the child elements of the XML root <ddObj> element are imported into this object, creating DD child objects. The 'Name' attribute of the XML root <ddObj> element is ignored.

XML prolog information and non-element XML objects (processing instructions, CDATA sections, comments) are not imported. If the specified DD object (root or parent) does not exist, an error message is produced. All XML files which are valid with respect to dsdd.xsd can be imported.

The extended import mode produces data errors (invalid XML data):

- For unknown elements and attributes, a warning message is produced. The message describes the invalid data.
- For objects or properties without the 'Name' attribute, an object or property with a generic name is created, and a warning message is produced which describes the DD object or property.
- For DD objects with an unknown or invalid object kind, a generic DD object is created, and a warning message is produced which describes the DD object.
- For properties whose specified data type is invalid (not defined in the Data Model), a string property is created, and a warning message is produced which describes the property.
- Properties whose data type is unknown (neither defined by the Data Model, nor by the XML Type attribute), a string property is created.
- For properties whose value is invalid (e.g., which does not evaluate to a numerical value, although the property is numeric), a string property is created, and an error message that describes the property is produced.
- Errors that come up when data is written to the DD are passed to the caller as-is.

The DD XML Import does not abort importing data after one of these errors has come up.

Stylesheet Lets you specify a filter for the contents of the XML file. You can choose between various filters.

Value	Description
off	Allows you to import the entire contents of the XML file.
XPath	Allows you to select XML nodes via the XPath expression. The XPath expression must be specified in the edit field next to the filter selection list.
XSL-file	Allows you to select XML nodes via the style sheet transformation. The name of the style sheet must be specified in the edit field next to the filter selection list.

Schema Lets you specify whether to validate the XML file. The following values specify how validation is carried out:

Value	Description
auto	If an external DTD/schema has been specified in the edit field next to the selection of the validation type, the XML file is validated against it. If an external DTD/schema has not been specified, but a DTD/schema is referenced in the XML file, the referenced one is used. Otherwise no validation is performed.
on	The behavior is identical to auto, but a warning is issued if no DTD/schema can be found.
off	The XML file is not validated at all.

Import strategy

- Overwrite (default): If data is imported to an object that already exists in the Data Dictionary, this object is overwritten.
- Merge: If an XML element is imported for which a DD object already exists in the Data Dictionary, the DD object is not removed.

  Conflicts are handled as described below:
  - The object kind of the XML element does not match the object kind of the DD object: The object kind of the DD object is not modified, and an error message is produced which describes the DD object and the XML element.
  - Property XML elements describe properties which already exist at the target DD object: The DD properties are overwritten.
  - The access rights of the XML element do not match the access rights of the DD object: The access rights of the DD object are not modified, and a warning message is produced which describes the DD object and the XML element.

Command buttons

Import Imports an XML file according to the settings.

Close Closes the dialog.

Help Opens dSPACE Help.

Related topics

Basics

Exporting XML Files (TargetLink Interoperation and Exchange Guide)
Importing XML Files (TargetLink Interoperation and Exchange Guide)

References

Export as XML File..... 133

Import MATLAB Variable Objects

Purpose

To import variables from MATLAB workspaces or MATLAB files into the DD0 workspace.

Access

You can access this command via:

Menu bar	Tools
Context menu of	None
Shortcut key	None
Toolbar button	None

Result

The selected MATLAB variables are imported into the specified VariableGroup object in the DD0 workspace.

Dialog settings

Source Lets you select whether variables should be imported into the DD from the MATLAB Base Workspace, a model workspace, or a MATLAB file.

Model/Filename Lets you select the model workspace or MATLAB file to import variables from. This text box is disabled if variables are to be imported from the MATLAB Base Workspace.

Destination Lets you select the DD0 VariableGroup to import MATLAB variables to.

Delete all objects in VariableGroup before import Lets you specify whether to delete all objects in the selected VariableGroup before import.

Synchronize value properties only Lets you specify to only synchronize value properties instead of performing a full import.

Import controlled by Displays the file path of the MLIE API function that controls the import.

Note

If the Import dialog is closed by clicking OK, the entries made for Source, Destination, and Model/Filename are saved. These entries are also passed to the Export dialog as it is very likely that you will later want to export the same variable objects that were imported.

Related topics

HowTos

How to Import Variable Objects via the Data Dictionary Manager (📖 TargetLink Interoperation and Exchange Guide)

References

Export DD Variable Objects..... 134

Manage Build

Purpose

To import information on an ECU application into a DD Build object.

Access

You can access this dialog via:

Menu bar	Tools
Context menu of	None
Shortcut key	None
Toolbar button	None

Result

Target information has been specified. This is a preparatory step for A2L file export.

Description

The information on the ECU application is imported into the specified DD Build object belonging to the specified DD Application object. If this DD Build object does not yet exist, it will automatically be created.

The command is not available in stand-alone mode.

Dialog settings

Application Lets you specify the DD Application object for which you want to export an A2L file.

Build Lets you specify the DD Build object for which you want to specify information on the build of your ECU application, such as platform-specific data and ECU address information.

Target Info file Lets you specify the target Info file.

Target config file Lets you specify the target config file.

Linker MAP file Lets you specify the linker MAP file.

Related topics

Basics

Specifying the Build Object (TargetLink Interoperation and Exchange Guide)

HowTos

How to Create and Specify the Build Object (TargetLink Interoperation and Exchange Guide)

Open Container Manager

Purpose

To start the Container Manager to manage containers for exchanging data between TargetLink and SystemDesk.

Access

You can access this command via:

Menu bar	Tools
Context menu of	None
Shortcut key	None
Toolbar button	

Result

TargetLink opens the Container Manager. Refer to Container Manager User Interface (Container Management Manual).

Select Catalog Dialog

Access

You can access this command via:

Browse buttons	Export Container dialog Import Container dialog
Context menu of	None
Shortcut key	None
Toolbar button	None

Purpose

To select a catalog.

Result

The Select Catalog dialog is displayed.

Select Catalog dialog

The dialog displays a tree view with recently used container set and catalog files. You can either select one of the displayed catalog files, or you can browse for new container set files on disk.

The dialog has the following columns for each file:

- Name: The file name
- Path: The file location

Note

Only one container set or catalog file can be selected at a time.
If you opened the Import Container or Export Container dialog for a specific software component, you can select only existing catalog files for that component, or create new catalog files, but you cannot import from or export to a container of another component.

Container set Lets you select a container set file (CTS).

- Click the Browse button to open an existing file.
- (Not available when you import a container or when you use the History command) Click the  button to create a new file.

Note

After creating a new file, use the Select Owner dialog to specify the tool that owns the file.

Related topics

References

Export as Container 129
Import from Container 138

Subsystem Selection Dialog

Purpose

Lets you select one or more available subsystems.

Access

You can access this dialog:

By clicking the Browse button in the Export submenu of A2L files, AUTOSAR files and OIL files.

Result	You have selected one or more subsystems.
Dialog settings	**Select All** To select all subsystems. **Select none** To clear all selected subsystems.

Miscellaneous Commands

Where to go from here

Information in this section

Clear

Purpose

To remove all entries from the currently selected pane.

Access

You can access this command via:

Menu bar	File
Context menu of	▪ Message Browser ▪ Find References Results pane ▪ Find Object Results pane ▪ a <Custom Output View> pane
Shortcut key	None
Toolbar button	None

Result

The Data Dictionary Manager removes all the entries from the currently selected pane.

Copy

Purpose

To copy the selected items to the Clipboard.

Access

You can access this command via:

Menu bar	Edit
Context menu of	▪ DD objects ▪ Message Browser ▪ Find References Results pane ▪ Find Object Results pane ▪ <Custom Output View> pane
Shortcut key	**Ctrl** + **C**
Toolbar button	

Result

The selected items are copied to the Clipboard.

Related topics

References

Paste..... 168

Copy All

Purpose

To copy all items from the list to the Clipboard.

Access

You can access this command via:

Menu bar	None
Context menu of	▪ Message Browser ▪ Find References Results pane ▪ Find Object Results pane ▪ <Custom Output View> pane
Shortcut key	None
Toolbar button	None

Result All items from the list are copied to the Clipboard.

Related topics

References

Copy 150

Exit

Purpose To exit the current Data Dictionary Manager session.

Access You can access this command via:

Menu bar	File
Context menu of	None
Shortcut key	**Alt + F4**
Toolbar button	None

Result The Data Dictionary Manager is closed.

Description The data remains in the memory.

Open Code Editor

Purpose To open the code editor.

Access You can access this command via:

Menu bar	None
Context menu of	None

Shortcut key	None
Toolbar button	

Result

The code editor is opened.

Note

This command is not available in the stand-alone mode of the Data Dictionary Manager.

Refresh

Purpose

To refresh the whole DD Manager GUI and re-read the data.

Access

You can access this command via:

Menu bar	View
Context menu of	None
Shortcut key	**F5**
Toolbar button	

Result

The whole DD Manager GUI is refreshed and the data is re-read.

Synchronize System Signature

Purpose

To create or synchronize a Simulink or TargetLink system for a DD Signature object or a DD Function Block object.

Access

You can access this command via:

Menu bar	None
Context menu of	`SignatureGroup` and `Signature` objects `Function Block` objects
Shortcut key	None
Toolbar button	None

Description

- If the DD Signature object is specified, it adds a TargetLink Function block to the created system and TL_[Bus]Inport and TL_[Bus]Outport blocks according to the data set at the DD Signature object.

  If there is exactly one Block object with BlockType = TL_Function that references the specified DD Signature object, the inserted TargetLink Function block is linked and its data is synchronized with this Block object.
- If a Function Block object is specified, it adds a TargetLink Function block to the system. The inserted TargetLink Function block is linked and its data is synchronized with the specified Block object.

  If this Block object references a DD Signature object, TL_[Bus]Inport and TL_[Bus]Outport blocks are also added according to the data set at the referenced DD Signature object.
- A **System Signature Synchronization Report** is generated. The report contains detailed information about differences in the specification of the DD Signature object and the interface of the Function system.

For further information, refer to

- Basics on Centrally Specifying Function System Signatures (🕮 TargetLink Customization and Optimization Guide)
- How to Specify Function Block Data (🕮 TargetLink Customization and Optimization Guide)
- How to Specify System Signature Data (🕮 TargetLink Customization and Optimization Guide)

Related TargetLink API functions

For detailed information, also refer to the corresponding TargetLink API function: tlSyncSystemSignature (🕮 TargetLink API Reference).

Result

You have created or synchronized a Simulink or TargetLink system for a DD Signature object or a DD Function Block object.

Variant

Purpose

To select a variant configuration.

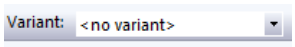
Preconditions

At least one VariantConfig object has to be created (in `/Config/VariantConfigs`).

Access

You can access this command via:

Menu bar	None
Context menu of	None

Shortcut key	None
Toolbar button	

Result

A variant is selected.

Description

The required variant configuration can be selected in the Variant list.

Related topics

HowTos

How to Activate Code Variants (📖 TargetLink Customization and Optimization Guide)

Object/Property-Related Commands

Where to go from here

Information in this section

Add Custom Property

Purpose

To add a new custom property to the selected DD object.

Access

You can access this command via:

Menu bar	None
Context menu of	DD objects
Shortcut key	None
Toolbar button	None

Result

After you select the necessary custom property in the submenu, it is added to the selected DD object.

Description

Custom properties are properties that you can add to DD objects and configure according to your requirements. In the **Property Value List**, the symbol indicates that a property is a custom property. For more information on custom properties, refer to Mapping Properties Between MATLAB and DD Variable Objects (MLIE).

Check Reference Properties

Purpose

To check the reference properties. The result is shown in the **Message Browser**.

Access

You can access this command via:

Menu bar	Tools
Context menu of	None
Shortcut key	None
Toolbar button	None

Result

You have checked the reference properties.

Description

Reference Properties are properties which point to objects. For example, the **Scaling** property of **Variable** objects points to the **Scaling** object which defines the Variables's scaling parameters. To check them, you can use this command. For details, refer to *Reference Properties* in the TargetLink Data Dictionary Reference.

Copy To

Purpose

You can copy DD objects to different DD workspaces in order to quickly transfer object settings from one DD workspace to another.

Access

You can access this command via:

Menu bar	None
Context menu of	DD objects
Shortcut key	None
Toolbar button	None

Result

The object with all its properties and child objects is copied to another DD workspace.

Description

The selected objects are copied to the destination DD with their path retained, relative to the root object. If the parent object, the copied child object belongs to, does not exist in the destination DD, it is created automatically. Otherwise the existing object is overwritten. Then the destination DD workspace opens with the new object highlighted.

If you do not want to overwrite existing objects, use the **Merge** command instead. It allows you to merge objects with other DD objects without overwriting them. Refer to Merge and Replace on page 94.

Related topics

References

Create <DD Object>

Purpose

To create a new <DD object> in the active DD workspace.

Access

You can access this command via:

Menu bar	None
Context menu of	DD objects
Shortcut key	None
Toolbar button	None

Result

The new <DD object> is created and displayed in the DD object tree.

Description

The expression *<DD object>* functions as a wildcard for all DD objects. The kind and amount of DD objects which can be created depend on the selected DD object in the DD object tree.

Related topics

HowTos

How to Create New Data Dictionary Objects (TargetLink Data Dictionary Basic Concepts Guide)

Create <Property>

Purpose

To create a new <property> in the Property Value List.

Access

You can access this command via:

Menu bar	None
Context menu of	DD objects
Shortcut key	None
Toolbar button	None

Result

The new <property> is created and displayed in the Property Value List.

Description

The expression *<property>* functions as a wildcard for all properties. The kind and amount of properties which can be created depend on the selected DD object.

Create Reference to <Object>

Purpose

To create a reference to another DD object in the current DD object.

Access

You can access this command via:

Menu bar	None
Context menu of	DD objects
Shortcut key	None
Toolbar button	None

Result

The new reference to another object is created in the current DD object and displayed in the property value list.

Cut

Purpose

To mark a DD object for removal.

Access

You can access this command via:

Menu bar	Edit
Context menu of	DD objects
Shortcut key	**Ctrl** + **X**
Toolbar button	

Result

The DD object is marked for removal. The actual removal takes place when (and only when) you paste the object.

Related topics

References

Paste 168

Delete/Unset Property

Purpose

To delete or unset the selected property.

Access

You can access this command via:

Menu bar	Edit
Context menu of	Property value list
Shortcut key	**Del**
Toolbar button	

Result

If the selected property is not mandatory, it is deleted, else it is reset.

Duplicate

Purpose

To duplicate a selected DD object with all its properties and DD child objects.

Access

You can access this command via:

Menu bar	Edit
Context menu of	DD objects
Shortcut key	**Ctrl** + **D**
Toolbar button	None

Result

The selected DD object is duplicated with all its properties and child objects.

Description

The command is executed only if more than one object of the same object kind is allowed at the selected position.

Related topics

References

Copy 150
Paste 168

Edit Object

Purpose

To open the editor, view and alter the settings for the selected DD object.

Access

You can access this command via:

Menu bar	None
Context menu of	DD objects
Shortcut key	None
Toolbar button	None

You can also open the editor by double-clicking the object in the Data Dictionary Navigator.

Result

The editor is opened.

Description

The editor that opens depends on the selected object kind. For more information on the different editors and their dialog settings, refer to Object Dialogs on page 39.

Related topics

Basics

Basics on Handling Multiple DD Objects with the Object Explorer (TargetLink Data Dictionary Basic Concepts Guide)

References

Object Dialogs 39

Edit Target

Purpose

To edit the DD object which is referenced by the currently selected property.

Access

You can access this command via:

Menu bar	None
Context menu of	Property value list
Shortcut key	None
Toolbar button	None

Result

The object which is referenced by the currently selected property is edited.

Description

If you have created a reference for a DD object, the edit dialog of this object opens when you use this command.

The command is available for references to **Variable** and **Typedef** objects.

Find

Purpose

To find a certain DD object in the DD object tree.

Access

You can access this command via:

Menu bar	Edit
Context menu of	DD objects
Shortcut key	**Ctrl** + **F**
Toolbar button	

Result

If a DD object with the specified search criteria is found, its first appearance is highlighted in the DD object tree, depending on which position the search was started at.

Tip

If you want to continue your search, use the **Find Next** command or press **F3**.

Description

With the search criteria, an **AND** operation is applied, i.e., all the specified criteria must be matched. During the search, empty fields are ignored. Therefore you have to use the **Find empty** option, if you want to find an object with a property that is not set.

Dialog settings

Object name Lets you specify the name of the DD object you are searching for. You can use wildcards or regular expressions in your search string.

Property name Lets you specify the name of a property of the DD object you are searching for. You can use wildcards or regular expressions in your search string.

Property value Lets you specify the value of a property of the DD object you are searching for as a string. You can use wildcards or regular expressions in your search string.

Find empty Lets you select whether to search for a property value whose value is empty.

Start at Lets you specify the DD object where the search should be started.

Object kind Lets you choose an object kind.

Case sensitive Lets you select whether the search should observe capitalization.

Match rule Lets you select how the specified search criteria should match the object during the search.

Match rule	Description	
Wildcard	Specifies that the specified search criteria may contain wildcards.	**?** matches any single character. * matches 0, 1 or any number of characters.
Regular Expression	Specifies that the specified search criteria are regular expressions.	
Exact	Specifies that the specified search criteria should match the object exactly.	

Find Next Lets you continue the search with the specified search criteria.

Find All Lets you find all the objects that match the specified search criteria. The Find Object Results on page 25 pane opens and displays the search results.

Related topics

HowTos

How to Find Data Dictionary Object References (TargetLink Data Dictionary Basic Concepts Guide)

References

Find Next 164

Find Next

Purpose

To continue a search.

Access

You can access this command via:

Menu bar	Edit
Context menu of	None
Shortcut key	**F3**
Toolbar button	
Button in	Find Object dialog

Result

The next DD object which matches the previously specified search criteria is highlighted in the DD object tree.

Related topics

HowTos

How to Find Data Dictionary Object References (TargetLink Data Dictionary Basic Concepts Guide)

References

Find 162

Find References

Purpose

To find DD objects that reference the selected DD object.

Access

You can access this command via:

Menu bar	None
Context menu of	DD objects
Shortcut key	None
Toolbar button	None

Result

The **Find References Results** pane opens and displays the references to the selected object. You can click on the reference to navigate to the respective object.

Description

Depending on its object kind, a DD object can be referenced by various kinds of objects:

- Other DD objects
- Simulink blocks some of which are based on objects in libraries
- Stateflow objects some of which are based on objects in libraries

Note

By default, the **Find References** command also searches in referenced models. These models are opened automatically. If references are found, the models remain open. Otherwise, they are closed.

Related topics

HowTos

How to Find Data Dictionary Object References (TargetLink Data Dictionary Basic Concepts Guide)

References

Find References Results..... 26

tlFindDDReferences (TargetLink API Reference)

Goto Target

Purpose

To go to the DD object which is referenced by the currently selected property.

Access

You can access this command via:

Menu bar	None
Context menu of	Property value list
Shortcut key	None
Toolbar button	None

Result

The DD object which is referenced by the currently selected property is selected.

Description

If you have created a reference for a DD object, this object is selected when you use the **Goto Target** command.

Move

Purpose

To rearrange DD objects within the same hierarchy level or to rearrange properties.

Access

You can access this command via:

Menu bar	None
Context menu of	DD objects Property Value List
Shortcut key	**Ctrl** + **arrow keys up** and **Ctrl** + **arrow keys down**
Toolbar button	None

Result

The DD objects in the DD object tree or the properties are shown in the desired order.

Description

DD objects can be moved up, down, topmost or bottommost in the same hierarchy level. Properties can be moved up and down.

Tip

To move DD objects across the same hierarchy level you can use drag & drop or cut & paste. When using this feature the drag source always becomes a child of the drop target.

Next

Purpose

To switch back to the DD object you selected before.

Access

You can access this command via:

Menu bar	None
Context menu of	None
Shortcut key	None
Toolbar button	

Result

You have switched back to the object you selected before. This does not work if you did not use the Previous command.

Related topics

References

Previous..... 169

Open Editor

Purpose

To open the editor, view and alter the settings for the selected property.

Access

You can access this command via:

Menu bar	None
Context menu of	Property value list

Shortcut key	None
Toolbar button	

You can also open the editor by clicking the Edit button ··· in the property value list.

Result

The editor is opened.

Description

The editor that opens depends on the selected property type. For more information on the different editors and their dialog settings, refer to Basic Dialogs on page 29.

Related topics

References

Paste

Purpose

To paste contents (temporarily saved with the Cut or Copy command) into the currently open TargetLink Data Dictionary.

Access

You can access this command via:

Menu bar	Edit
Context menu of	DD objects
Shortcut key	**Ctrl** + **V**
Toolbar button	

Result

If the temporary memory contents are valid for the currently open DD workspace, they are pasted.

Description

The temporary memory of the TargetLink Data Dictionary Manager is independent of the Windows Clipboard.

Related topics

References

Copy 150
Cut 159

Previous

Purpose

To switch to the previously selected DD object.

Access

You can access this command via:

Menu bar	None
Context menu of	None
Shortcut key	None
Toolbar button	

Result

The previously selected DD object is selected.

Related topics

References

Next 167

Reference Handling Options

Purpose

If DD objects are moved or renamed, the references to the DD object become invalid. The Data Dictionary Manager automatically adapts the references in the active DD workspace, the root level model, and libraries to keep them valid. The Reference Handling Options lets you specify the way the references are adapted.

Access

You can access this command via:

Menu bar	Edit
Context menu of	None
Shortcut key	None
Toolbar button	None

Result

You specified the way references in DD objects are adapted.

Description

Show Context Menu If this option is set, the Reference Handling Options context menu opens when you move or rename DD objects.

Show Selection Dialog If this option is set, a selection dialog opens when you move or rename DD objects. Via the selection dialog, you can specify the object paths (Adapt references in) in which references are adapted.

Never Adapt References If this option is set, no references are adapted.

Adapt References Only in DD If this option is set, references are adapted in the active DD workspace. The root level model and libraries remain unchanged.

Adapt References in DD and Models If this option is set, references are adapted in the active DD workspace and the root level model. Libraries remain unchanged.

Adapt References in DD, Models and Libraries (default) If this option is set, references are adapted in the active DD workspace, the root level model, and libraries.

Adapt References in DD, Models, Libraries and Referenced Models If this option is set, the references are adapted in the active DD workspace, the root level model, libraries, and referenced models.

Note

All settings are automatically saved and remain for the next start of the TargetLink Data Dictionary Manager.

Related topics

References

Move 166
Rename 171

Rename

Purpose

To rename the selected DD object or property.

Access

You can access this command via:

Menu bar	Edit
Context menu of	DD objects Property Value List
Shortcut key	**F2**
Toolbar button	

Result

The selected DD object or property is renamed.

Related topics

HowTos

How to Rename Data Dictionary Objects (TargetLink Data Dictionary Basic Concepts Guide)

Reset

Purpose

To reset the properties of a predefined DD object to their default values.

Access

You can access this command via:

Menu bar	None
Context menu of	Predefined DD objects
Shortcut key	None
Toolbar button	None

Result

The properties of the selected predefined DD object are reset to their default values.

Description

The properties of a predefined DD object, such as the TLPredefinedOptionSet object, can be changed manually. To use the object with its original properties, you can reset it to its default values.

Show Associated Block

Purpose

To show the Simulink model, subsystem, or block associated with the DD object currently selected in the Subsystems area of the TargetLink Data Dictionary.

Access

You can access this command via:

Menu bar	None
Context menu of	None
Shortcut key	None
Toolbar button	

Result

The Simulink model, subsystem, or block associated with the currently selected DD object is shown.

Description

A Simulink model must be open.

Note

This command is not available in the stand-alone mode of the Data Dictionary Manager.

Show Property in Object Explorer

Purpose

To show the selected property in the Object Explorer.

Access

You can access this command via:

Menu bar	None
Context menu of	Property value list

Shortcut key	None
Toolbar button	None

Tip

You can also add a property to the Object Explorer using drag & drop. Drag the selected property to the Object Explorer.

Result

The selected property is shown in the Object Explorer.

Related topics

HowTos

How to Specify Properties (Columns) Shown in the Object Explorer (📖 TargetLink Data Dictionary Basic Concepts Guide)

Show Property Types

Purpose

To show or hide the property types in the property value list.

Access

You can access this command via:

Menu bar	View
Context menu of	None
Shortcut key	None
Toolbar button	None

Result

The property types are shown or hidden in the property value list.

Show Unset Properties

Purpose

To show or hide properties in the current window, which are defined in the Data Model but are currently not set.

Access

You can access this command via:

Menu bar	View
Context menu of	None
Shortcut key	None
Toolbar button	

Result

The Properties which are defined in the Data Model but are currently not set are shown/hidden in the current window.

Sort

Purpose

To sort the children of a DD object in alphabetical order and by object kind.

Access

You can access this command via:

Menu bar	None
Context menu of	DD objects
Shortcut key	**F4**
Toolbar button	

Result

The children of the selected DD object are sorted alphabetically and by object kind.

Description

The command changes not only the way the DD objects are displayed in the Data Dictionary Navigator but also the order of the DD objects inside the DD object tree. Therefore the command is not available with all DD objects, for example, the children of the `/Pool` object cannot be sorted.

Up

Purpose

To move up one hierarchy level in the DD Navigator.

Access

You can access this command via:

Menu bar	None
Context menu of	None
Shortcut key	None
Toolbar button	

Result

You have moved up one hierarchy level in the DD Navigator.

Pane-Related Commands

Where to go from here

Information in this section

Docking

Purpose

To connect the currently selected floating pane to its neighboring panes.

Access

You can access this command via:

Menu bar	None
Context menu of	Panes in floating mode
Shortcut key	None
Toolbar button	None

Result The pane is connected to its neighboring panes.

Related topics

References

Floating 177
Tabbed Document 185

Floating

Purpose To make the currently selected pane movable over the entire screen.

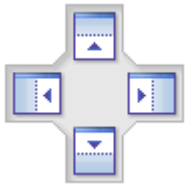
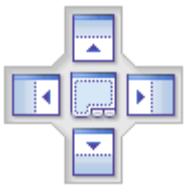
Access You can access this command via:

Menu bar	None
Context menu of	Panes
Shortcut key	None
Toolbar button	None

Result The selected pane is movable.

Description

While a pane is in floating mode, you can use docking stickers to arrange the pane. Refer to the table below.

Docking Sticker	Description
	The pane is docked to the top of your application's main window.
	The pane is docked to the bottom of your application's main window.
	The pane is docked to the left of your application's main window.
	The pane is docked to the right of your application's main window.
	The pane is docked to the top, bottom, left, or right of your application's working area.
	The pane is docked above, below, to the left, or to the right of the selected pane. If you drag the mouse onto the middle docking sticker, the pane is docked as a new tab.

Related topics

References

Docking 176
Tabbed Document 185

Hide

Purpose

To hide the currently selected pane.

Access

You can access this command via:

Menu bar	None
Context menu of	Panes

Shortcut key	None
Toolbar button	None

Result

The currently selected pane is hidden.

Note

This command is not available for the DD0 workspace.

Object Explorer - Depth

Purpose

To set the depth of the hierarchy of DD objects to be displayed in the Object Explorer.

Access

You can access this command via:

Menu bar	View
Context menu of	None
Shortcut key	None
Toolbar button	None

Result

You have set the depth of the hierarchy of DD objects.

Description

Selected Objects Only To display the currently selected DD object in the Object Explorer.

One Level To display the currently selected DD object and its child objects which are one level below in the DD object tree in the Object Explorer.

All Levels To display the currently selected DD object and all of its child objects in the Object Explorer.

Related topics

HowTos

How to Specify Properties (Columns) Shown in the Object Explorer (TargetLink Data Dictionary Basic Concepts Guide)

Object Explorer - Filter

Purpose

To set filter criteria in the Object Explorer.

Access

You can access this command via:

Menu bar	View
Context menu of	None
Shortcut key	None
Toolbar button	None

Result

You have set filter criteria in the Object Explorer.

Description

The following filter criteria apply:

All Objects To display all object kinds in the Object Explorer.

Only Variables To display only Variable objects in the Object Explorer.

Only Scalings To display only Scaling objects in the Object Explorer.

Only Typedefs To display only Typedef objects in the Object Explorer.

Only Variable Classes To display only VariableClass objects in the Object Explorer.

Related topics

HowTos

How to Specify Properties (Columns) Shown in the Object Explorer (📖 TargetLink Data Dictionary Basic Concepts Guide)

Object Explorer - Show Add/Remove Column

Purpose

To show or hide the Show in Object Explorer column in the Object Explorer.

Access

You can access this command via:

Menu bar	View
Context menu of	None

Shortcut key	None
Toolbar button	None

Description

You can select the following options for the **Show Add/Remove Column** command:

- **Never**: The **Show in Object Explorer** column is always hidden.
- **Automatically** (default): Show/hide is controlled by the TargetLink Data Dictionary Manager.
- **Always**: The **Show in Object Explorer** column is always shown.

Result

The **Show in Object Explorer** column in the **Object Explorer** is shown or hidden.

Related topics

HowTos

How to Specify Properties (Columns) Shown in the Object Explorer (TargetLink Data Dictionary Basic Concepts Guide)

Object Explorer - Show All Selected Properties

Purpose

To show or hide columns in the Object Explorer for properties which are selected in the Property Selector but do not exist at the displayed DD objects.

Access

You can access this command via:

Menu bar	View
Context menu of	None
Shortcut key	None
Toolbar button	

Result

The columns for properties which are selected in the Property Selector but do not exist at the displayed DD objects are shown or hidden in the Object Explorer.

Related topics

HowTos

How to Specify Properties (Columns) Shown in the Object Explorer (📖 TargetLink Data Dictionary Basic Concepts Guide)

Remove All Columns

Purpose

To remove all columns from the table displayed in the Object Explorer.

Access

You can access this command via:

Menu bar	None
Context menu of	Header bar in the Object Explorer
Shortcut key	None
Toolbar button	None

Result

All columns are removed.

Remove Column

Purpose

To remove a column from the table displayed in the Object Explorer.

Access

You can access this command via:

Menu bar	None
Context menu of	Header bar in the Object Explorer
Shortcut key	None
Toolbar button	None

Result

The selected column is removed.

Tip

You can also remove a column by clearing the respective check mark in the Property Selector.

Show Object Explorer

Purpose

To show or hide the Object Explorer.

Access

You can access this command via:

Menu bar	View
Context menu of	None
Shortcut key	None
Toolbar button	

Result

The Object Explorer is shown or hidden in a pane of the TargetLink Data Dictionary Manager.

Related topics

References

Object Explorer 19

Show Output View

Purpose

To show or hide the Details, Embedded Help, and Message Browser panes.

Access

You can access this command via:

Menu bar	View-Show Output View- ▪ Details ▪ Embedded Help ▪ Message Browser
Context menu of	Each DD object and property (Details only)
Shortcut key	None
Toolbar button	 (Message Browser)

Result

The Details, Embedded Help and Message Browser panes are displayed or hidden. If Details and Embedded Help are not available, you can use Show Details on page 119 and Show Embedded Help on page 120 from **Help** menu to display these panes. For more information, refer to Panes of the GUI on page 11.

Related topics

References

Details....23
Embedded Help....17
Message Browser....21
Show Details....119
Show Embedded Help....120

Show Property Selector

Purpose

To show or hide the Property Selector.

Access

You can access this command via:

Menu bar	View
Context menu of	None
Shortcut key	None
Toolbar button	

Result

The Property Selector is shown or hidden.

Description

The Property Selector can be shown only if the Object Explorer is displayed. If the Object Explorer is hidden, the Property Selector is hidden too, and the command is disabled.

Related topics

References

Property Selector....20

Tabbed Document

Purpose

To arrange the panes as tabbed pages.

Access

You can access this command via:

Menu bar	None
Context menu of	Panes
Shortcut key	None
Toolbar button	None

Result

The panes are arranged as tabbed pages.

Related topics

References

Docking..... 176
Floating..... 177

Privilege-Related Commands

Where to go from here

Information in this section

Access

Purpose

To set the access rights (administrator or user) for specific DD objects.

Access

You can access this command via:

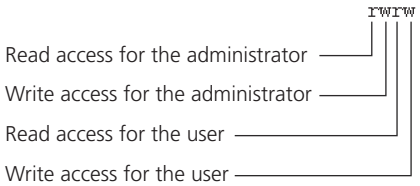
Menu bar	None
Context menu of	DD objects
Shortcut key	None
Toolbar button	None

Result

The access rights (administrator or user) are set for specific DD objects.

Description

Two users are known to the Data Dictionary Manager, the administrator and the user. The following access properties are supported:



Properties that have read-only access are indicated by a symbol.

Dialog settings

Set the access rights to Lets you specify the access properties as described above.

For this object only Indicates that the access rights are set only for the selected DD object.

For child objects only Indicates that the access rights are set only for the child objects.

For this object and all of its children Indicates that the access rights are set for the selected DD object and its direct children.

For all objects in this subtree Indicates that the access rights are set for all DD objects in the selected subtree.

Related topics

Basics

Basics on User Modes and Access Rights (📖 TargetLink Data Dictionary Basic Concepts Guide)

HowTos

How to Change Access Rights (📖 TargetLink Data Dictionary Basic Concepts Guide)

Change to Administrator/User Mode

Purpose

To change to the administrator or user mode.

Access

You can access this command via:

Menu bar	Extras
Context menu of	None
Shortcut key	None
Toolbar button	None

Result

The selected mode is set. The mode that is set is displayed in the status bar.

Description

Two user roles are known to the TargetLink Data Dictionary, the administrator and the user. By default, the user mode is set. Reading and writing data to and from the Data Dictionary Manager is performed with user role specified here. If

the user mode is set, you cannot access DD properties which have no read access for the user.

Related topics

HowTos

How to Switch User Modes (TargetLink Data Dictionary Basic Concepts Guide)

Set Admin Password

Purpose

To set a password for the admin mode.

Access

You can access this command via:

Menu bar	Extras
Context menu of	None
Shortcut key	None
Toolbar button	None

Result

The currently open DD workspace is protected by an admin password.

Description

The password must not exceed 16 characters. If a password already exists, you are prompted to enter it. If no password exists, you must enter a new one. If you want to reset a password of a DD project file, enter an empty password.

Dialog settings

Enter new admin password Lets you enter a new admin password.

Re-enter new admin password Lets you enter the new admin password again.

Related topics

HowTos

How to Switch User Modes (TargetLink Data Dictionary Basic Concepts Guide)

Workspace-Related Commands

Where to go from here

Information in this section

<Name> (DDn)

Purpose

To change to the selected DD workspace. All active DD workspaces are listed in the **Window** menu.

Access

You can access this command via:

Menu bar	Window
Context menu of	None
Shortcut key	None
Toolbar button	None

Result

You have changed to the selected workspace.

Close All DD Workspace Panes

Purpose

To close all DD workspace panes.

Access

You can access this command via:

Menu bar	Window
Context menu of	None
Shortcut key	None
Toolbar button	None

Result

To close all the workspace panes. If the workspace panes are closed, the workspaces itself remains in memory and can be opened again by clicking **Show DD Workspace Pane**.

Note

You cannot close the DD0 workspace tab. The **Close All DD Workspace Panes** command only works if you have more than one DD workspace pane opened.

Related topics

References

Show DD Workspace Pane..... 198

Close DD Workspace Pane

Purpose

To close the active DD workspace pane (except the DD0 workspace pane).

Access

You can access this command via:

Menu bar	Window
Context menu of	None
Shortcut key	None
Toolbar button	None

Result

To close the active workspace pane. If the DD workspace pane is closed, the workspace itself remains in memory and can be opened again by clicking **Show DD Workspace Pane**.

Note

You cannot close the DD0 workspace pane. The **Close DD Workspace Pane** command only works if you have more than one DD workspace pane opened.

Related topics

References

Show DD Workspace Pane..... 198

Make Primary DD Workspace

Purpose

To switch the active DD workspace with the DD0 workspace. This command is identical with the **Swap with DD0** command in the **DD Workspace Overview**. Refer to How to Swap a Secondary DD Workspace with the Primary DD Workspace (TargetLink Data Dictionary Basic Concepts Guide).

Access

You can access this command via:

Menu bar	None
Context menu of	None
Shortcut key	None
Toolbar button	

Result

The selected DD workspace is now DD0.

Related topics

HowTos

How to Swap a Secondary DD Workspace with the Primary DD Workspace (TargetLink Data Dictionary Basic Concepts Guide)

New - Create New DD Workspace

Purpose

To create a new Data Dictionary in a new DD workspace.

Access

You can access this command via:

Menu bar	File – New
Context menu of	None
Shortcut key	None
Toolbar button	None

Result

A new DD workspace is created. The workspace name of this new, currently active DD workspace is displayed in the title bar of the Data Dictionary Manager.

Note

The DD1 ... DD3 workspaces are used internally by TargetLink. If you create a new DD workspace, it is named DD4, DD5, etc.

Description

The Select DD Project File Template dialog opens. Use the relevant radio buttons to select one of these two options:

- Pre-defined template
- Other template

Predefined system templates

When you create a new DD workspace, you can either choose between 4 predefined system templates or select other templates.

dsdd_master_basic.dd [System] Provides the new DD workspace with a basic set of predefined DD objects, such as DD Typedef, DD VariableClass, and DD FunctionClass objects.

dsdd_master_advanced.dd [System] Provides the new DD workspace with additional predefined objects, such as DD Template objects and DD RTOS objects required for multirate code generation.

dsdd_master_autosar4.dd [System] Provides the new DD workspace with additional predefined objects, such as DD Typedef and DD Variable objects to be used in AUTOSAR SWCs, and a DD object for specifying AUTOSAR software components and interfaces. Use this template to work with the AUTOSAR 4.x standard.

User-defined templates

You can use any DD file containing a complete DD workspace with a valid data model as a user-defined template.

Tip

If you place a TXT file with the name of the DD file you want to use as a user-defined template, TargetLink automatically adds the contained text as a description for the template. This is useful if you share DD files with your team members: For example, you can post the contents of the template and your last changes.

Related topics

HowTos

How to Create DD Workspaces (TargetLink Data Dictionary Basic Concepts Guide)

New – Use Active DD Workspace

Purpose

To create a new Data Dictionary in the active DD workspace.

The content of the active DD workspace is cleared and new content is created, obtaining the active workspace name (e.g., DD0).

Access

You can access this command via:

Menu bar	File – New
Context menu of	None
Shortcut key	**Ctrl** + **N**
Toolbar button	None

Result

A new Data Dictionary is created in the active DD workspace. The workspace name of the active DD workspace is displayed in the title bar of the Data Dictionary Manager.

If you want to use a **User-defined template**, click the **Open DD Project File** icon. If the model folder contains a text file with the same name as the DD project file, its content is displayed in the **Create New DD Workspace** dialog as additional information.

Description

The Select DD Template dialog opens. Use the relevant radio buttons to select one of these two options:

- a predefined System template
- a User-defined template

System template

You are prompted to select a template to copy the settings for the new DD workspace. The following templates are available:

dsdd_master_basic.dd [System] Provides the new DD workspace with a basic set of predefined DD objects, such as DD Typedef, DD VariableClass, and DD FunctionClass objects.

dsdd_master_advanced.dd [System] Provides the new DD workspace with additional predefined objects, such as DD Template objects and DD RTOS objects required for multirate code generation.

dsdd_master_autosar4.dd [System] Provides the new DD workspace with additional predefined objects, such as DD Typedef and DD Variable objects to be used in AUTOSAR SWCs, and a DD object for specifying AUTOSAR software components and interfaces. Use this template to work with the AUTOSAR 4.x standard.

User-defined template

Lets you browse for a custom template to copy the settings for the new DD workspace.

Related topics

HowTos

How to Create DD Workspaces (TargetLink Data Dictionary Basic Concepts Guide)

Open - Create New DD Workspace

Purpose

To open a DD file in a new DD workspace.

Access

You can access this command via:

Menu bar	File - Open
Context menu of	None
Shortcut key	None
Toolbar button	None

Result

The selected DD file is opened in a new DD workspace.

Note

The DD1 ... DD3 workspaces are used internally by TargetLink. If you create a new DD workspace, it is named DD4, DD5, etc.

Opening multiple DD files

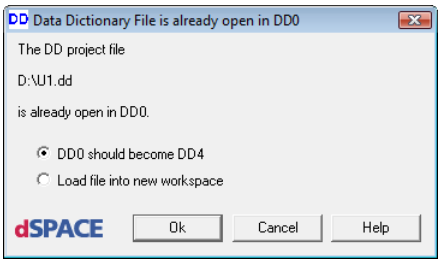
It is now possible to load multiple DD files using the Open command in the File menu of your Data Dictionary Manager. However, only specific combinations of DD files are allowed:

- You can multi-select files containing one complete Data Dictionary and one or more partial Data Dictionaries.
- You can multi-select files containing complete Data Dictionaries.
- You can multi-select files containing partial Data Dictionaries.

Special loading behavior

Some constellations when opening one or more DD files with this command evoke a special loading behavior:

- The content of a DD project file you want to open in an new DD workspace already exists in another DD workspace. You are prompted to choose an option as shown below:



- You multi-selected one DD project file and multiple partial DD files: The DD project file is loaded into a new DD workspace. The partial Data Dictionaries are loaded into the same DD workspace. If they contain metadata about their positions in the DD object tree, they are inserted at those positions. If they contain no metadata, they are inserted at valid positions according to the data model. If this is not possible, they are loaded into `/tmp`.
- You multiselected multiple partial DD files: The files are loaded into the new DD workspace. If they contain metadata about their positions in the DD object tree, they are inserted at those positions. If they contain no metadata, they are inserted at valid positions according to the data model. If this is not possible, they are loaded into `/tmp`.
- The selected files contain no valid data: No action.
- The selected files contain multiple DD project files and one or more partial DD files: No action, because this combination is not possible.

Related topics

HowTos

How to Open DD Project Files in DD Workspaces (TargetLink Data Dictionary Basic Concepts Guide)

Open - Use Active DD Workspace

Purpose

To open a DD file in the active DD workspace.

Access

You can access this command via:

Menu bar	File - Open
Context menu of	None
Shortcut key	**Ctrl + O**
Toolbar button	None

Result

The selected DD file is opened.

- For DD project files: The content of the active DD workspace is *cleared* and the content of the DD project file is loaded into the DD workspace (you are prompted to save unsaved changes).
- For partial DD files: The DD file is loaded to the location that its metadata contains. If no metadata is available, a valid location according to the data model is used. Otherwise, it is loaded into `/tmp`.

Opening multiple DD files

It is now possible to load multiple DD files using the Open command in the File menu of your Data Dictionary Manager. However, only specific combinations of DD files are allowed:

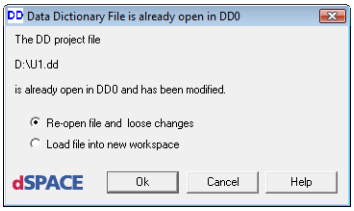
- You can multi-select files containing one complete Data Dictionary and one or more partial Data Dictionaries.
- You can multi-select files containing complete Data Dictionaries.
- You can multi-select files containing partial Data Dictionaries.

Special loading behavior

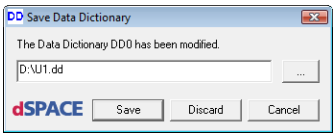
Some constellations when opening one or more DD files evoke a special loading behavior:

- The content of a DD project file you want to open already exists in the active DD workspace: The DD file is reloaded into the active DD workspace.

- The content of a DD project file you want to open is different from the content of your active DD workspace: The DD file is loaded into the active DD workspace (the content is overwritten).
- The content of a DD project file you want to open already exists in the active DD workspace, but the active DD workspace contains unsaved changes. You are prompted to choose an option as shown below:



- The content of a DD project file you want to open is different from the content of your active DD workspace, which contains unsaved changes. You are prompted to choose an option as shown below:



  After you chose **Save** or **Discard**, the content of the different DD project file is loaded into the active DD workspace. **Cancel** aborts the **Open** command.
- You multi-selected one DD project file and multiple partial DD files: The DD project file is loaded into the active DD workspace. (You are prompted if your current workspace contains unsaved changes.) Next, the partial DD files are loaded. If they contain metadata about their positions in the DD object tree, they are inserted at those positions. If they contain no metadata, they are inserted at valid positions according to the data model. If this is not possible, they are loaded into `/tmp`.
- You multi-selected multiple partial DD files: The files are loaded into the active DD workspace. If they contain metadata about their positions in the DD object tree, they are inserted at those positions. If they contain no metadata, they are inserted at valid positions according to the data model. If this is not possible, they are loaded into `/tmp`.
- The selected files contain no valid data: No action.
- The selected files contain multiple DD project files and one or more partial DD files: No action, because this combination is not possible.

Related topics

HowTos

How to Open DD Project Files in DD Workspaces (TargetLink Data Dictionary Basic Concepts Guide)

References

Load 107

Show DD Workspace Overview

Purpose

To show the DD Workspace Overview pane.

Access

You can access this command via:

Menu bar	▪ Window ▪ View
Context menu of	None
Shortcut key	None
Toolbar button	None

Result

You have opened the DD Workspace Overview pane.

Description

For a detailed description of the DD Workspace Overview pane, refer to DD Workspace Overview on page 14.

Show DD Workspace Pane

Purpose

All existing DD workspaces, even those without an associated DD project file, are listed in the submenu of this command. You can select one to open it.

Access

You can access this command via:

Menu bar	Window
Context menu of	None

Shortcut key	None
Toolbar button	None

Result You have listed all non-active DD project files.

Show in DD Workspace Pane

Purpose To highlight a DD object in the DD workspace pane.

Access You can access this command via:

Menu bar	None
Context menu of	DD two-way comparison DD three-way merge
Shortcut key	None
Toolbar button	None

Result The DD object is highlighted in the DD Workspace pane.

Glossary

Introduction

The glossary briefly explains the most important expressions and naming conventions used in the TargetLink documentation.

Where to go from here

Information in this section

Numerics

1-D look-up table A look-up table that maps one input value (x) to one output value (y).

2-D look-up table A look-up table that maps two input values (x,y) to one output value (z).

A

Abstract interface An interface that allows you to map a project-specific, physical specification of an interface (made in the TargetLink Data Dictionary) to a logical interface of a modular unit. If the physical interface changes, you do not have to change the Simulink subsystem or the partial DD file and therefore neither the generated code of the modular unit.

Access function (AF) A C function or function-like preprocessor macro that encapsulates the access to an interface variable.

See also read/write access function and variable access function.

Acknowledgment Notification from the RTE that a data element or an event message have been transmitted.

Activating RTE event An RTE event that can trigger one or more runnables.

See also activation reason.

Activation reason The activating RTE event that actually triggered the runnable.

Activation reasons can group several RTE events.

Active page pointer A pointer to a data page. The page referenced by the pointer is the active page whose values can be changed with a calibration tool.

Adaptive AUTOSAR Short name for the AUTOSAR *Adaptive Platform* standard. It is based on a service-oriented architecture that aims at on-demand software updates and high-end functionalities. It complements Classic AUTOSAR.

Adaptive AUTOSAR behavior code Code that is generated for model elements in Adaptive AUTOSAR Function subsystems or Method Behavior subsystems. This code represents the behavior of the model and is part of an adaptive application. Must be integrated in conjunction with ARA adapter code.

Adaptive AUTOSAR Function A TargetLink term that describes a C++ function representing a partial functionality of an adaptive application. This function can be called in the C++ code of an adaptive application. From a higher-level perspective, Adaptive AUTOSAR functions are analogous to runnables in Classic AUTOSAR.

Adaptive AUTOSAR Function subsystem An atomic subsystem used to generate code for an Adaptive AUTOSAR Function. It contains a **Function** block whose **AUTOSAR mode** is set to `Adaptive` and whose **Role** is set to `Adaptive AUTOSAR Function`.

ANSI C Refers to C89, the C language standard ANSI X3.159-1989.

Application area An optional DD object that is a child object of the DD root object. Each **Application** object defines how an ECU program is built from the generated subsystems. It also contains some experiment data, for example, a list of variables to be logged during simulations and results of code coverage tests.

Build objects are children of **Application** objects. They contain all the information about the binary programs built for a certain target platform, for example, the symbol table for address determination.

Application data type Abstract type for defining types from the application point of view. It allows you to specify physical data such as measurement data. Application data types do not consider implementation details such as bit-size or endianness.

Application data type (ADT) According to AUTOSAR, application data types are used to define types at the application level of abstraction. From the application point of view, this affects physical data and its numerical representation. Accordingly, application data types have physical semantics but do not consider implementation details such as bit width or endianness.

Application data types can be constrained to change the resolution of the physical data's representation or define a range that is to be considered.

See also implementation data type (IDT).

Application layer The topmost layer of the ECU software. The application layer holds the functionality of the ECU software and consists of atomic software components (atomic SWCs).

ARA adapter code Adapter code that connects Adaptive AUTOSAR behavior code with the Adaptive AUTOSAR API or other parts of an adaptive application.

Array-of-struct variable An array-of-struct variable is a structure that either is non-scalar itself or that contains at least one non-scalar substructure at any nesting depth. The use of array-of-struct variables is linked to arrays of buses in the model.

Artifact A file generated by TargetLink:

- Code coverage report files
- Code generation report files
- Metadata files
- Model-linked code view files
- Production code files
- Simulation application object files
- Simulation frame code files
- Stub code files

Artifact location A folder in the file system that contains an artifact. This location is specified relatively to a project folder.

ASAP2 File Generator A TargetLink tool that generates ASAP2 files for the parameters and signals of a Simulink model as specified by the corresponding TargetLink settings and generated in the production code.

ASCII In production code, strings are usually encoded according to the ASCII standard. The ASCII standard is limited to a set of 127 characters implemented by a single byte. This is not sufficient to display special characters of different languages. Therefore, use another character encoding, such as UTF-8, if required.

Asynchronous operation call subsystem A subsystem used when modeling *asynchronous* client-server communication. It is used to generate the call of the `Rte_Call` API function and for simulation purposes.

See also operation result provider subsystem.

Asynchronous server call returns event An RTE event that specifies whether to start or continue the execution of a runnable after the execution of a server runnable is finished.

Atomic software component (atomic SWC) The smallest element that can be defined in the application layer. An atomic SWC describes a single functionality and contains the corresponding algorithm. An atomic SWC communicates with the outside only via the interfaces at the SWC's ports. An atomic SWC is defined by an internal behavior and an implementation.

Atomic software component instance An atomic software component (atomic SWC) that is actually used in a controller model.

AUTOSAR Abbreviation of *AUT*omotive *O*pen *S*ystem *AR*chitecture. The AUTOSAR partnership is an alliance in which the majority of OEMs, suppliers, tool providers, and semiconductor companies work together to develop and establish a de-facto open industry-standard for automotive electric/electronics (E/E) architecture and to manage the growing E/E complexity.

AUTOSAR import/export Exchanging standardized software component descriptions between AUTOSAR tools.

AUTOSAR subsystem An atomic subsystem that contains a Function block whose AUTOSAR mode property is set to `Classic`. See also operation subsystem, operation call with runnable implementation subsystem, and runnable subsystem.

AUTOSAR tool Generic term for the following tools that are involved in the ECU network software development process according to AUTOSAR:

- Behavior modeling tool
- System-level tool
- ECU-centric tool

TargetLink acts as a behavior modeling tool in the ECU network software development process according to AUTOSAR.

Autoscaling Scaling is performed by the Autoscaling tool, which calculates worst-case ranges and scaling parameters for the output, state and parameter variables of TargetLink blocks. The Autoscaling tool uses either worst-case ranges or simulated ranges as the basis for scaling. The upper and lower worst-case range limits can be calculated by the tool itself. The Autoscaling tool always focuses on a subsystem, and optionally on its underlying subsystems.

B

Basic software The generic term for the following software modules:

- System services (including the operating system (OS) and the ECU State Manager)
- Memory services (including the NVRAM manager)
- Communication services
- I/O hardware abstraction
- Complex device drivers

Together with the RTE, the basic software is the platform for the application layer.

Batch mode The mode for batch processing. If this mode is activated, TargetLink does not open any dialogs. Refer to How to Set TargetLink to Batch Mode (TargetLink Orientation and Overview Guide).

Behavior model A model that contains the control algorithm for a controller (function prototyping system) or the algorithm of the controlled system (hardware-in-the-loop system). Can be connected in ConfigurationDesk via model ports to build a real-time application (RTA). The RTA can be executed on real-time hardware that is supported by ConfigurationDesk.

Block properties Properties belonging to a TargetLink block. Depending on the kind of the property, you can specify them at the block and/or in the Data Dictionary. Examples of block properties are:

- Simulink properties (at a masked Simulink block)
- Logging options or saturation flags (at a TargetLink block)
- Data types or variable classes (referenced from the DD)
- Variable values (specified at the block or referenced from the DD)

Bus A bus consists of subordinate bus elements. A bus element can be a bus itself.

Bus element A bus element is a part of a bus and can be a bus itself.

Bus port block Bus Inport, Bus Outport are bus port blocks. They are similar to the TargetLink Input and Output blocks. They are virtual, and they let you configure the input and output signals at the boundaries of a TargetLink subsystem and at the boundaries of subsystems that you want to generate a function for.

Bus signal Buses combine multiple signals, possibly of different types. Buses can also contain other buses. They are then called nested buses.

Bus-capable block A block that can process bus signals. Like bus port blocks, they allow you to assign a type definition and, therefore, a variable class to all the bus elements at once. The following blocks are bus-capable:

- Constant
- Custom Code (type II) block
- Data Store Memory, Data Store Read, and Data Store Write

- Delay
- Function Caller
- ArgIn, ArgOut
- Merge
- Multiport Switch (Data Input port)
- Probe
- Sink
- Signal Conversion
- Switch (Data Input port)
- Unit Delay
- Stateflow Data
- MATLAB Function Data

C

Calibratable variable Variable whose value can be changed with a calibration tool during run time.

Calibration Changing the calibration parameter values of ECUs.

Calibration parameter Any ECU variable type that can be calibrated. The term *calibration parameter* is independent of the variable type's dimension.

Calprm Defined in a calprm interface. Calprms represent calibration parameters that are accessible via a measurement and calibration system.

Calprm interface An interface that is provided or required by a software component (SWC) via a port (AUTOSAR).

Calprm software component A special software component (SWC) that provides calprms. Calprm software components have no internal behavior.

Canonical In the DD, array-of-struct variables are specified canonically. Canonical means that you specify one array element as a representative for all array elements.

Catalog file (CTLG) A description of the content of an SWC container. It contains file references and file category information, such as source code files (`C` and `H`), object code files (such as `O` or `OBJ`), variable description files (`A2L`), or AUTOSAR files (`ARXML`).

Characteristic table (Classic AUTOSAR) A look-up table as described by Classic AUTOSAR whose values are measurable or calibratable. See also compound primitive data type

Classic AUTOSAR Short name for the AUTOSAR *Classic Platform* standard that complements Adaptive AUTOSAR.

Classic initialization mode The initialization mode used when the Simulink diagnostics parameter Underspecified initialization detection is set to `Classic`.

See also simplified initialization mode.

Client port A require port in client-server communication as described by Classic AUTOSAR. In the Data Dictionary, client ports are represented as DD ClientPort objects.

Client-server interface An interface that describes the operations that are provided or required by a software component (SWC) via a port (AUTOSAR).

Code generation mode One of three mutually exclusive options for generating TargetLink standard production code, AUTOSAR-compliant production code or RTOS-compliant (multirate RTOS/OSEK) production code.

Code generation unit (CGU) The smallest unit for which you can generate code. These are:

- TargetLink subsystems
- Subsystems configured for incremental code generation
- Referenced models
- DD CodeGenerationUnit objects

Code output style definition file To customize code formatting, you can modify a code output style definition file (XML file). By modifying this file, you can change the representation of comments and statements in the code output.

Code output style sheets To customize code formatting, you can modify code output style sheets (XSL files).

Code section A section of generated code that defines and executes a specific task.

Code size Amount of memory that an application requires specified in RAM and ROM after compilation with the target cross-compiler. This value helps to determine whether the application generated from the code files fits in the ECU memory.

Code variant Code variants lead to source code that is generated differently depending on which variant is selected (i.e., varianted at code generation time). For example, if the Type property of a variable has the two variants Int16 and Float32, you can generate either source code for a fixed-point ECU with one variant, or floating-point code with the other.

Compatibility mode The default operation mode of RTE generators. The object code of an SWC that was compiled against an application header generated in compatibility mode can be linked against an RTE generated in compatibility mode (possibly by a different RTE generator). This is due to using standardized data structures in the generated RTE code.

See also vendor mode.

Compiler inlining The process of replacing a function call with the code of the function body during compilation by the C compiler via inline expansion.

This reduces the function call overhead and enables further optimizations at the potential cost of larger code size.

Composition A structuring element in the application layer. A composition consists of software components and their interconnections via ports.

Compound primitive data type A primitive application data type (ADT) as defined by Classic AUTOSAR whose category is one of the following:

- COM_AXIS
- CUBOID
- CUBE_4
- CUBE_5
- CURVE
- MAP
- RES_AXIS
- VAL_BLK
- STRING

Compute-through-overflow (CTO) Calculation method for additions and subtraction where overflows are allowed in intermediate results without falsifying the final result.

Concern A concept in component-based development. It describes the idea that components separate their concerns. Accordingly, they must be developed in such a way that they provide the required functionality, are flexible and easy to maintain, and can be assembled, reused, or replaced by newer, functionally equivalent components in a software project without problems.

Config area A DD object that is a child object of the DD root object. The Config object contains configuration data for the tools working with the TargetLink Data Dictionary and configuration data for the TargetLink Data Dictionary itself. There is only one Config object in each DD workspace. The configuration data for the TargetLink Data Dictionary is a list of included DD files, user-defined views, data for variant configurations, etc. The data in the Config area is typically maintained by a Data Dictionary administrator.

ConfigurationDesk A dSPACE software tool for implementing and building real-time applications (RTA).

Constant value expression An expression for which the Code Generator can determine the variable values during code generation.

Constrained range limits User-defined minimum (Min) or maximum (Max) values that the user ensures will never be exceeded. The Code Generator relies on these ranges to make the generated production code more efficient. If no

Min/Max values are entered, the implemented range limits are used during production code generation.

Constrained type A DD Typedef object whose Constraints subtree is specified.

Container A bundle of files. The files are described in a catalog file that is part of the container. The files of a container can be spread over your file system.

Container Manager A tool for handling containers.

Container set file (CTS) A file that lists a set of containers. If you export containers, one container set file is created for every TargetLink Data Dictionary.

Conversion method A method that describes the conversion of a variable's integer values in the ECU memory into their physical representations displayed in the Measurement and Calibration (MC) system.

Custom code Custom code consists of C code snippets that can be included in production code by using custom code files that are associated with custom code blocks. TargetLink treats this code as a black box. Accordingly, if this code contains custom code variables you must specify them via custom code symbols.. See also external code.

Custom code symbol A variable that is used in a custom code file. It must be specified on the Interface page of custom code blocks.

Customer-specific C function An external function that is called from a Stateflow diagram and whose interface is made known to TargetLink via a scripting mechanism.

D

Data element Defined in a sender-receiver interface. Data elements are information units that are exchanged between sender ports, receiver ports and sender-receiver ports. They represent the data flow.

Data page A structure containing all of the calibratable variables that are generated during code generation.

Data prototype The generic term for one of the following:

- Data element
- Operation argument
- Calprm
- Interrunnable variable (IRV)
- Shared or PerInstance Calprm
- Per instance memory

Data receive error event An RTE event that specifies to start or continue the execution of a runnable related to receiver errors.

Data received event An RTE event that specifies whether to start or continue the execution of a runnable after a data element is received by a receiver port or sender-receiver port.

Data semantics The communication of data elements with last-is-best semantics. Newly received data elements overwrite older ones regardless of whether they have been processed or not.

Data send completed event An RTE event that specifies whether to start or continue the execution of a runnable related to a sender acknowledgment.

Data transformation A transformation of the data of inter-ECU communication, such as end-to-end protection or serialization, that is managed by the RTE via transformers.

Data type map Defines a mapping between implementation data types (represented in TargetLink by DD **Typedef** objects) and application data types.

Data type mapping set Summarizes all the data type maps and mode request type maps of a software component (SWC).

Data variant One of two or more differing data values that are generated into the same C code and can be switched during ECU run time using a calibratable variant ID variable. For example, the **Value** property of a gain parameter can have the variants 2, 3, and 4.

DataItemMapping (DIM) A DataItemMapping object is a DD object that references a ReplaceableDataItem (RDI) and a DD variable. It is used to define the DD variable object to map an RDI object to, and therefore also the implementation variable in the generated code.

DD child object The DD object below another DD object in the DD object tree.

DD data model The DD data model describes the object kinds, their properties and constraints as well as the dependencies between them.

DD file A DD file (*.dd) can be a DD project file or a partial DD file.

DD object Data item in the **Data Dictionary** that can contain DD child objects and DD properties.

DD object tree The tree that arranges all DD objects according to the DD data model.

DD project file A file containing the DD objects of a DD workspace.

DD root object The topmost DD object of the DD workspace.

DD subtree A part of the DD object tree containing a DD object and all its descendants.

DD workspace An independent organizational unit (central data container) and the largest entity that can be saved to file or loaded from a DD project file. Any number of DD workspaces is supported, but only the first (DD0) can be used for code generation.

Default enumeration constant Represents the default constant, i.e., the name of an enumerated value that is used for initialization if an initial value is required, but not explicitly specified.

Direct reuse The Code Generator adds the instance-specific variables to the reuse structure as leaf struct components.

E

ECU Abbreviation of *electronic control unit*.

ECU software The ECU software consists of all the software that runs on an ECU. It can be divided into the basic software, run-time environment (RTE), and the application layer.

ECU State Manager A piece of software that manages modes. An ECU state manager is part of the basic software.

Enhanceable Simulink block A Simulink® block that corresponds to a TargetLink simulation block, for example, the **Gain** block.

Enumerated value An enumerated value consists of an enumeration constant and a corresponding underlying integer value (enumeration value).

Enumeration constant An enumeration constant defines the name for an enumerated value.

Enumeration data type A data type with a specific name, a set of named enumerated values and a default enumeration constant.

Enumeration value An enumeration value defines the integer value for an enumerated value.

Event message Event messages are information units that are defined in a sender-receiver interface and exchanged between sender ports or receiver ports. They represent the control flow. On the receiver side, each event message is related to a buffer that queues the received messages.

Event semantics Communication of data elements with first-in-first-out semantics. Data elements are received in the same order they were sent. In simulations, TargetLink behaves as if data semantics was specified, even if you specified event semantics. However, TargetLink generates calls to the correct RTE API functions for data and event semantics.

ExchangeableWidth A DD object that defines code variants or improves code readability by using macros for signal widths.

Exclusive area Allows for specifying critical sections in the code that cannot preempt/interrupt each other. An exclusive area can be used to specify the mutual exclusion of runnables.

Executable application The generic term for offline simulation applications and real-time applications.

Explicit communication A communication mode in Classic AUTOSAR. The data is exchanged whenever data is required or provided.

Explicit object An explicit object is an object in production code that the Code Generator created from a direct specification made at a DD object or at a model element. For comparison, see implicit object.

Extern C Stateflow symbol A C symbol (function or variable) that is used in a Stateflow chart but that is defined in an external code module.

External code Existing C code files/modules from external sources (e.g., legacy code) that can be included by preprocessor directives and called by the C code generated by TargetLink. Unlike Custom code, external code is used as it is.

External container A container that is owned by the tool with that you are exchanging a software component but that is not the tool that triggers the container exchange. This container is used when you import files of a software component which were created or changed by the other tool.

F

Filter An algorithm that is applied to received data elements.

Fixed-Point Library A library that contains functions and macros for use in the generated production code.

Function AF The short form for an access function (AF) that is implemented as a C function.

Function algorithm object Generic term for either a MATLAB local function, the interface of a MATLAB local function or a local MATLAB variable.

Function class A class that represents group properties of functions that determine the function definition, function prototypes and function calls of a function in the generated production code. There are two types of function classes: predefined function class objects defined in the `/Pool/FunctionClasses` group in the DD and implicit function classes (default function classes) that can be influenced by templates in the DD.

Function code Code that is generated for a modular unit that represents functionality and can have abstract interfaces to be reused without changes in different contexts, e.g. in different integration models.

Function inlining The process of replacing a function call with the code of the function body during code generation by TargetLink via inline expansion. This reduces the function call overhead and enables further optimizations at the potential cost of larger code size.

Function interface An interface that describes how to pass the inputs and outputs of a function to the generated production code. It is described by the function signature.

Function subsystem A subsystem that is atomic and contains a Function block. When generating code, TargetLink generates it as a C function.

Functional Mock-up Unit (FMU) An archive file that describes and implements the functionality of a model based on the Functional Mock-up Interface (FMI) standard.

G

Global data store The specification of a DD DataStoreMemoryBlock object that references a variable and is associated with either a Simulink.Signal object or Data Store Memory block. The referenced variable must have a module specification and a fixed name and must be global and non-static. Because of its central specification in the Data Dictionary, you can use it across the boundaries of CGUs.

I

Implementation Describes how a specific internal behavior is implemented for a given platform (microprocessor type and compiler). An implementation mainly consists of a list of source files, object files, compiler attributes, and dependencies between the make and build processes.

Implementation data type (IDT) According to AUTOSAR, implementation data types are used to define types on the implementation level of abstraction. From the implementation point of view, this regards the storage and manipulation of digitally represented data. Accordingly, implementation data types have data semantics and do consider implementation details, such as the data type.

Implementation data types can be constrained to change the resolution of the digital representation or define a range that is to be considered. Typically, they correspond to typedef statements in C code and still abstract from platform specific details such as endianness.

See also application data type (ADT).

Implementation variable A variable in the generated production code to which a ReplaceableDataItem (RDI) object is mapped.

ImplementationPolicy A property of data element and Calprm elements that specifies the implementation strategy for the resulting variables with respect to consistency.

Implemented range The range of a variable defined by its scaling parameters. To avoid overflows, the implemented range must include the maximum and minimum values the variable can take in the simulation application and in the ECU.

Implicit communication A communication mode in Classic AUTOSAR. The data is exchanged at the start and end of the runnable that requires or provides the data.

Implicit object Any object created for the generated code by the TargetLink Code Generator (such as a variable, type, function, or file) that may not have been specified explicitly via a TargetLink block, a Stateflow object, or the TargetLink Data Dictionary. Implicit objects can be influenced via DD templates. For comparison, see explicit object.

Implicit property If the property of a DD object or of a model based object is not directly specified at the object, this property is created by the Code Generator and is based on internal templates or DD **Template** objects. These properties are called implicit properties. Also see implicit object and explicit object.

Included DD file A partial DD file that is inserted in the proper point of inclusion in the DD object tree.

Incremental code generation unit (CGU) Generic term for code generation units (CGUs) for which you can incrementally generate code. These are:

- Referenced models
- Subsystems configured for incremental code generation

Incremental CGUs can be nested in other model-based CGUs.

Indirect reuse The Code Generator adds pointers to the reuse structure which reference the indirectly reused instance-specific variables.

Indirect reuse has the following advantages to direct reuse:

- The combination of shared and instance-specific variable.
- The reuse of input/output variables of neighboring blocks.

Inline expansion The process of replacing a function call with the code of the function body. See also function inlining and compiler inlining.

Instance-specific variable A variable that is accessed by one reusable system instance. Typically, instance-specific variables are used for states and parameters whose value are different across instances.

Instruction set simulator (ISS) A simulation model of a microprocessor thatcan execute binary code compiled for the corresponding microprocessor. This allows the ISS to behave in the same way as the simulated microprocessor.

Integration model A model or TargetLink subsystem that contains modular units which it integrates to make a larger entity that provides its functionality.

Interface Describes the data elements, NvData, event messages, operations, or calibration parameters that are provided or required by a software component (SWC) via a port (AUTOSAR).

Internal behavior An element that represents the internal structure of an atomic software component (atomic SWC). It is characterized by the following entities and their interdependencies:

- Exclusive area
- Interrunnable variable (IRV)
- Per instance memory
- Per instance parameter
- Runnable
- RTE event
- Shared parameter

Interrunnable variable (IRV) Variable object for specifying communication between the runnables in one atomic software component (atomic SWC).

Interrupt service routine (ISR) function A function that implements an ISR and calls the step functions of the subsystems that are assigned by the user or by the TargetLink Code Generator during multirate code generation.

Intertask communication The flow of data between tasks and ISRs, tasks and tasks, and between ISRs and ISRs for multirate code generation.

Is service A property of an interface that indicates whether the interface is provided by a basic software service.

ISV Abbreviation for instance-specific variable.

L

Leaf bus element A leaf bus element is a subordinate bus element that is not a bus itself.

Leaf bus signal See also leaf bus element.

Leaf struct component A leaf struct component is a subordinate struct component that is not a struct itself.

Legacy function A function that contains a user-provided C function.

Library subsystem A subsystem that resides in a Simulink® library.

Local container A container that is owned by the tool that triggers the container exchange.

The tool that triggers the exchange transfers the files of a software component to this container when you export a software component. The external container is not involved.

Local MATLAB variable A variable that is generated when used on the left side of an assignment or in the interface of a MATLAB local function. TargetLink does not support different data types and sizes on local MATLAB variables.

Look-up function A function for a look-up table that returns a value from the look-up table (1-D or 2-D).

M

Macro A literal representing a C preprocessor definition. Macros are used to provide a fixed sequence of computing instructions as a single program statement. Before code compilation, the preprocessor replaces every occurrence of the macro by its definition, i.e., by the code that it stands for.

Macro AF The short form for an access function (AF) that is implemented as a function-like preprocessor macro.

MATLAB code elements MATLAB code elements include MATLAB local functions and local MATLAB variables. MATLAB code elements are not available in the Simulink Model Explorer or the Property Manager.

MATLAB local function A function that is scoped to a MATLAB main function and located at the same hierarchy level. MATLAB local functions are treated like MATLAB main functions and have the same properties as the MATLAB main function by default.

MATLAB main function The first function in a MATLAB function file.

Matrix AF An access function resulting from a DD AccessFunction object whose VariableKindSpec property is set to `APPLY_TO_MATRIX`.

Matrix signal Collective term for 2-D signals implemented as matrix variable in production code.

Matrix variable Collective term for 2-D arrays in production code that implement 2-D signals.

Measurement Viewing and analyzing the time traces of calibration parameters and measurement variables, for example, to observe the effects of ECU parameter changes.

Measurement and calibration system A tool that provides access to an ECU for measurement and calibration. It requires information on the calibration parameters and measurement variables with the ECU code.

Measurement variable Any variable type that can be measured but not calibrated. The term *measurement variable* is independent of a variable type's dimension.

Memory mapping The process of mapping variables and functions to different memory sections.

Memory section A memory location to which the linker can allocate variables and functions.

Message Browser A TargetLink component for handling fatal (F), error (E), warning (W), note (N), and advice (A) messages.

MetaData files Files that store metadata about code generation. The metadata of each code generation unit (CGU) is collected in a DD Subsystem object that is written to the file system as a partial DD file called `<CGU>_SubsystemObject.dd`.

Method Behavior subsystem An atomic subsystem used to generate code for a method implementation. From the TargetLink perspective, this is an Adaptive AUTOSAR Function that can take arguments.

It contains a Function block whose AUTOSAR mode is set to `Adaptive` and whose Role is set to `Method Behavior`.

Method Call subsystem An atomic subsystem that is used to generate a method call in the code of an Adaptive AUTOSAR Function. The subsystem contains a Function block whose AUTOSAR mode is set to `Adaptive` and whose Role is set to `Method Call`. The subsystem interface is used to generate the function interface while additional model elements that are contained in the subsystem are only for simulation purposes.

Microcontroller family (MCF) A group of microcontroller units with the same processor, but different peripherals.

Microcontroller unit (MCU) A combination of a specific processor with additional peripherals, e.g. RAM or AD converters. MCUs with the same processor, but different peripherals form a microcontroller family.

MIL simulation A simulation method in which the function model is computed (usually with double floating-point precision) on the host computer as an executable specification. The simulation results serve as a reference for SIL simulations and PIL simulations.

MISRA Organization that assists the automotive industry to produce safe and reliable software, e.g., by defining guidelines for the use of C code in automotive electronic control units or modeling guidelines.

Mode An operating state of an ECU, a single functional unit, etc..

Mode declaration group Contains the possible operating states, for example, of an ECU or a single functional unit.

Mode manager A piece of software that manages modes. A mode manager can be implemented as a software component (SWC) of the application layer.

Mode request type map An entity that defines a mapping between a mode declaration group and a type. This specifies that mode values are instantiated in the software component (SWC)'s code with the specified type.

Mode switch event An RTE event that specifies to start or continue the execution of a runnable as a result of a mode change.

Model Compare A dSPACE software tool that identifies and visualizes the differences in the contents of Simulink/TargetLink models (including Stateflow). It can also merge the models.

Model component A model-based code generation unit (CGU).

Model element A model in MATLAB/Simulink consists of model elements that are TargetLink blocks, Simulink blocks, and Stateflow objects, and signal lines connecting them.

Model port A port used to connect a behavior model in ConfigurationDesk. In TargetLink, multiple model ports of the same kind (data in or data out) can be grouped in a model port block.

Model port block A block in ConfigurationDesk that has one or more model ports. It is used to connect the behavior model in ConfigurationDesk.

Model port variable A DD Variable object that represents a model port of a behavior model in ConfigurationDesk.

Model-dependent code elements Code elements that (partially) result from specifications made in the model.

Model-independent code elements Code elements that can be generated from specifications made in the Data Dictionary alone.

Modular unit A submodel containing functionality that is reusable and can be integrated in different integration models. The production code for the modular unit can be generated separately.

Module A DD object that specifies code modules, header files, and other arbitrary files.

Module specification The reference of a DD Module object at a **Function Block** (📖 TargetLink Model Element Reference) block or DD object. The resulting code elements are generated into the module. See also production code and stub code.

ModuleOwnership A DD object that specifies an owner for a module (module owner) or module group, i.e. the owning code generation unit (CGU) that generates the production code for it or declares the module as external code that is not generated by TargetLink.

N

Nested bus A nested bus is a bus that is a subordinate bus element of another bus.

Nested struct A nested struct is a struct that is a subordinate struct component of another struct.

Non-scalar signal Collective term for vector and matrix signals.

Non-standard scaling A scaling whose LSB is different from `2^0 or whose Offset is not 0.`

Nv receiver port A require port in NvData communication as described by Classic AUTOSAR. In the Data Dictionary, nv receiver ports are represented as DD NvReceiverPort objects.

Nv sender port A provide port in NvData communication as described by Classic AUTOSAR. In the Data Dictionary, nv sender ports are represented as DD NvSenderPort objects.

Nv sender-receiver port A provide-require port in NvData communication as described by Classic AUTOSAR. In the Data Dictionary, nv sender-receiver ports are represented as DD NvSenderReceiverPort objects.

NvData Data that is exchanged between an atomic software component (atomic SWC) and the ECU's NVRAM.

NvData interface An interface used in NvData communication.

NVRAM Abbreviation of *non volatile random access memory*.

NVRAM manager A piece of software that manages an ECU's NVRAM. An NVRAM manager is part of the basic software.

O

Offline simulation application (OSA) An application that can be used for offline simulation in VEOS.

Online parameter modification The modification of parameters in the production code before or during a SIL simulation or PIL simulation.

Operation Defined in a client-server interface. A software component (SWC) can request an operation via a client port. A software component can provide an operation via a server port. Operations are implemented by server runnables.

Operation argument Specifies a C-function parameter that is passed and/or returned when an operation is called.

Operation call subsystem A collective term for synchronous operation call subsystem and asynchronous operation call subsystem.

Operation call with runnable implementation subsystem An atomic subsystem that contains a Function block whose AUTOSAR mode property is set to `Classic` and whose Role is set to `Operation call with runnable implementation`.

Operation invoked event An RTE event that specifies to start or continue the execution of a runnable as a result of a client call. A runnable that is related to an operation invoked event represents a server.

Operation result provider subsystem A subsystem used when modeling *asynchronous* client-server communication. It is used to generate the call of the `Rte_Result` API function and for simulation purposes.

See also asynchronous operation call subsystem.

Operation subsystem A collective term for operation call subsystem and operation result provider subsystem.

OSEK Implementation Language (OIL) A modeling language for describing the configuration of an OSEK application and operating system.

P

Package A structuring element for grouping elements of software components in any hierarchy. Using package information, software components can be spread across or combined from several software component description (SWC-D) files during AUTOSAR import/export scenarios.

Parent model A model containing references to one or more other models by means of the Simulink Model block.

Partial DD file A DD file that contains only a DD subtree. If it is included in a DD project file, it is called Included DD file. The partial DD file can be located on a central network server where all team members can share the same configuration data.

Per instance memory The definition of a data prototype that is instantiated for each atomic software component instance by the RTE. A data type instance can be accessed only by the corresponding instance of the atomic SWC.

Per instance parameter A parameter for measurement and calibration unique to the instance of a software component (SWC) that is instantiated multiple times.

Physical evaluation board (physical EVB) A board that is equipped with the same target processor as the ECU and that can be used for validation of the generated production code in PIL simulation mode.

PIL simulation A simulation method in which the TargetLink control algorithm (production code) is computed on a microcontroller target (physical or virtual).

Plain data type A data type that is not struct, union, or pointer.

Platform A specific target/compiler combination. For the configuration of platforms, refer to the **Code generation target settings** in the **TargetLink Main Dialog Block** block.

Pool area A DD object which is parented by the DD root object. It contains all data objects which can be referenced in TargetLink models and which are used for code generation. Pool data objects allow common data specifications to be reused across different blocks or models to easily keep consistency of common properties.

Port (AUTOSAR) A part of a software component (SWC) that is the interaction point between the component and other software components.

Port-defined argument values Argument values the RTE can implicitly pass to a server.

Preferences Editor A TargetLink tool that lets users view and modify all user-specific preference settings after installation has finished.

Production code The code generated from a code generation unit (CGU) that owns the module containing the code. See also stub code.

Project folder A folder in the file system that belongs to a TargetLink code generation project. It forms the root of different artifact locations that belong to this project.

Property Manager The TargetLink user interface for conveniently managing the properties of multiple model elements at the same time. It can consist of menus, context menus, and one or more panes for displaying property–related information.

Provide calprm port A provide port in parameter communication as described by Classic AUTOSAR. In the Data Dictionary, provide calprm ports are represented as DD **ProvideCalPrmPort** objects.

R

Read/write access function An access function (AF) that *encapsulates the instructions* for reading or writing a variable.

Real-time application An application that can be executed in real time on dSPACE real-time hardware such as SCALEXIO.

Receiver port A require port in sender-receiver communication as described by Classic AUTOSAR. In the Data Dictionary, receiver ports are represented as DD **ReceiverPort** objects.

ReplaceableDataItem (RDI) A ReplaceableDataItem (RDI) object is a DD object that describes an abstract interface's basic properties such as the data type, scaling and width. It can be referenced in TargetLink block dialogs and is generated as a global macro during code generation. The definition of the RDI macro can then be generated later, allowing flexible mapping to an implementation variable.

Require calprm port A require port in parameter communication as described by Classic AUTOSAR. In the Data Dictionary, require calprm ports are represented as DD **RequireCalPrmPort** objects.

RequirementInfo An object of a DD RequirementInfo object. It describes an item of requirement information and has the following properties: Description, Document, Location, UserTag, ReferencedInCode, SimulinkStateflowPath.

Restart function A production code function that initializes the global variables that have an entry in the **RestartfunctionName** field of their variable class.

Reusable function definition The function definition that is to be reused in the generated code. It is the code counterpart to the reusable system definition in the model.

Reusable function instance An instance of a reusable function definition. It is the code counterpart to the reusable system instance in the model.

Reusable model part Part of the model that can become a reusable system definition. Refer to Basics on Function Reuse (TargetLink Customization and Optimization Guide).

Reusable system definition A model part to which the function reuse is applied.

Reusable system instance An instance of a reusable system definition.

Root bus A root bus is a bus that is not a subordinate part of another bus.

Root function A function that represents the starting point of the TargetLink-generated code. It is called from the environment in which the TargetLink-generated code is embedded.

Root model The topmost parent model in the system hierarchy.

Root module The module that contains all the code elements that belong to the production code of a code generation unit (CGU) and do not have their own module specification.

Root step function A step function that is called only from outside the production code. It can also represent a non-TargetLink subsystem within a TargetLink subsystem.

Root struct A root struct is a struct that is not a subordinate part of another struct.

Root style sheet A root style sheet is used to organize several style sheets defining code formatting.

RTE event The abbreviation of run-time environment event.

Runnable A part of an atomic SWC. With regard to code execution, a runnable is the smallest unit that can be scheduled and executed. Each runnable is implemented by one C function.

Runnable execution constraint Constraints that specify runnables that are allowed or not allowed to be started or stopped before a runnable.

Runnable subsystem An atomic subsystem that contains a Function block whose AUTOSAR mode property is set to `Classic` and whose Role is set to `Runnable`.

Run-time environment (RTE) A generated software layer that connects the application layer to the basic software. It also interconnects the different SWCs of the application layer. There is one RTE per ECU.

Run-time environment event A part of an internal behavior. It defines the situations and conditions for starting or continuing the execution of a specific runnable.

S

Scaling A parameter that specifies the fixed-point range and resolution of a variable. It consists of the data type, least significant bit (LSB) and offset.

Sender port A provide port in sender-receiver communication as described by Classic AUTOSAR. In the Data Dictionary, sender ports are represented as DD SenderPort objects.

Sender-receiver interface An interface that describes the data elements and event messages that are provided or required by a software component (SWC) via a port (AUTOSAR).

Sender-receiver port A provide-require port in sender-receiver communication as described by Classic AUTOSAR. In the Data Dictionary, sender-receiver ports are represented as DD SenderReceiverPort objects.

Server port A provide port in client-server communication as described by Classic AUTOSAR. In the Data Dictionary, server ports are represented as DD ServerPort objects.

Server runnable A runnable that provides an operation via a server port. Server runnables are triggered by operation invoked events.

Shared parameter A parameter for measurement and calibration that is used by several instances of the same software component (SWC).

Shared variable A variable that is accessed by several reusable system instances. Typically, shared variables are used for parameters whose values are the same across instances. They increase code efficiency.

SIC runnable function A void (void) function that is called in a task. Generated into the Simulink implementation container (SIC) to call the root function that is generated by TargetLink from a TargetLink subsystem. In ConfigurationDesk, this function is called *runnable function*.

SIL simulation A simulation method in which the control algorithm's generated production code is computed on the host computer in place of the corresponding model.

Simple TargetLink model A simple TargetLink model contains at least one TargetLink Subsystem block and exactly one MIL Handler block.

Simplified initialization mode The initialization mode used when the Simulink diagnostics parameter Underspecified initialization detection is set to `Simplified`.

See also classic initialization mode.

Simulation application An application that represents a graphical model specification (implemented control algorithm) and simulates its behavior in an offline Simulink environment.

Simulation code Code that is required only for simulation purposes. Does not belong to the production code.

Simulation S-function An S-function that calls either the root step functions created for a TargetLink subsystem, or a user-specified step function (only possible in test mode via API).

Simulink data store Generic term for a memory region in MATLAB/Simulink that is defined by one of the following:

- A Simulink.Signal object
- A Simulink Data Store Memory block

Simulink function call The location in the model where a Simulink function is called. This can be:

- A Function Caller block
- The action language of a Stateflow Chart
- The MATLAB code of a MATLAB function

Simulink function definition The location in the model where a Simulink function is defined. This can be one of the following:

- Simulink Function subsystem
- Exported Stateflow graphical function
- Exported Stateflow truthtable function
- Exported Stateflow MATLAB function

Simulink function ports The ports that can be used in a Simulink Function subsystem. These can be the following:

- TargetLink ArgIn and ArgOut blocks

  These ports are specific for each Simulink function call.
- TargetLink InPort/OutPort and Bus Inport/Bus Outport blocks

  These ports are the same for all Simulink function calls.

Simulink Function subsystem A subsystem that contains a Trigger block whose Trigger Type is `function-call` and whose Treat as Simulink Function checkbox is selected.

Simulink implementation container (SIC) A file that contains all the files required to import production code generated by TargetLink into ConfigurationDesk as a behavior model with model ports.

Slice A section of a vector or matrix signal, whose elements have the same properties. If all the elements of the vector/matrix have the same properties, the whole vector/matrix forms a slice.

Software component (SWC) The generic term for atomic software component (atomic SWC), compositions, and special software components, such as calprm software components. A software component logically groups and encapsulates single functionalities. Software components communicate with each other via ports.

Software component description (SWC-D) An XML file that describes software components according to AUTOSAR.

Stateflow action language The formal language used to describe transition actions in Stateflow.

Struct A struct (short form for structure) consists of subordinate struct components. A struct component can be a struct itself.

Struct component A struct component is a part of a struct and can be a struct itself.

Structure A structure (long form for struct) consists of subordinate struct components. A struct component can be a struct itself.

Stub code Code that is required to build the simulation application but that belongs to another code generation unit (CGU) than the one used to generate production code.

Subsystem area A DD object which is parented by the DD root object. This object consists of an arbitrary number of **Subsystem** objects, each of which is the result of code generation for a specific code generation unit (CGU). The **Subsystem** objects contain detailed information on the generated code, including C modules, functions, etc. The data in this area is either automatically generated or imported from ASAM MCD-2 MC, and must not be modified manually.

Supported Simulink block A TargetLink-compliant block from the Simulink library that can be directly used in the model/subsystem for which the Code Generator generates production code.

SWC container A container for files of one SWC.

Synchronous operation call subsystem A subsystem used when modeling *synchronous* client-server communication. It is used to generate the call of the `Rte_Call` API function and for simulation purposes.

T

Table function A function that returns table output values calculated from the table inputs.

Target config file An XML file named `TargetConfig.xml`. It contains information on the basic data types of the target/compiler combination such as the byte order, alignment, etc.

Target Optimization Module (TOM) A TargetLink software module for optimizing production code generation for a specific microcontroller/compiler combination.

Target Simulation Module (TSM) A TargetLink software module that provides support for a number of evaluation board/compiler combinations. It is used to test the generated code on a target processor. The TSM is licensed separately.

TargetLink AUTOSAR Migration Tool A software tool that converts classic, non-AUTOSAR TargetLink models to AUTOSAR models at a click.

TargetLink AUTOSAR Module A TargetLink software module that provides extensive support for modeling, simulating, and generating code for AUTOSAR software components.

TargetLink Base Suite The base component of the TargetLink software including the ANSI C Code Generator and the Data Dictionary Manager.

TargetLink base type One of the types used by TargetLink instead of pure C types in the generated code and the delivered libraries. This makes the code platform independent.

TargetLink Blockset A set of blocks in TargetLink that allow production code to be generated from a model in MATLAB/Simulink.

TargetLink Data Dictionary The central data container thats holds all relevant information about an ECU application, for example, for code generation.

TargetLink simulation block A block that processes signals during simulation. In most cases, it is a block from standard Simulink libraries but carries additional information required for production code generation.

TargetLink subsystem A subsystem from the TargetLink block library that defines a section of the Simulink model for which code must be generated by TargetLink.

Task A code section whose execution is managed by the real-time operating system. Tasks can be triggered periodically or based on events. Each task can call one or more SIC runnable functions.

Task function A function that implements a task and calls the functions of the subsystems which are assigned to the task by the user or via the TargetLink Code Generator during multirate code generation.

Term function A function that contains the code to be executed when the simulation finishes or the ECU application terminates.

Terminate function A runnable that finalizes a SWC, for example, by calling code that has to run before the application shuts down.

Timing event An RTE event that specifies to start or continue the execution of a runnable at constant time intervals.

tllib A TargetLink block library that is the source for creating TargetLink models graphically. Refer to How to Open the TargetLink Block Library (TargetLink Orientation and Overview Guide).

Transformer The Classic AUTOSAR entity used to perform a data transformation.

TransformerError The parameter passed by the run-time environment (RTE) if an error occurred in a data transformation. The `Std_TransformerError` is a struct whose components are the transformer class and the error code. If the error is a hard error, a special runnable is triggered via the TransformerHardErrorEvent to react to the error.

In AUTOSAR releases prior to R19-11 this struct was named `Rte_TransformerError`.

TransformerHardErrorEvent The RTE event that triggers the runnable to be used for responding to a hard TransformerError in a data transformation for client-server communication.

Type prefix A string written in front of the variable type of a variable definition/declaration, such as `MyTypePrefix Int16 MyVar`.

U

Unicode The most common standard for extended character sets is the Unicode standard. There are different schemes to encode Unicode in byte format, e.g., UTF-8 or UTF-16. All of these encodings support all Unicode characters. Scheme conversion is possible without losses. The only difference between these encoding schemes is the memory that is required to represent Unicode characters.

User data type (UDT) A data type defined by the user. It is placed in the Data Dictionary and can have associated constraints.

Utility blocks One of the categories of TargetLink blocks. The blocks in the category keep TargetLink-specific data, provide user interfaces, and control the simulation mode and code generation.

V

Validation Summary Shows unresolved model element data validation errors from all model element variables of the **Property View**. It lets you search, filter, and group validation errors.

Value copy AF An access function (AF) resulting from DD AccessFunction objects whose AccessFunctionKind property is set to `READ_VALUE_COPY` or `WRITE_VALUE_COPY`.

Variable access function An access function (AF) that *encapsulates the access* to a variable for reading or writing.

Variable class A set of properties that define the role and appearance of a variable in the generated production code, e.g. CAL for global calibratable variables.

VariantConfig A DD object in the Config area that defines the code variants and data variants to be used for simulation and code generation.

VariantItem A DD object in the DD Config area used to variant individual properties of DD Variable and ExchangeableWidth objects. Each variant of a property is associated with one variant item.

V-ECU implementation container (VECU) A file that consists of all the files required to build an offline simulation application (OSA) to use for simulation with VEOS.

V-ECU Manager A component of TargetLink that allows you to configure and generate a V-ECU implementation.

Vendor mode The operation mode of RTE generators that allows the generation of RTE code which contains vendor-specific adaptations, e.g., to reduce resource consumption. To be linkable to an RTE, the object code of an SWC must have been compiled against an application header that matches the RTE code generated by the specific RTE generator. This is the case because the data structures and types can be implementation-specific.

See also compatibility mode.

VEOS A dSPACE software platform for the C-code-based simulation of virtual ECUs and environment models on a PC.

Virtual ECU (V-ECU) Software that emulates a real ECU in a simulation scenario. The virtual ECU comprises components from the application and the basic software, and provides functionalities comparable to those of a real ECU.

Virtual ECU testing Offline and real-time simulation using virtual ECUs.

Virtual evaluation board (virtual EVB) A combination of an instruction set simulator (ISS) and a simulated periphery. This combination can be used for validation of generated production code in PIL simulation mode.

W

Worst-case range limits A range specified by calculating the minimum and maximum values which a block's output or state variable can take on with respect to the range of the inputs or the user-specified constrained range limits.

Symbols

A

B

C

D

E

F

G

I

L

M

N

O

P

R

S

U

V