

dSPACE Release

# 新機能と移行手順

Release 2023-B – 2023 年 11 月

## dSPACE へのお問い合わせ

E-mail: [info@dspace.jp](mailto:info@dspace.jp)  
Web: <https://www.dspace.jp>

## dSPACE サポートへのお問い合わせ

dSPACE 製品の使用時に問題が発生した場合は、dSPACE Japan 株式会社まで、WEB ページ、Email または Fax にてお問い合わせください。

- dSPACE 現地法人および販売代理店: <http://www.dspace.jp/go/locations>
- 現地にオフィスがない場合は、dSPACE GmbH(ドイツ、パーダーボルン)にお問い合わせください。電話:+49 5251 1638-941 E-mail:[support@dspace.de](mailto:support@dspace.de)

サポートお問い合わせフォームもご利用いただけます。

[http://www.dspace.jp/go/jpn\\_supportrequestmydSPACE](http://www.dspace.jp/go/jpn_supportrequestmydSPACE) にログインいただいた状態ですと、自動的に認証されますため、ご連絡先を手動入力していただく必要はありません。

可能な場合、サポートお問い合わせの際にはハードウェアのシリアル番号、dSPACE ライセンス ID、または CmContainer のシリアル番号を常にお知らせください。

## ソフトウェアのアップデートとパッチ

既存の dSPACE インストールに対して、最新のパッチをダウンロードしてインストールすることを強くお勧めします。ソフトウェアのアップデートとパッチ自体、および dSPACE ソフトウェアのアップデートやパッチが利用可能になったときに自動通知を受け取る方法などの詳細については、<http://www.dspace.jp/go/patches> を参照してください。

## 重要なお知らせ

本書には、著作権法により保護された情報が含まれています。すべての権利は留保されています。本書は、すべての商標表示をすべての印刷コピーに保持するという条件で、個人または内部での使用を目的として印刷することができます。それ以外のすべての場合において、dSPACE GmbH の書面による事前の許可なく、本書のすべてもしくは一部を、コピー、複製、翻訳、または電子的媒体もしくは機械可読形式に変換することを禁じます。

Copyright 2000 - 2023  
dSPACE GmbH  
Rathenaustraße 26  
33102 Paderborn  
Germany

本出版物と内容は、予告なしで変更されることがあります。

AUTERA、ConfigurationDesk、ControlDesk、MicroAutoBox、MicroLabBox、SCALEXIO、SIMPHERA、SYNECT、SystemDesk、TargetLink、および VEOS は、米国、その他の国々、またはその両方における dSPACE GmbH の登録商標です。その他のブランド名または製品名は、その企業または組織の商標または登録商標です。

# 目次

本書について	11
dSPACE Release 2023-B の概要	13
一般的な機能拡張および変更	13
廃止	15
製品バージョンの概要	17
製品の主な新機能	20
新しい使用例: データプレイ	26
以前のリリースからの移行について	29
dSPACE Release 2023-B への移行	29
Python 3.9 から Python 3.11 への Python スクリプトの移行	31
Python 3.11 の主な変更点	31
dSPACE ソフトウェアでの Python 3.11 の取り扱いの主な変更点	32
Python インストールの使用に関する一般情報	33
技術的変更	35
移行に関する製品固有の情報	35
AutomationDesk	39
AutomationDesk 2023-B の新機能	39
AutomationDesk 2023-B への移行	40
Automotive Simulation Models (ASM)	41
すべての ASM 製品	42
すべての ASM デモモデルの変更	42
すべての ASM ブロックセットの移行	42
ASM Battery	43
ASM Battery Blockset 2023-B の新機能	43
ASM Battery デモモデルの変更	43
ASM Battery Blockset 2023-B への移行	44
ASM Diesel Engine	45
ASM Diesel Exhaust Blockset 2023-B への移行	45

ASM Drivetrain Basic.....	46
ASM Drivetrain Basic Blockset 2023-B の新機能.....	46
ASM Drivetrain Basic Blockset 2023-B への移行.....	46
ASM Electric Components.....	47
ASM Electric Components Blockset 2023-B の新機能.....	47
ASM Electric Components デモモデルの変更.....	47
ASM Electric Components Blockset 2023-B への移行.....	47
ASM Fuel Cell.....	48
ASM Fuel Cell Blockset 2023-B の新機能.....	48
ASM Fuel Cell デモモデルの変更.....	49
ASM Fuel Cell Blockset 2023-B への移行.....	49
ASM Gasoline Engine.....	50
ASM Gasoline Engine Blockset 2023-B への移行.....	50
ASM Traffic.....	51
ASM Traffic デモモデルの変更.....	51
ASM Trailer.....	52
ASM Trailer Blockset 2023-B の新機能.....	52
ASM Trailer デモモデルの変更.....	53
ASM Truck.....	54
ASM Truck Blockset 2023-B の新機能.....	54
ASM Truck デモモデルの変更.....	54
ASM Vehicle Dynamics.....	56
ASM Vehicle Dynamics Blockset 2023-B の新機能.....	56
ASM Vehicle Dynamics デモモデルの変更.....	57
ASM Vehicle Dynamics Blockset 2023-B への移行.....	57
<b>Bus Manager (スタンドアロン).....</b>	<b>59</b>
Bus Manager (スタンドアロン) 2023-B の新機能.....	59
Bus Manager (スタンドアロン) 2023-B への移行.....	62
以前のバージョンで作成されたプロジェクトとアプリケーションの移行.....	62
Bus Manager (スタンドアロン) の廃止.....	65
<b>ConfigurationDesk.....</b>	<b>67</b>
ConfigurationDesk 2023-B の新しい一般機能.....	67
I/O 機能に関する新機能とハードウェアサポート.....	69
ConfigurationDesk の Bus Manager の新機能.....	74
サポートされるコンテナファイルバージョン.....	76
ConfigurationDesk 2023-B への移行.....	79

以前のバージョンで作成されたプロジェクトとアプリケーションの移行.....	83
ConfigurationDesk の廃止.....	86
<b>ControlDesk</b> .....	<b>87</b>
ControlDesk 2023-B の新機能.....	88
プラットフォーム管理およびプラットフォーム／デバイスの新機能 (ControlDesk 2023-B).....	88
新しい変数管理機能 (ControlDesk 2023-B).....	89
新しい計器機能 (ControlDesk 2023-B).....	90
新しい Bus Navigator 機能 (ControlDesk 2023-B).....	93
新しい電氣的欠陥シミュレーション機能 (ControlDesk 2023-B).....	101
新しいユーザインターフェース処理機能 (ControlDesk 2023-B).....	101
ControlDesk 2023-B への移行.....	104
ControlDesk 2023-B への移行.....	104
<b>DCI Configuration Tool</b> .....	<b>107</b>
DCI Configuration Tool 2023-B の新機能.....	107
<b>dSPACE ECU Flash Programming Tool</b> .....	<b>109</b>
dSPACE ECU Flash Programming Tool 2023-B の新機能.....	109
<b>dSPACE FlexRay Configuration Package</b> .....	<b>111</b>
dSPACE FlexRay Configuration Package 2023-B の新機能.....	111
<b>dSPACE Installation Manager</b> .....	<b>113</b>
dSPACE Installation Manager 23.2 の新機能.....	113
dSPACE Installation Manager 23.2 への移行.....	114
<b>dSPACE Python Extensions</b> .....	<b>117</b>
dSPACE Python Extensions 2023-B の新機能.....	117
<b>dSPACE XIL API .NET</b> .....	<b>119</b>
dSPACE XIL API .NET 2023-B の新機能.....	119
dSPACE XIL API .NET 2023-B への移行.....	120
<b>ECU Interface Manager</b> .....	<b>123</b>
ECU Interface Manager 2023-B の新機能.....	123

ECU Interface Manager 2023-B の互換性.....	124
ECU Interface Manager 2023-B への移行.....	124
<b>Firmware Manager</b>	<b>127</b>
Firmware Manager 23.2 の新機能.....	127
<b>FPGA Programming Blockset</b>	<b>129</b>
FPGA Programming Blockset 2023-B の新機能.....	129
FPGA Programming Blockset 2023-B への移行.....	131
<b>MicroAutoBox III のファームウェア</b>	<b>135</b>
MicroAutoBox III Firmware 23.2 の新機能.....	135
<b>Model Compare</b>	<b>137</b>
Model Compare 2023-B の新機能.....	137
Model Compare 2023-B への移行.....	138
<b>Model および Sensor Interface Blockset</b>	<b>139</b>
Model and Sensor Interface Blockset 2023-B の新機能.....	139
Model and Sensor Interface Blockset 2023-B への移行.....	139
<b>ModelDesk</b>	<b>141</b>
ModelDesk 2023-B の新機能.....	141
ModelDesk 2023-B への移行.....	142
<b>Model Interface Package for Simulink</b>	<b>143</b>
Model Interface Package for Simulink 2023-B の新機能.....	143
Model Interface Package for Simulink 2023-B への移行.....	144
<b>MotionDesk</b>	<b>145</b>
MotionDesk 2023-B の新機能.....	145
MotionDesk 2023-B への移行.....	145
<b>Real-Time Testing</b>	<b>147</b>
Real-Time Testing 2023-B の新機能.....	147
Real-Time Testing 2023-B への移行.....	147

RTI/RTI-MP および RLib	149
RTI/RTI-MP および RLib の新機能	149
RTI/RTI-MP および RLib の移行上の注意点	149
RTI Bypass Blockset	151
RTI Bypass Blockset 2023-B への移行	151
RTI CAN MultiMessage Blockset	153
RTI CAN MultiMessage Blockset 2023-B の新機能	153
RTI CAN MultiMessage Blockset 2023-B への移行	153
RTI LIN MultiMessage Blockset	155
RTI LIN MultiMessage Blockset 2023-B の新機能	155
RTI LIN MultiMessage Blockset 2023-B への移行	155
RTI Synchronized Time Base Manager Blockset	157
RTI Synchronized Time Base Manager Blockset 2023-B の新機能	157
SCALEXIO Firmware	159
SCALEXIO Firmware 23.2 の新機能	159
SCALEXIO Firmware 23.2 への移行	160
SCALEXIO ファームウェアの廃止	161
Signal List Editor	163
Signal List Editor 2023-B (23.1)の機能	163
SYNECT	165
SYNECT 2023-B への移行	166
SYNECT 2023-A からの移行	166
データベースの移行	166
SYNECT 2023-A から SYNECT 2023-B へのデータモデルの変更	167
SystemDesk	169
SystemDesk 2023-B の新機能	170
新しい一般機能	170

新しい V-ECU 生成機能.....	171
A2L ファイル処理の改善.....	171
SystemDesk 2023-B への移行.....	174
dSPACE Release 2023-B の廃止.....	174
SystemDesk 2023-B への移行.....	175
API スクリプトの移行.....	176

## TargetLink 181

TargetLink 2023-B の新機能.....	182
Simulink または Stateflow でのモデリング.....	182
構造体配列のサポートの改善.....	183
動的エレメントの指定.....	183
連立一次方程式の解法.....	184
Ensure Outport is Virtual のサポート.....	185
Prelookup ブロックの可変ベクトル幅のサポート.....	185
Stateflow オブジェクトの使用.....	186
機能インターフェースに関する改善.....	186
Adaptive AUTOSAR.....	187
サポートされている Adaptive AUTOSAR リリース.....	187
Adaptive AUTOSAR のフレームモデル生成.....	187
C++ 列挙クラスのデータタイプ.....	188
新しい Adaptive AUTOSAR モデリング機能.....	188
AUTOSAR バージョン R22-11 に準拠した変更.....	188
C++ の名前空間.....	189
Classic AUTOSAR.....	189
サポートされる Classic AUTOSAR リリース.....	189
構造体配列のサポートの改善 (Classic AUTOSAR).....	190
AUTOSAR のインポートおよびエクスポートの改良.....	190
コード生成のコア機能.....	191
コード効率性の向上.....	191
MIL (Model-in-the-Loop) シミュレーション.....	191
シミュレーション結果の分析.....	191
ターゲットシミュレーション (PIL).....	192
TargetLink 2023-B でのターゲットシミュレーションモジュールの変更.....	192
ユーザビリティ.....	192
ユーザビリティに関する改良点.....	192
コードジェネレータオプション.....	195
新しいコードジェネレータオプション.....	195



API 関数とフックスクリプト.....	196
新しい API 関数.....	196
新しいフックスクリプト.....	197
その他.....	197
FMI 3.0 のサポート.....	198
TargetLink のデモと例.....	198
TargetLink 2023-B および TargetLink Data Dictionary 2023-B への移行.....	200
移行に関する一般的な情報.....	200
モデル、ライブラリ、Data Dictionary のアップグレード.....	200
TargetLink バージョン間の移行に関する基礎.....	201
インクルード DD ファイルのある Data Dictionary をアップグレードおよび保存する方法.....	204
API を使用してライブラリとモデルを手動でアップグレードする方法.....	204
Data Dictionary の CodeDecorationSets への移行.....	205
正確に表現可能な浮動小数点数の丸め動作の変更 (Microsoft).....	208
正確に表現可能な浮動小数点数の丸め動作の変更 (Microsoft).....	208
TargetLink 2022-B から 2023-B への移行.....	209
コードジェネレータオプション.....	209
コードジェネレータオプションに関する移行上の注意点.....	209
API 関数とフックスクリプト.....	211
API 関数の変更.....	212
フックスクリプトの変更.....	215
移行に関するその他の注意点.....	217
移行に関するその他の注意点.....	217
MATLAB のサポート.....	218
MATLAB R2023a および R2023b はサポートされない.....	218
TargetLink 2022-B と TargetLink 2023-B 間でのコードの変更.....	218
アトミックサブシステムおよび機能インターフェースの主な(または最新の)変更点.....	218
コードの効率性.....	223
Adaptive-AUTOSAR 関連と CPP 関連.....	231
構造体関連と構造体配列関連.....	233
Classic AUTOSAR 関連.....	237
その他のコード変更.....	238
TargetLink 2023-B での廃止項目.....	247
廃止された TargetLink の機能.....	247
廃止された制限事項.....	248
廃止されたコードジェネレータオプション.....	249










今後の TargetLink バージョンでの変更予定.....	250
廃止予定の機能.....	250
廃止予定の API 関数.....	251
廃止されたコードジェネレータオプション.....	251
<b>VEOS</b> .....	<b>253</b>
VEOS 2023-B の新機能.....	253
VEOS 2023-B の互換性.....	255
VEOS 2023-B への移行.....	259
VEOS 2024-A での廃止予定項目.....	260
<b>互換性情報</b> .....	<b>261</b>
サポートしている MATLAB リリース.....	261
オペレーティングシステム.....	263
仮想マシン (VM) での dSPACE ソフトウェアの使用.....	265
dSPACE ソフトウェアのランタイム互換性.....	270
Windows 機能の使用に関する制限事項.....	270
Linux 機能の使用に関する制限事項.....	272
<b>索引</b> .....	<b>273</b>

# 本書について

**内容** 本書では、Release 2023-B に含まれるすべての dSPACE ソフトウェア製品の new 機能について説明します。以前の dSPACE リリースからの変更がない、または変更が少ないソフトウェア製品についても概要を示します。また、以前の dSPACE リリース、特に以前の製品バージョンからの移行手順についても、必要に応じて説明します。

**印刷版ドキュメント** ご要望に応じて、本書の印刷版をご利用いただけます。次のリンクから、無償にてご注文いただけます。<http://www.dspace.jp/go/requestreleasematerial>。

**記号** dSPACE のユーザマニュアルでは以下の記号を使用します。

記号	説明
 危険	回避しないと死亡または重度の人身傷害につながる危険な状況を示します。
 警告	回避しないと死亡または重度の人身傷害につながる可能性がある危険な状況を示します。
 注意	回避しないと小規模または軽度の人身傷害につながる可能性がある危険な状況を示します。
 注意	回避しないと物的損害につながる可能性がある危険を示します。
 注記	誤動作を回避するために考慮すべき重要な情報を示します。
 ヒント	作業を円滑に進めるのに役立つヒントを示します。
 	リンク先が用語解説の定義を参照していることを示します。用語解説は特に記載がないかぎり本書の末尾に記載されています。
	リンク先が別のドキュメントを参照する場合にドキュメントタイトルの前に付記されます。

---

## 表記規則

dSPACE のユーザマニュアルでは以下の表記規則を使用します。

**%name%** パーセント記号で囲まれた名前は、ファイルとパス名の環境変数を表します。

**<>** 山形括弧で囲まれた表記は、任意のファイル名やパス名などを表すワイルドカード文字またはプレースホルダを示します。

---

## 特別な Windows フォルダ

Windows ベースのソフトウェア製品は、次の特別なフォルダを使用します。

**共通プログラムデータフォルダ** アプリケーション固有のプログラムデータ用の標準フォルダで、すべてのユーザが使用します。

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

または

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

**ドキュメントフォルダ** 現在のユーザが使用する、アプリケーション固有の標準フォルダ。

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

**ローカルプログラムデータフォルダ** アプリケーション固有のプログラムデータ用の標準フォルダで、現在の非ローミングユーザが使用します。

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


---

## dSPACE Help および PDF ファイルの利用

Windows ベースの dSPACE ソフトウェアをインストールして復号化すると、インストールした製品に関するドキュメントが dSPACE Help または PDF ファイルで参照できるようになります。

**dSPACE Help (ローカル)** ローカルにインストールされた dSPACE Help を開くことができます。

- Windows の[スタート]メニューを使用すると、dSPACE Help のホームページが開きます。
- F1 キーによる状況依存ヘルプを使用すると、個別の内容に関するページが開きます。

**PDF ファイル** dSPACE Help で  アイコンを使用すると、PDF ファイルにアクセスすることができます。PDF の最初のページが開きます。

**dSPACE Help (ウェブ)** ソフトウェアインストールに関係なく、dSPACE Help のウェブ版の dSPACE Help には <http://www.dspace.jp/go/help> からアクセスすることができます。

ウェブ版にアクセスするには、*mydSPACE* アカウントが必要です。mydSPACE 登録プロセスの詳細については、[www.dspace.com/faq?097](http://www.dspace.com/faq?097) を参照してください。

# dSPACE Release 2023-B の概要

**概要** Release 2023-B の主な新機能について説明します。また、変更のない製品に関する情報についても紹介します。

## 次のステップ

### 本章の内容

一般的な機能拡張および変更.....	13
廃止.....	15
製品バージョンの概要.....	17
製品の主な新機能.....	20
新しい使用例: データリプレイ.....	26

## 一般的な機能拡張および変更

**概要** 複数の dSPACE 製品に関する新機能と変更を下記に示します。

### サードパーティ製ソフトウェアの使用に関する法律上の注意

一部の dSPACE ソフトウェア製品ではサードパーティ製ソフトウェアを使用または含んでいます。サードパーティ製コンポーネントは、追加の利用規約または dSPACE ソフトウェアに適用されるものとは異なる利用規約が適用される場合があります。サードパーティ製コンポーネントの著作権表示とライセンス条件を順守する必要があります。ユーザがサードパーティ製コンポーネントの使用を明示し、利用規約を順守する必要がある場合は、利用規約に関する情報が各 dSPACE 製品の別個の OSSAcknowledgements (OSS に対する謝辞) ファイルに記載されています。

OSSAcknowledgements ファイルは、次の方法で見つけることができます。

- 両方の dSPACE DVD のルートディレクトリ、およびダウンロードされたインストールेशनファイル(マウントされた ISO イメージまたは解凍された ZIP ファイル)の

ルートディレクトリから Legal フォルダにアクセスすると、OSSAcknowledgements ファイルが見つかります。

- dSPACE ソフトウェアのインストール後は、<dSPACE Setup のメイン インストールパス>\InstallationSet のサブフォルダ>\Legal フォルダに OSSAcknowledgements ファイルがあります。

## Python ディストリビューション

Python 3.9 のサポートは dSPACE Release 2023-A で廃止され、Python 3.11 に置き換えられます。詳細については、「[Python 3.9 から Python 3.11 への Python スクリプトの移行](#)」(31 ページ)を参照してください。

## 新しい dSPACE ハードウェアのサポート

dSPACE Release 2023-B は、次の新しい dSPACE ハードウェアを導入します。

### ▪ MicroLabBox II

新しい MicroLabBox II は MicroLabBox の後継製品で、すぐに使えるラピッドコントロールプロトタイピングシステムであり、ラボ環境向けの HIL (Hardware-in-the-Loop) テストシステムとして以下のような特長があります。

- リアルタイムプロセッサの演算能力が向上し、プロセッサには 4 つの CPU コアがあります。
- 10 GBit Ethernet ポートや CAN FD SIC など、通信インターフェースの向上。
- 実績のあるパネルと接続コンセプト: アナログ信号用の BNC コネクタを備えたトップパネルタイプ、または D-Sub コネクタを備えたフロントパネルタイプをご注文いただけます。

New: フロントパネルタイプは 3 個までスタック可能です。

- リアルタイムアプリケーションを実装するための ConfigurationDesk によるサポート。

サポートされる I/O 機能の詳細については、「[Overview of MicroLabBox II I/O Functionality](#)」([『MicroLabBox II Hardware Installation and Configuration』](#))を参照してください。

MicroLabBox II の基本的な操作手順については、「[Get the MicroLabBox II Started](#)」([『MicroLabBox II - Getting Started』](#))を参照してください。

MicroLabBox および RTI ユーザについては、「[Migrating MicroLabBox RTI Models](#)」([『MicroLabBox II - Getting Started』](#))以下を参照してください。

## RCP and HIL Software: MATLAB MEX ファイルのビルド用 C/C++コンパイラ

RCP and HIL Software (RTI CAN MultiMessage Blockset, RTI LIN MultiMessage Blockset, または Automotive Simulation Models など)で、MATLAB R2023b 用の MATLAB MEX ファイルをビルドする次の C/C++コンパイラがサポートされるようになりました。

- MinGW (GNU Compiler Collection (GCC 8.1.0))
- Microsoft Visual Studio 2022 Professional

## ライセンスの概要を PDF ファイルで提供

2023 年 7 月より、dSPACE はライセンス概要(購入製品のチケット ID を含む)を PDF ファイルで提供し、Excel シートでの提供は行いません。

## ランタイムバージョンライセンスの新しい適合性チェック

リアルタイムアプリケーションには、ランタイムバージョンライセンスを必要とするモデルが含まれていることが多く、たとえば、Automotive Simulation Model (ASM) の

ランタイムバージョンライセンス(ASM\_...\_RTV ライセンス)などが挙げられます。これらのライセンスは、SCALEXIO などの dSPACE リアルタイムハードウェア上でリアルタイムアプリケーションをダウンロードし、実行するために必要です。

dSPACE Release 2023-A では、リアルタイムアプリケーションのダウンロード時に、dSPACE ソフトウェアがそのようなランタイムバージョンライセンスの適合性チェックを実行します。このチェックにより、dSPACE エンドユーザライセンス契約書(EULA)の契約内容を確認し、履行することができます。

#### 注記

- dSPACE Release 2023-A では、Automotive Simulation Model(ASM)の必要なランタイムバージョンライセンス(たとえば、ASM Battery Library(ランタイムバージョン)ライセンス)についてのみ適合性チェックを導入しています。他の製品のランタイムバージョンライセンスのチェックは、後の dSPACE Release に含まれる予定です。
- VEOS プラットフォーム上でオフラインシミュレーションアプリケーション(OSA)を実行する場合は、Automotive Simulation Model(ASM)のランタイムバージョンライセンスも必要となります。OSAに必要なライセンスの利用可能性チェックは、dSPACE Release 2017-B から既実装されています。このチェックは、シミュレーションの開始時に VEOS によって実行されます。必要なライセンスのうち少なくとも 1 つが利用できない場合、OSA を実行することはできません。

詳細については、「ランタイムバージョンライセンスの適合性チェック」(『CodeMeter ライセンステクノロジーの使用』)を参照してください。

## 廃止

### 概要

次のソフトウェアおよびハードウェアの廃止は、現在のリリースに関連しているか、または将来のリリースで計画されています。

サポート終了のお知らせについては、<http://www.dspace.jp/go/discontinuation> を参照してください。

### ソフトウェアサポートの廃止

**SCALEXIO Firmware** dSPACE Release 2023-B 以降、64 ビット Linux ベースのリアルタイムオペレーティングシステム(RTOS)が、SCALEXIO システムおよび DS6001 Processor Board のデフォルト RTOS です。32 ビット Linux ベースの RTOS サポートは、dSPACE Release 2023-A 後に廃止されました。32 ビットベースのリアルタイムアプリケーションは生成できなくなりました。

### ソフトウェアサポートの廃止予定

**PHS バスハードウェア** dSPACE Release 2023-B は、PHS バスハードウェア(DS1006 Processor Board、DS1007 PPC Processor Board、およびすべての PHS バス I/O ボード)をサポートする最後のリリースです。これは、dSPACE Mid-Size シミュレータおよび dSPACE Simulator EcoLine にも当てはまります。  
新規プロジェクトには、モジュール型リアルタイムシステムである SCALEXIO を使用することをお勧めします。

**MotionDesk** MotionDesk および MotionDesk Blockset は、リリース 2023-B 以降に廃止されます。廃止までの間、MotionDesk はカメラセンサのシミュレーションをサポートします。MotionDesk を使用したレーダーセンサと LiDAR センサはサポート対象外となりました。詳細については、<http://www.dspace.jp/go/elamd> を参照してください。

**RTI CAN/LIN MultiMessage Blockset for SCALEXIO** RTI CAN MultiMessage Blockset for SCALEXIO および RTI LIN MultiMessage Blockset for SCALEXIO は、Release 2024-B 後に廃止される予定です。詳細については、<http://www.dspace.jp/go/elaRTIxxMMBS> を参照してください。  
新規 SCALEXIO プロジェクトには、Bus Manager を使用することをお勧めします。

#### dSPACE ハードウェアの廃止予定

**MicroAutoBox II** MicroAutoBox II の全バリエーション (1401/1507、1401/1511、1401/1511/1514、1401/1513、1401/1513/1514) は 2027 年末に廃止される予定です。2024 年末までは製品を購入することができ、ソフトウェアサポートは dSPACE Release 2026-B まで継続されます。  
新規プロジェクトには、MicroAutoBox III を使用することをお勧めします。

#### レガシーライセンステクノロジーの廃止

dSPACE Release 2023-A 以降、dSPACE Installation Manager はレガシーライセンステクノロジー (ドングルライセンス用の Wibu-Systems 社の WibuKey、フローティングネットワークライセンス用の Flexera 社の FlexNet) をサポートしなくなりました。これには、以下のような意味があります。

- dSPACE Installation Manager 23.1 (dSPACE Release 2023-A に付属) 以降では、dSPACE Release 2017-B で導入された CodeMeter ライセンステクノロジーのみサポートします。
- dSPACE Installation Manager 23.1 以降では、CodeMeter ライセンスに基づくレガシーライセンス用ファイルダウンロード (keys.dsp および license.dsp) を提供しなくなります。
- dSPACE Release 2023-A 以降を dSPACE Release 2017-A 以前と同じ PC にインストールすることは、サポートが終了しているため、推奨できません。
- dSPACE Release 2023-A 以降、dSPACE License Manager (Legacy) は dSPACE Installation Manager と共に出荷もインストールもされなくなりました。関連するコマンドラインユーティリティ (特に IMLicUtil.exe および InstallationReporter.exe ユーティリティなど) も廃止されています。
- それでもレガシーライセンス方式を使用する必要がある場合は、このテクノロジーをサポートする dSPACE Installation Manager の最新バージョンを使用する必要があります。今回のリリースでは、dSPACE Installation Manager 22.3 が最新バージョンです。dSPACE Installation Manager 22.3 は、<http://www.dspace.jp/go/IM> からダウンロードすることができます。

#### dSPACE Installation Manager によるドングル移行の廃止

dSPACE Release 2023-A 以降、dSPACE Installation Manager では、古いドングル (dSPACE Release 2017-A 以前に納品) から現在納品されている CmDongle への移行をサポートしなくなります。



dSPACE Release 2017-B 以降を使用するために移行が必要な古い dongle をまだお持ちの場合は、次のようになります。

- WibuKey dongle (タイプ 1) を使用しているユーザの場合、今回のリリースから移行ができなくなりました。dSPACE による dongle の交換が必要です。詳細については、<http://www.dspace.jp/go/DongleReplacement> を参照してください。
- 第一世代の CmDongle (タイプ 2、2-xxxx で始まるシリアル番号の白色のコネクタ) を使用しているユーザの場合、今回のリリースから移行ができなくなりました。dSPACE による dongle の交換が必要です。詳細については、<http://www.dspace.jp/go/DongleReplacement> を参照してください。
- シリアル番号が 3-xxxx で始まり、古いファームコードを使用した CmDongle (タイプ 3) を使用しているユーザの場合、dSPACE Installation Manager の旧バージョン (5.0 から 22.3 まで) を使用して移行する必要があります。dSPACE Installation Manager 22.3 のダウンロードについては、<http://www.dspace.jp/go/IM> を参照してください。

## 製品バージョンの概要

### 製品バージョン

次の表に、各製品の最新のリリースおよび過去 3 回のリリースのバージョン履歴を示します。新機能が追加されている場合は、本書での参照先を示しています。

製品名	dSPACE Release			
	2022-A	2022-B <sup>1)</sup>	2023-A	2023-B
AutomationDesk	6.7	2022-B	2023-A	2023-B 「AutomationDesk」(39 ページ)を参照。
Automotive Simulation Model	10.1	2022-B	2023-A	2023-B 「Automotive Simulation Models (ASM)」(41 ページ)を参照。
Bus Manager (スタンドアロン)	22.1	2022-B	2023-A	2023-B 「Bus Manager (スタンドアロン)」(59 ページ)を参照。
ConfigurationDesk for RapidPro	– <sup>2)</sup>	–	–	–
ConfigurationDesk	22.1	2022-B	2023-A	2023-B 「ConfigurationDesk」(67 ページ)を参照。
Container Manager	5.2	–	–	–
ControlDesk	7.6	2022-B	2023-A	2023-B 「ControlDesk」(87 ページ)を参照。
DCI Configuration Tool	3.13	2022-B	2023-A	2023-B 「DCI Configuration Tool」(107 ページ)を参照。
dSPACE AUTOSAR Compare	1.1	2022-B	2022-B	2022-B
dSPACE Bus API Package	–	–	2023-A	2023-B
dSPACE CAN API Package	4.0.8	2022-B	2023-A	2023-B

製品名	dSPACE Release			
	2022-A	2022-B <sup>1)</sup>	2023-A	2023-B
dSPACE ECU Flash Programming Tool	2.8	2022-B	2023-A	2023-B 「 <a href="#">dSPACE ECU Flash Programming Tool</a> 」(109 ページ)を参照してください。
dSPACE FlexRay Configuration Package	4.9	2022-B	2023-A	2023-B 「 <a href="#">dSPACE FlexRay Configuration Package</a> 」(111 ページ)を参照。
dSPACE Installation Manager	5.9	22.2	23.1	23.2 「 <a href="#">dSPACE Installation Manager</a> 」(113 ページ)を参照してください。
dSPACE Python Extensions	4.2	2022-B	2023-A	2023-B 「 <a href="#">dSPACE Python Extensions</a> 」(117 ページ)を参照してください。
dSPACE XIL API .NET	2022-A	2022-B	2023-A	2023-B 「 <a href="#">dSPACE XIL API .NET</a> 」(119 ページ)を参照。
ECU Interface Manager	2.11	2022-B	2023-A	2023-B 「 <a href="#">ECU Interface Manager</a> 」(123 ページ)を参照。
Firmware Manager	3.5	22.2	23.1	23.2 「 <a href="#">Firmware Manager</a> 」(127 ページ)を参照。
FPGA Programming Blockset <sup>3)</sup>	3.13	2022-B	2023-A	2023-B 「 <a href="#">FPGA Programming Blockset</a> 」(129 ページ)を参照。
MicroAutoBox III のファームウェア	6.0	22.2	23.1	23.2 「 <a href="#">MicroAutoBox III のファームウェア</a> 」(135 ページ)を参照。
Model and Sensor Interface Blockset	1.3	2022-B	2023-A	2023-B 「 <a href="#">Model および Sensor Interface Blockset</a> 」(139 ページ)を参照。
Model Compare	3.2	2022-B	2022-B	2023-B 「 <a href="#">Model Compare</a> 」(137 ページ)を参照してください。
Model Container Utility	–	2022-B	2023-A	2023-B
ModelDesk	5.7	2022-B	2023-A	2023-B 「 <a href="#">ModelDesk</a> 」(141 ページ)を参照。
Model Interface Package for Simulink	22.1	2022-B	2023-A	2023-B 「 <a href="#">Model Interface Package for Simulink</a> 」(143 ページ)を参照。
MotionDesk	4.10	2022-B	2023-A	2023-B 「 <a href="#">MotionDesk</a> 」(145 ページ)を参照。
MotionDesk Blockset	2.6.5	2022-B	2023-A	2023-B 「 <a href="#">MotionDesk</a> 」(145 ページ)を参照。
Real-Time Testing	5.2	2022-B	2023-A	2023-B 「 <a href="#">Real-Time Testing</a> 」(147 ページ)を参照。

製品名	dSPACE Release			
	2022-A	2022-B <sup>1)</sup>	2023-A	2023-B
RTI <sup>4)</sup>	7.18	2022-B	2023-A	2023-B 「RTI/RTI-MP および RTLib」(149 ページ)を参照してください。
RTI-MP <sup>5)</sup>	7.18	2022-B	2023-A	2023-B 「RTI/RTI-MP および RTLib」(149 ページ)を参照してください。
RTI Bypass Blockset	3.18	2022-B	2023-A	2023-B 「RTI Bypass Blockset」(151 ページ)を参照。
RTI CAN Blockset	3.4.14	2022-B	2023-A	2023-B
RTI CAN MultiMessage Blockset	5.8	2022-B	2023-A	2023-B 「RTI CAN MultiMessage Blockset」(153 ページ)を参照。
RTI Electric Motor Control Blockset	1.4.5	2022-B	2023-A	2023-B
RTI Ethernet Blockset	1.2.7	2022-B	2023-A	2023-B
RTI Ethernet (UDP) Blockset	1.4.7	2022-B	2023-A	2023-B
RTI LIN MultiMessage Blockset	3.8	2022-B	2023-A	2023-B 「RTI LIN MultiMessage Blockset」(155 ページ)を参照。
RTI RapidPro Control Unit Blockset	2.2.7	2022-B	2023-A	2023-B
RTI Synchronized Time Base Manager Blockset	1.4.4	2022-B	2023-A	2023-B 「RTI Synchronized Time Base Manager Blockset」(157 ページ)を参照してください。
RTI USB Flight Recorder Blockset	1.2.6	2022-B	2023-A	2023-B
RTI Watchdog Blockset	2.1.5	2022-B	2023-A	2023-B
SCALEXIO Firmware	6.0	2022-B	2023-A	2023-B 「SCALEXIO Firmware」(159 ページ)を参照。
センサシミュレーション	1.7	2022-B	–	–
Signal List Editor	–	–	–	2023-B 「Signal List Editor」(163 ページ)を参照してください。
SYNECT	2.13	2022-B	2023-A	2023-B 「SYNECT」(165 ページ)を参照。
SystemDesk	5.6	2022-B	2022-B	2023-B 「SystemDesk」(169 ページ)を参照してください。
TargetLink	5.2	2022-B	2022-B	2023-B 「TargetLink」(181 ページ)を参照してください。
VEOS	5.4	2022-B	2023-A	2023-B 「VEOS」(253 ページ)を参照。

1) dSPACE 製品の新しいバージョン体系を導入。

2) dSPACE Release 2021-B 以降、ConfigurationDesk for RapidPro は dSPACE Release の一部ではなくなりました。dSPACE の Web サイトから、専用のセットアッププロ

グラムを備えた独立したソフトウェアパッケージとしてダウンロードすることができます。  
<http://www.dspace.jp/go/CFDRP> を参照してください。

- 3) Release 2022-A 以前の RTI FPGA Programming Blockset。
- 4) 標準の I/O ブロックセットを含みます。
- 5) RTI Gigalink Blockset を含みます。

定期的に更新を行っていない場合は、新機能と必要な移行手順について、上記の各 dSPACE Release の『[新機能と移行手順](#)』マニュアルを参照してください。

## 製品の主な新機能

**概要** 各製品の重要な新機能の概要は次のとおりです。詳細については、各製品のセクションを参照してください。

### AutomationDesk

AutomationDesk の主な新機能は次のとおりです。

- Python 3.11 のサポート
- 新しい MicroLabBox II のサポート
- XIL API Convenience ライブラリに新しい Sleep ブロックが追加

新機能の詳細については、「[AutomationDesk 2023-B の新機能](#)」(39 ページ)を参照してください。

### Bus Manager(スタンドアロン)

Bus Manager(スタンドアロン)の主な新機能は次のとおりです。

- バス設定エレメントをコピー、カット、ペーストすることで、設定されたバス通信をバス設定に転送します。
- PDU ゲートウェイのシミュレーション(プレビュー版)
- LIN PDU の欠落している ECU を生成
- 通信マトリクスの IPDU にイベント制御タイミングエレメントを追加
- 強化された J1939-22 のサポート
- 多重化 IPDU で利用できる PDU Enable 機能
- Counter Signal 機能が検査に利用可能

詳細については、「[Bus Manager\(スタンドアロン\)2023-B の新機能](#)」(59 ページ)を参照してください。

### ConfigurationDesk

ConfigurationDesk の主な新機能は、次のとおりです。

- MicroLabBox II チャンネルタイプをサポートするためのファンクションブロックタイプの調整を含む、MicroLabBox II ターゲットプラットフォームをサポート。
- DS6242 D/A Board のチャンネルタイプをサポートするためのファンクションブロックタイプの調整を含む、DS6242 D/A Board(SCALEXIO)のサポート。
- 障害シミュレーション用の新しい SCALEXIO FSX ハードウェアのサポート。
- SCALEXIO と MicroLabBox II 間のモデルコンテナの移植性。
- バイナリ入出力変数を提供する FMU のサポート。

- 端末を提供する FMU のサポート。
- シミュレーション、検査、操作目的でのバス通信の設定に関する Bus Manager のさまざまな機能強化

詳細については、「[ConfigurationDesk](#)」(67 ページ)を参照してください。

## ControlDesk

ControlDesk の主な新機能は、次のとおりです。

### 一般的な機能強化

- 新しい MicroLabBox II のサポート
- Python 3.11 のサポート

### プラットフォーム/デバイスの機能強化

- 新しい MicroLabBox II のプラットフォームサポート
- SCALEXIO プラットフォーム : DS6242 D/A Board および DS6365-PE Automotive Ethernet Board のサポート
- GNSS デバイス : 新しいステータス変数 (使用中の衛星数の表示など)

詳細については、「[プラットフォーム管理およびプラットフォーム/デバイスの新機能 \(ControlDesk 2023-B\)](#)」(88 ページ)を参照してください。

### 変数管理の強化

- 変数マッピングの改善

詳細については、「[新しい変数管理機能 \(ControlDesk 2023-B\)](#)」(89 ページ)を参照してください。

### 計器の拡張

- 時間プロッタ、インデックスプロッタ : 凡例にシグナル名を表示
- 時間プロッタ : 時間およびデータカーソルのツールチップに接続名を表示
- バリアブルアレイ : カスタムヘッダーテキストの指定
- 音声コントローラ : MP3 ファイルのサポート
- 階層配列 : ユーザビリティの改善
- 日付と時刻計器カテゴリ

詳細については、「[新しい計器機能 \(ControlDesk 2023-B\)](#)」(90 ページ)を参照してください。

### Bus Navigator の機能強化

- SOME/IP-SD メッセージのデコードに AUTOSAR ファイルのポート情報を使用
- SOME/IP イベントパラメータのインポートとビジュアル表示 (プレビュー機能)
- バス計器と生成のカスタマイズ (プレビュー機能)
- J1939-22 : マルチ PG IPDU のためのコンパクトおよび操作バス計器の生成 (Bus Manager アプリケーションのみ)
- J1939-22 : Suspend PDU 送信操作サポート (Bus Manager アプリケーションのみ)
- PDU ユーザポート操作サポート (Bus Manager アプリケーションのみ)
- SecOC Freshness Overwrite Value 操作サポート (Bus Manager アプリケーションのみ)
- LIN スケジュールテーブルアクセスサポート (Bus Manager アプリケーションのみ)
- GTS 通信操作サポート (dSPACE FlexRay Configuration Package ベースのアプリケーションのみ)

- 領域ヘッダー内の PDU 状態の表示 (Bus Manager アプリケーションのみ)
- 「Signal Value」が有効なシグナルのビジュアル表示 (Bus Manager アプリケーションのみ)
- R22-11 ARXML ファイルのサポート
- ユーザビリティの改善

詳細については、「[新しい Bus Navigator 機能 \(ControlDesk 2023-B\)](#)」(93 ページ)を参照してください。

#### 電氣的欠陥シミュレーションの機能強化

- 障害シミュレーション用の新しい SCALEXIO FSX ハードウェアのサポート

詳細については、「[新しい電氣的欠陥シミュレーション機能 \(ControlDesk 2023-B\)](#)」(101 ページ)を参照してください。

#### ユーザインターフェースの操作拡張

- 変数とブロックグループの一元的な検索
- プレビュー機能

詳細については、「[新しいユーザインターフェース処理機能 \(ControlDesk 2023-B\)](#)」(101 ページ)を参照してください。

### dSPACE ECU Flash Programming Tool

dSPACE ECU Flash Programming Tool の主な新機能は次のとおりです。

- CAN\_MAX\_DLC の値を使用します。

新機能の詳細については、「[dSPACE ECU Flash Programming Tool 2023-B の新機能](#)」(109 ページ)を参照してください。

### dSPACE FlexRay Configuration Package

dSPACE FlexRay Configuration Package の主な新機能は、次のとおりです。

- AUTOSAR Classic Platform R22-11 のサポート

dSPACE FlexRay Configuration Tool の主な新機能は、次のとおりです。

- Create Corresponding ECU コマンドの機能強化。

新機能の詳細については、「[dSPACE FlexRay Configuration Package 2023-B の新機能](#)」(111 ページ)を参照してください。

### dSPACE XIL API

dSPACE XIL API の主な新機能は次のとおりです。

- Python 3.11 のサポート
- MAPort: 新しい MicroLabBox II のサポート
- EESPort: 新しい SCALEXIO FSX ハードウェアのサポート
- ECUPort: キャプチャしたデータを MF4 ファイルに保存できるようになりました。

新機能の詳細については、「[dSPACE XIL API .NET 2023-B の新機能](#)」(119 ページ)を参照してください。

### ECU Interface Manager

ECU Interface Manager の主な新機能は、次のとおりです。

- XCP タイムスタンプ同期
- 新しい MicroLabBox II のサポート

新機能の詳細については、「[ECU Interface Manager 2023-B の新機能](#)」(123 ページ)を参照してください。

---

#### Firmware Manager

Firmware Manager の主な新機能は次のとおりです。

- 新しい MicroLabBox II のサポート
- SCALEXIO ファームウェアアーカイブの事前選択時の動作の変更

詳細については、「[Firmware Manager 23.2 の新機能](#)」(127 ページ)を参照してください。

---

#### FPGA Programming Blockset

FPGA Programming Blockset 2023-B の主な新機能は、次のとおりです。

- FPGA 設計ツールのサポートを拡張。
- MicroLabBox のサポート II。
- FPGA フレームワークの拡張。

新機能の詳細については、「[FPGA Programming Blockset 2023-B の新機能](#)」(129 ページ)を参照してください。

---

#### Model Compare

Model Compare 2023-B の主な新機能は、次のとおりです。

- 新しいファイルベースの加速モデル比較メソッドは、純粋なモデルファイルを直接解析し、比較結果を即座に表示します。
- TargetLink サポート内蔵の textconv ドライバを使用すると、TargetLink ブロックのプロパティとパラメータの変更を詳細に分析することができます。

新機能の詳細については、「[Model Compare 2023-B の新機能](#)」(137 ページ)を参照してください。

---

#### ModelDesk

ModelDesk の主な新機能は次のとおりです。

- 全般: ModelDesk のコア技術は、最新のオペレーティングシステムに対応するために更新されました。
- Road Generator: OpenDRIVE フォーマット 1.6 の道路をインポートすることができ、エクスポート品質が向上しています。

新機能の詳細については、「[ModelDesk 2023-B の新機能](#)」(141 ページ)を参照してください。

---

#### Model Interface Package for Simulink

Model Interface Package for Simulink の主な新機能は、次のとおりです。

- 動作モデルのルートレベルでの Simulink ® In および Out ブロックのサポートが強化されました。
- MATLAB® R2023b のサポート

新機能の詳細については、「[Model Interface Package for Simulink](#)」(143 ページ)を参照してください。

---

**Python Extensions**

Python Extensions の主な新機能は次のとおりです。

- Python 3.11 のサポート

詳細については、「[dSPACE Python Extensions](#)」(117 ページ)を参照してください。

---

**Real-Time Testing**

Real-Time Testing の主な新機能は、次のとおりです。

- MicroLabBox II のサポート。

新機能の詳細については、「[Real-Time Testing 2023-B の新機能](#)」(147 ページ)を参照してください。

---

**RTI、RTI-MP、RTLib**

RTI、RTI-MP、および RTLib の主な新機能は、次のとおりです。

- MATLAB® R2023b のサポート。

詳細については、「[RTI/RTI-MP および RTLib の新機能](#)」(149 ページ)を参照してください。

---

**RTI CAN MultiMessage Blockset**

RTI CAN MultiMessage Blockset の主な新機能は次のとおりです。

- AUTOSAR Classic Platform R22-11 のサポート

新機能の詳細については、「[RTI CAN MultiMessage Blockset 2023-B の新機能](#)」(153 ページ)を参照してください。

---

**RTI LIN MultiMessage Blockset**

RTI LIN MultiMessage Blockset の主な新機能は次のとおりです。

- AUTOSAR Classic Platform R22-11 のサポート

新機能の詳細については、「[RTI LIN MultiMessage Blockset 2023-B の新機能](#)」(155 ページ)を参照してください。

---

**RTI Synchronized Time Base Manager Blockset**

RTI Synchronized Time Base Manager Blockset の主な新機能は、次のとおりです。

- 新しい MicroLabBox II のサポート。

新機能の詳細については、「[RTI Synchronized Time Base Manager Blockset 2023-B の新機能](#)」(157 ページ)を参照してください。

---

**SCALEXIO Firmware**

SCALEXIO Firmware の主な新機能は次のとおりです。

- DS6242 D/A Board のサポート。
- DS6365-PE Ethernet Board のサポート。
- SCALEXIO FSX ハードウェアのサポート。

新機能の詳細については、「[SCALEXIO Firmware 23.2 の新機能](#)」(159 ページ)を参照してください。



---

**SystemDesk**

SystemDesk 2023-B の主な新機能は、次のとおりです。

- AUTOSAR 準拠の変数記述のための A2L 生成。
- FMI Layered Standard に準拠した XCP サービス統合。
- AUTOSAR R22-11 リリースのサポート

新機能の詳細については、「[SystemDesk 2023-B の新機能](#)」(170 ページ)を参照してください。

---

**TargetLink**

TargetLink 2023-B の主な新機能は、次のとおりです。

- 連立一次方程式を解く SOLE (Systems Of Linear Equations) ソルバーブロック
  - カルマンフィルタや最小 2 乗曲線適合のような問題に数値的に効率的なソリューションを提供します。
- Data Store Read ブロックおよび Data Store Write ブロックでの動的エレメント指定
  - エレメントの選択と割り当てのための Simulink® Enable indexing オプションを有効にして、Data Store Read ブロックと Data Store Write ブロックの変数に動的にアクセスします。
- 豊富な C++ 構文
  - モジュールレベルで列挙クラスデータタイプ、静的キャスト、名前空間を使用します。
- Simulink® パラメータ Ensure output is virtual のサポート
  - MIL シミュレーションと簡易初期化モードで生成されたコードの差異を避けるために、拡張されていない Outport ブロックで Ensure output is virtual チェックボックスを選択することができます。
- .ddjson ファイルとして保存されたテキストベースの DD ファイル
  - バージョン管理システムやテキストベースのワークフローで DD ファイルを使用する可能性を提供します。

これらの新機能の詳細については、「[TargetLink 2023-B の新機能](#)」(182 ページ)を参照してください。

TargetLink の移行に関する (TargetLink、TargetLink AUTOSAR Module、TargetLink Data Dictionary) 詳細については、「[TargetLink 2023-B および TargetLink Data Dictionary 2023-B への移行](#)」(200 ページ)を参照してください。

---

**VEOS**

VEOS の主な新機能は、次のとおりです。

- VEOS を使用した協調シミュレーション
- VEOS シミュレーションへの QEMU の統合
- VEOS Simulator Console の機能強化

新機能の詳細については、「[VEOS 2023-B の新機能](#)」(253 ページ)を参照してください。

## 新しい使用例: データリプレイ

### 概要

dSPACE Release 2023-B のツールには、ADAS/AD ECU および AD ソフトウェアスタックの検証をサポートする新機能が追加されています。

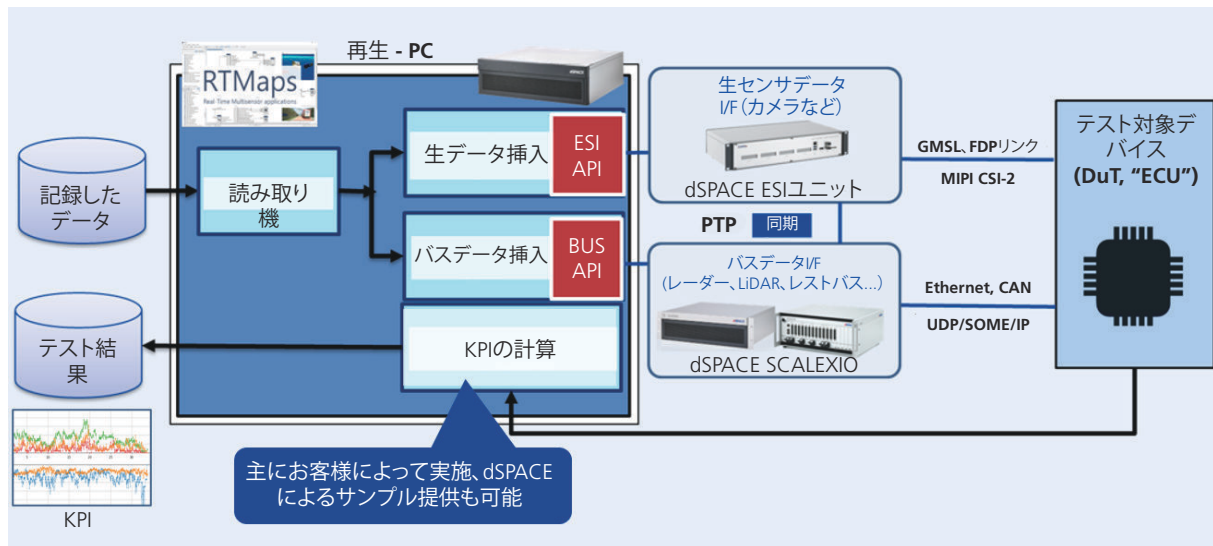
### センサ ECU の妥当性確認

ADAS/AD ECU は、スマートセンサ、ドメインコントローラ、あるいは高性能コンピュータとして使用することができます。これは通常、カメラ、レーダー、LiDAR、またはこれらすべてを組み合わせたものなどのセンサフロントエンドに接続され、物理環境を認識し、デジタル生データとして利用可能にします。生データは認識およびセンサフュージョンアルゴリズムに供給され、自車の周辺にある物体を検出します。検出された物体は、車両の環境モデルを生成するために同じユニット内で処理することもできるし、中央処理ユニットがさらなるアクションを計画できるように、車両ネットワーク内にさらに分散させることもできます。

記録したデータを使用して実際の ECU または仮想センサ ECU の妥当性を確認することができます。dSPACE ツールチェーンは、HIL (Hardware-in-the-Loop) シミュレーションおよび SIL (Software-in-the-Loop) シミュレーションでのデータ再生に対応しています。ADAS/AD ECU は、データを評価でき、KPI (主要パフォーマンスインジケータ) の計算を実行することによって結果をグラントゥールズデータと比較することができます。

実際の ADAS/AD ECU を HIL (Hardware-in-the-Loop) シミュレーションでテストすることができます。ECU は HIL シミュレータ (SCALEXIO システム) に接続され、バスデータを介してセンサ情報 (LiDAR ポイントクラウドなど) を提供します。また、この ECU は、カメラセンサの生データを提供する Environment Sensor Interface (ESI) ユニットに接続されています。SCALEXIO システムと ESI ユニットは、タイムスタンプ付きの生データとバスデータをデータストリームに供給する再生 PC に接続されています。これはたとえば、記録された実世界のデータを読み取り、接続された機器に配信する RTMaps によって行うことができます。Ethernet 上で PTP (Precision Time Protocol) を使用して、SCALEXIO システムと ESI ユニットが SCALEXIO のシステム時間に正確に同期されます。

SIL (Software-in-the-Loop) シミュレーションで ADAS/AD ECU (仮想センサ ECU) のソフトウェアスタックをテストすることができます。このようなテストは、専用のハードウェアがなくても PC やクラウド環境で実行することができます。センサ ECU のソフトウェアは、V-ECU に組み込まれている必要があります。バス通信は、バスシミュレーションコンテナ (BSC) に組み込まれている必要があります。RTMaps は、記録された実世界のデータを読み出し、これを BSC に配信します。



関連トピック

基本概要

実際のセンサ ECU の妥当性確認のためのデータ再生 (『SCALEXIO -ハードウェアおよびソフトウェアの概要』(1))



# 以前のリリースからの移行について

概要	最新の dSPACE Release の製品をインストールした後、いくつかの追加手順が必要になる場合があります。前回の dSPACE Release から更新する場合に必要な手順は、本書の製品固有の移行トピックスに記載されています。それ以前の dSPACE Release から更新する場合は、関連する『 <a href="#">新機能と移行手順</a> 』を参照してください。
----	---

## dSPACE Release 2023-B への移行

概要	Release 2023-B をインストールした後、いくつかの追加手順が必要な場合があります。
----	---

dSPACE Release 2023-A からの移行	<b>製品固有の移行手順</b> 製品ごとに必要な移行手順は、一般的に自動的に実行されます。例外については、製品ごとの移行に関する説明を参照してください。
-----------------------------	---

dSPACE Release 2022-B 以前のリリースからの移行	dSPACE Release 2022-B 以前のリリースから Release 2023-B への移行を行うには、その間の dSPACE Release バージョンへの移行手順も併せて実行する必要があります。dSPACE Release 2023-B のソフトウェアを使用すると、移行に必要なすべての手順を実行することができます。
------------------------------------	---

必要な移行手順の詳細については、各 dSPACE Release バージョンの『[新機能と移行手順](#)』ドキュメントを参照してください。

以前のリリースのドキュメント	以前のリリースの PDF ファイルの名前は、NewFeaturesAndMigrationxx.pdf (xx はリリース番号) です。
----------------	--

以前のリリースの『[新機能と移行手順](#)』ファイルは次の場所にあります。

- 最新の dSPACE Help インストールフォルダの、C:\Program Files\Common Files\dSPACE\Help 2023-B\Print\PreviousReleases を参照してください。

- dSPACE DVD の、\Doc\PreviousReleases を参照してください。
- <http://www.dspace.jp/go/migration> でダウンロードすることができます。ここには、かなり以前のリリースの『新機能と移行手順』もあります。

# Python 3.9 から Python 3.11 への Python スクリプトの移行

## 次のステップ

## 本章の内容

Python 3.11 の主な変更点.....	31
Python 3.11 の一般情報	
dSPACE ソフトウェアでの Python 3.11 の取り扱いの主な変更点.....	32
dSPACE Release 2023-B に含まれる Python 3.11 ディストリビューションの パッケージの概要を示します。	
Python インストールの使用に関する一般情報.....	33
Python インストールの並行使用に関する情報。	
技術的変更.....	35
Python 3.11 への移行に伴い、Python API には変更が発生しています。	
移行に関する製品固有の情報.....	35

## Python 3.11 の主な変更点

### Python 3.9 から Python 3.11 へ 移行する理由

Python 3.11 の最終リリースは 2022 年 10 月から提供されており、Python 3.9 のサポート終了は 2025 年 10 月に予定されています。最新の Python には、新機能やセキュリティアップデート、バグフィックスを提供する Python パッケージが含まれています。

このため、dSPACE は dSPACE Release 2023-B で Python 3.11 に切り替えます。

### Python のウェブサイトですぐ入手可能な関連ドキュメント

Python 3.11 の主要な変更点の概要については、「[技術的変更](#)」(35 ページ)を参照してください。Python 言語および Python 3.9 から Python 3.11 への環境のすべての変更に関する情報については、[www.python.org](http://www.python.org) を参照してください。

Python Software Foundation は、以下のドキュメントを提供しています。

- 『What's New from Python 3.9 to 3.10』  
<https://docs.python.org/3.10/whatsnew/3.10.html>
- 『What's New from Python 3.10 to 3.11』  
<https://docs.python.org/3.11/whatsnew/3.11.html>

## 関連トピック

基本概要

技術的変更.....35

## dSPACE ソフトウェアでの Python 3.11 の取り扱いの主な変更点

### 概要

Python 3.11 では、dSPACE ソフトウェアでの Python の取り扱いが一部変更されています。

### ライブラリ

dSPACE DVD で提供される Python 3.11 で使用するパッケージが次の表に示すように変更されています。

パッケージ	Python 3.9 (Release 2023-A)	Python 3.11 (Release 2023-B)
certifi	2022.12.7	2023.7.22
cff	-	1.15.1 (新規)
charset-normalizer	3.0.1	3.2.0
clr-loader	-	0.2.5 (新規)
comtypes	1.1.14	1.2.0
countourpy	1.0.6	1.1.0
Core	3.9.16.1 <sup>1)</sup>	3.11.5
cycler	0.11.0	0.11.0
fonttools	4.38.0	4.42.1
future	0.18.3	0.18.3
grpcio	1.51.1	1.57.0
grpcio_tools	1.51.1	1.57.0
idna	3.4	3.4
kiwisolver	1.4.4	1.4.5
lxml	4.9.2	4.9.3
matplotlib	3.6.3	3.7.2
numpy	1.24.1	1.25.2
packaging	23.0	23.1
pillow	9.4.0	10.0.0



パッケージ	Python 3.9 (Release 2023-A)	Python 3.11 (Release 2023-B)
pip	22.3.1	23.2.1
protobuf	4.21.12	4.24.2
pyparser	2.21	2.21
pyglet	2.0.3	2.0.9
pyparsing	3.0.9	3.0.9
pypubsub	4.0.3	4.0.3
Python-dateutil	2.8.2	2.8.2
pythonnet	2.5.3 <sup>1)</sup>	3.0.10 <sup>1)</sup>
pytz	2020.7	2023.3
pywin32	305.10 <sup>1)</sup>	306.10 <sup>1)</sup>
requests	2.28.2	2.31.0
scipy	1.10.0	1.11.2
six	1.16.0	1.16.0
urllib3	1.26.14	2.0.4
wxPython	4.2.0	4.2.1
yapsy	1.12.2	1.12.2

<sup>1)</sup> このコンポーネントには、dSPACE 独自のバグフィックスが含まれています。Python の標準配布物に含まれるコンポーネントと置き換えないでください。

## Python インストールの使用に関する一般情報

### 概要

Python 3.9 をサポートする dSPACE Release 2023-A 以前の dSPACE ソフトウェアと、Python 3.11 をサポートする dSPACE Release 2023-B の dSPACE ソフトウェアを併用する場合は、両方の Python バージョンが PC にインストールされ、並列で使用することが可能です。

### Python 3.9 および Python 3.11 の両方のバージョンを使用する場合の制約

同時に両方のバージョンの Python を使用することができます。ただし、次の制約があることに注意が必要です。

- PY および PYW ファイルを関連付ける場合、設定を行えるのは 1 つの Python バージョンのみです。通常、これはインストールされている中で最も新しい Python バージョンです。

#### 注記

dSPACE Release 2023-A 以前を dSPACE Release 2023-B のインストール後にインストールする場合、Python 3.9 が最後にインストールされます。したがって、Python ファイルを開くために PythonWin 3.9 が登録されています。ファイルの関連付けを PythonWin 3.11 に設定するには、<DVDRoot>\Products\Common\Python3.11 から PythonInstaller.exe を再度実行してください。

一時的に PythonWin 3.11 で Python ファイルを開くには、ファイルのコンテキストメニューにある [Edit with PythonWin 3.11] コマンドを使います。

- 環境変数は、Python の両バージョンによって使用されます。PYTHONHOME の値などの環境変数の値は、使用する Python インストールに設定する必要があります。Python が設定する環境変数の概要については、次のサイトを参照してください。 <https://docs.python.org/3.11/using/windows.html>。

#### 注記

特定が困難な副次的な影響が発生しないようにするためには、環境変数を使用しないことをお勧めします。同じ理由から、Python インストールパスによってシステムパスを拡張しないでください。

### Python の両バージョンの並列動作による dSPACE テストオートメーションの使用

テストオートメーションスクリプトに dSPACE Python モジュールが使用されているがスクリプトを移行したくない場合は、Python の両バージョンを並列動作させる必要があります。Python 3.9 対応の dSPACE Python モジュールは、Release 2023-A 以前まで利用することができます。

#### 注記

Python スクリプトを使用する場合は、同じリリースの dSPACE ソフトウェアのみを使用することをお勧めします。両方のバージョンの Python を同じアプリケーションコンテキストで使用する (AutomationDesk により旧バージョンの ControlDesk へのアクセスを自動化しているなど)、または Python 3.11 をサポートしていないバージョンの Real-Time Testing をアクティブ化して AutomationDesk を使用すると、競合が発生する場合があります。

## 技術的変更

**概要** Python 3.9 から Python 3.11 への移行に伴い、Python API には変更が発生しています。

### 注記

手作業による変更が必要な数に関係なく、移行したコードは必ずテストしてください。

**移行の問題** Python 3.11 から pythonnet 3.0 への以下の変更により、Python スクリプトの手作業による移行が必要になります。

**pythonnet: .NET メソッドの戻り値の変更** Python 3.11 で pythonnet 3.0 を使用すると、インターフェースを返す .NET メソッドは自動的にそのインターフェースにラップされます。したがって、実装クラスの一部であるメンバーにアクセスすることはできません。インターフェースの一部であるメンバーにのみアクセスすることができます。

製品固有の移行手順については、「[移行に関する製品固有の情報](#)」(35 ページ)を参照してください。

**pythonnet: 列挙値の暗黙の型変換が無効になる** C# の enum から Python の int への暗黙の型変換が無効になりました。Enum オブジェクトが関連する整数値を持つメンバーを提供しない場合、enum メンバー、たとえば MyEnum.Option を使うか、enum コンストラクタ、たとえば MyEnum(<intValue>) を使用する必要があります。

**pythonnet: .NET のコレクションと配列が Python コレクションに自動的に変換されなくなりました。** pythonnet 2.4 で導入された .NET のコレクションと配列の自動変換は、pythonnet 3.0 で取り消されました。したがって、Python コレクションのメソッドとプロパティを .NET コレクションのメソッドとプロパティに移行する必要があります。たとえば、len(MyObject) ではなく、MyObject.Count を使用してください。

## 移行に関する製品固有の情報

**Python の内容を編集できる製品** 次の dSPACE 製品では Python スクリプトを明示的にまたは暗黙的に使用します。

- AutomationDesk
- Bus Manager (スタンドアロン)
- ConfigurationDesk
- ControlDesk
- ModelDesk
- dSPACE Platform API Package (Python Extensions、XIL API.NET、Platform Management API)
- Real-Time Testing

- SYNECT
- SystemDesk

それ以外のすべての dSPACE 製品では、Python の移行による影響はありません。

カスタムユーザ関数など、上記の製品のすべてのカスタムスクリプトについて、必要な移行を確認する必要があります。製品固有の詳細については、以下を参照してください。

#### AutomationDesk の使用

pythonnet 3.0 で列挙値の暗黙の型変換が無効になったため、XIL API Enum の取り扱いが変更されました。

Exec ブロックの Python コードで XIL API Enum を使用するには、関連する整数値の代わりに指定された Enum オブジェクトを使用する必要があります。

例: 以下の例では、Python 3.9 の pythonnet 2.5.3 と、Python 3.11 の pythonnet 3.0 の移行コードで、キャプチャの継続時間単位に秒を指定する方法を示しています。

```
# pythonnet 2.5.3
Capture.DurationUnit = 0
# pythonnet 3.0
Capture.DurationUnit = xilapi.Common.Capturing.Enum.DurationUnit.eSECOND
```

#### Bus Manager(スタンドアロン)の使用

考慮すべき製品固有の移行の問題はありません。

#### ConfigurationDesk の使用

考慮すべき製品固有の移行の問題はありません。

#### ControlDesk の使用

考慮すべき製品固有の移行の問題はありません。

#### dSPACE Platform API Package の使用

**Python Extensions の使用** 考慮すべき製品固有の移行の問題はありません。

**Platform Management API の使用** 考慮すべき製品固有の移行の問題はありません。

**XIL API .NET の使用** インターフェースオブジェクトを返す .NET メソッドの変更された動作は、コーデックを使用することによって XIL API .NET で内部的に調整されます。スクリプトを移行する必要はありません。

例: MAPort の read メソッドは、IBaseValue Read(variableName) として定義されます。pythonnet 3.0 を使用している場合、curValue = MAPort.Read(variableName) を実行すると、curValue が IBaseValue として返されます。

- IBaseValue インターフェースは Value プロパティを提供しないため、val = curValue.Value を実行するなどして curValue を読むと例外が発生します。
- curValue を書いても、たとえば curValue.Value = val を実行してもオブジェクトは変更されませんが、Python は例外を発生させることなく、新しいプロパ

ティを作成します。そのため、pythonnet 3.0 で変更された実装に起因するアプリケーションの不正な動作を検出することは困難です。

このようなアプリケーションの隠れた不正な動作の可能性を回避するため、dSPACE XIL API .NET 実装の次のインターフェースは、内部的に実装クラスに移行されています。

- IBaseError
- IBaseErrorBuilder
- IBaseValue
- ICapture
- ICaptureEvent
- ICompuMethod
- IDataFileSegment
- IDuration
- IEESPort
- IEESPortConfig
- IErrorSet
- IMAPort
- IMAPortConfig
- IMatrixValue
- IPort
- IScalarValues
- ISignalDescription
- ISignalSegment
- ISymbol
- ITestbench
- IVectorValue
- IWatcher

---

#### ModelDesk の使用

考慮すべき製品固有の移行の問題はありません。

---

#### Real-Time Testing の使用

Real-Time Testing では、ホスト PC やリアルタイムプラットフォームで Python が使用されます。

インタープリタは Python の各種バージョンをサポートしています。

- Python 3.11:
  - ホスト PC 上でのインタープリタ
- Python 3.6:
  - dSPACE Release 2019-B 以降の VEOS で使用されるインタープリタ
  - dSPACE Release 2019 B 以降の MicroAutoBox III で使用されるインタープリタ
  - dSPACE Release 2020 B 以降の SCALEXIO/DS6001 で使用されるインタープリタ
  - dSPACE Release 2023-B 以降の MicroAutoBox II で使用されるインタープリタ
- Python 2.7.11:
  - その他すべてのリアルタイムプラットフォーム


カスタムスクリプトを手作業で Python 3.11 に移行する必要があります。Real-Time Testing は、Python 3.9 から Python 3.11 へと自動的にカスタムスクリプトを移行する機能を一切提供していません。

アクティブ化した Real-Time Testing のバージョンと、Real-Time Testing を使用する dSPACE ソフトウェア (AutomationDesk など) は、同じバージョンの Python をサポートする必要があります。

---

### SYNECT の使用

Python 3.11 で配布された新しい pythonnet 3.0 バージョンは、スクリプトの取り扱いが厳しくなったため (暗黙の型変換の許容が以前に比べて少ないなど)、スクリプトの移行に問題が生じる可能性があります。

カスタム Python アダプタのために、SNDataExchange .NET DLL を使用することは、SYNECT による pythonnet の一般的な使用例の一つです。「[アダプタクラスの実装の基礎](#)」(『SYNECT ガイド』)を参照してください

#### 注記

SYNECT は Python 3.11 ではなく、IronPython インタプリタをサーバスクリプトに使用しているため、SYNECT サーバスクリプトは影響を受けません。

---

### SystemDesk の使用

SystemDesk のデモスクリプトは、Python 3.11 用に修正する必要はありませんでした。したがって、カスタムスクリプトの移行も必要ないと考えられます。

# AutomationDesk

## 次のステップ

## 本章の内容

AutomationDesk 2023-B の新機能.....	39
AutomationDesk 2023-B への移行.....	40

## AutomationDesk 2023-B の新機能

### 一般的な機能強化

**Python 3.11 のサポート** AutomationDesk は Python 3.11 と付属の pythonnet 3.0 パッケージをサポートしています。必要な移行手順については、「[Python 3.9 から Python 3.11 への Python スクリプトの移行](#)」(31 ページ)を参照してください。

**新しい dSPACE ハードウェアのサポート** AutomationDesk Platform Manager を使用すると、新しい MicroLabBox II を登録し、プラットフォームにシミュレーションアプリケーションをダウンロードすることができます。新しい MicroLabBox II の詳細については、「[一般的な機能拡張および変更](#)」(13 ページ)を参照してください。

### XIL API Convenience ライブラリの機能強化

**シミュレーションプラットフォームの時間を使用した Sleep ブロック** XIL API Convenience ライブラリは、アプリケーションを実行するシミュレーションプラットフォームの時間情報を使用する Sleep ブロックを提供するようになりました。このシミュレーションプラットフォームは、参照されている MAPort データオブジェクトによって表されます。

オフラインモードで、または MAPort データオブジェクトが初期化されていない場合、Sleep ブロックの動作は、ホスト PC のシステムクロックを使用するメインライブラリの Sleep ブロックと同等です。

詳細については、「[Sleep \(XIL API\)](#)」(『AutomationDesk シミュレーションプラットフォームへのアクセス』)を参照してください。

## AutomationDesk 2023-B への移行

### 移行に関する一般的な注意点

新しいバージョンの AutomationDesk で AutomationDesk プロジェクトを開くと、移行が必要かどうかソフトウェアによって自動的に検出されます。メッセージダイアログで[OK]をクリックすると、移行が開始されます。移行後のプロジェクトは別のパスや名前で作成してください。

#### 注記

AutomationDesk の新しいバージョンで以前のプロジェクトを開く場合、あらかじめ以下の条件を満たしておく必要があります。

- プロジェクトおよびリンクされたカスタムライブラリのバックアップを作成しておく必要があります。
- AutomationDesk が正常に実行されている必要があります。Log Viewer にエラーメッセージが表示されていないことが必要です。
- 組み込みライブラリ、必要なカスタムライブラリ、およびその他のパッケージが正常にロードされている必要があります。

バージョン管理システムを使用する場合、移行を正常に行うにはいくつかの前提条件を満たしている必要があります。詳細については、「[バージョン管理下のプロジェクトまたはカスタムライブラリを移行する方法](#)」(『AutomationDesk の基礎的な使用法』)を参照してください。

詳細については、「[移行](#)」(『AutomationDesk New Features and Migration』)を参照してください。

#### 注記

dSPACE Release 2023-B では、AutomationDesk は Python 3.11 をサポートしています。AutomationDesk での Python スクリプトの変更と移行手順については、「[移行に関する製品固有の情報](#)」(35 ページ)を参照してください。

### ASAM XIL API 2.1 のサポートの終了

AutomationDesk 2023-B は、ASAM XIL API 2.1 以前に基づく XIL API サーバをサポートしなくなりました。このことは、dSPACE XIL API .NET ベースのサーバおよびサードパーティのサーバに関連しています。

dSPACE XIL API サーバは、ASAM XIL API 2.2 ベースの dSPACE Release 2022-A 以降で使用することができます。



# Automotive Simulation Models (ASM)

## 次のステップ

## 本章の内容

すべての ASM 製品.....	42
ASM Battery.....	43
ASM Diesel Engine.....	45
ASM Drivetrain Basic.....	46
ASM Electric Components.....	47
ASM Fuel Cell.....	48
ASM Gasoline Engine.....	50
ASM Traffic.....	51
ASM Trailer.....	52
ASM Truck.....	54
ASM Vehicle Dynamics.....	56

## すべての ASM 製品

### 次のステップ

### 本章の内容

すべての ASM デモモデルの変更.....	42
すべての ASM ブロックセットの移行.....	42

## すべての ASM デモモデルの変更

### SCALEXIO でのリアルタイム OS サポート

今後、SCALEXIO プロセッシングハードウェアは、64 ビット Linux ベースのリアルタイム OS のみを提供します。そのため、ASM デモモデルは 32 ビットをサポートしなくなり、64 ビットのリアルタイム OS のみをサポートするようになりました。

### 通信信号

ASM モジュールの構造が改善されました。他のモジュールとの通信に使用されるインターフェースは、すべての ASM モジュールのトップレベルに移動しました。これで、モデルをさらに簡単に調整することができます。また、この目的のために、Interface\_Out の構造も最適化されています。

### 初期化ファイルの変更

ASM デモモデルの初期化ファイルが変更されました。

以下の情報フィールドは削除されました。

- 'Author'
- 'Version'
- 'LastModified'

移行されたモデルの初期化ファイルは変更されません。

## すべての ASM ブロックセットの移行

### デモモデルの Operator ライブラリブロック

デモモデル内のすべての Operator ライブラリブロックは、オープンスタンダードの ASM ライブラリにリンクされています。両ライブラリバージョン間を切り替えるための ASM Model Tools の OPERATOR ボタンと DEVELOPER ボタンが削除されました。

# ASM Battery

## 次のステップ

## 本章の内容

ASM Battery Blockset 2023-B の新機能.....	43
ASM Battery デモモデルの変更.....	43
ASM Battery Blockset 2023-B への移行.....	44

## ASM Battery Blockset 2023-B の新機能

**CHARGING\_TIME ブロック** 新しいブロックです。充電電力とバッテリー容量に応じて、バッテリー充電の残り時間を計算します。

**BATTERY\_PACK\_DELTA\_VOLTAGE ブロック** ASMSignalBus が拡張され、kWh 単位のバッテリー容量が追加されました。電力損失信号ベクトルは、ASMSignalBus の 1 次元信号に変更されました。これによって、リセットと SOC リセットを区別することが可能になりました。SOC リセットでは、充電状態のみがリセットされます。

**BATTERY\_PACK\_VOLTAGE ブロック** ASMSignalBus が拡張され、kWh 単位のバッテリー容量が追加されました。これによって、リセットと SOC リセットを区別することが可能になりました。SOC リセットでは、充電状態のみがリセットされます。

**CSC\_INTERFACE\_IN ブロック** 新しいブロックです。これは、たとえばバッテリーなどのプラントモデルから来る信号を、セル監視回路(CSC)にインターフェースで接続します。

**CSC\_INTERFACE\_OUT ブロック** 新しいブロックです。これは、セル監視回路(CSC)からプラントモデル(たとえばバッテリー)へ向かう信号をインターフェースで接続します。

## ASM Battery デモモデルの変更

**充電電源リレー** 車両充電制御 ECU は現在、充電ステーションのケーブル接続と高電圧バッテリー側を制御するために、2 つの電源リレーをシミュレートしています。

---

充電時間	バッテリー管理システムは、バッテリーの充電時間を計算できるように拡張されました。
------	--

---

バッテリー管理システムソフト ECU	<p>バッテリー管理システム(BMS)は、分散型トポロジを表現するために再構築されました。現在では、主に DC リンクを制御し、充電状態を推定する中央バッテリー管理ユニット(BMU)と、複数のセル監視回路(CSC)モジュールで構成されています。</p> <p>CSC の主なシミュレーションタスクは、セルのモニタリングとバランスングです。CSC あたりの監視セル数は、アクティブセル数を CSC モジュール数で割ることで自動的に決定されます。</p> <p>ASM バッテリーのデモモデルには、6 つの CSC モジュールが付属しています。この数値は要件に合わせて簡単に変更することができます。</p>
--------------------	---

## ASM Battery Blockset 2023-B への移行

---

BATTERY_PACK_VOLTAGE ブロック	新しい SOC リセット入力はゼロ定数で接続されています。
---------------------------	-------------------------------

---

BATTERY_PACK_DELTA_VOLTAGE ブロック	新しい SOC リセット入力はゼロ定数で接続されています。
---------------------------------	-------------------------------

---

関連トピック	基本概要
--------	------

[ASM モデルの移行 \(『ASM ユーザガイド』📖\)](#)

# ASM Diesel Engine

## ASM Diesel Exhaust Blockset 2023-B への移行

### MASS\_FRACTION ブロック

すべてのエンジン流量が 0 の場合、未燃焼空気の質量分率は 1 に設定されます。0 に設定する以前の動作は、移行中に復元されます。

### 関連トピック

#### 基本概要

[ASM モデルの移行 \(『ASM ユーザガイド』\)](#)

## ASM Drivetrain Basic

### 次のステップ

### 本章の内容

ASM Drivetrain Basic Blockset 2023-B の新機能.....	46
ASM Drivetrain Basic Blockset 2023-B への移行.....	46

## ASM Drivetrain Basic Blockset 2023-B の新機能

### TORQUE\_INTERVENTION\_ CONTROL ブロック

シミュレーション中の最初のギア要求が、連続するギア要求に比べて 1 ステップサイズ遅延する問題が修正されました。

## ASM Drivetrain Basic Blockset 2023-B への移行

### TORQUE\_INTERVENTION\_ CONTROL ブロック

ブロックが移行されるため、以前の動作は変更されません。新しいブロック機能を使用するには、移行ブロックを削除し、個々のポートを接続します。

### 関連トピック

### 基本概要

[ASM モデルの移行 \(『ASM ユーザガイド』\)](#)

# ASM Electric Components

## 次のステップ

## 本章の内容

ASM Electric Components Blockset 2023-B の新機能.....	47
ASM Electric Components デモモデルの変更.....	47
ASM Electric Components Blockset 2023-B への移行.....	47

## ASM Electric Components Blockset 2023-B の新機能

### ELECTRIC\_MACHINE\_BASIC ブロック

現在では、発電機モードとモーターモードでモデルを別々にパラメータ化するために、異なるトルク速度曲線が使用されています。また、機械効率は効率表に基づいて計算されるか、銅損のみを使用して計算されるようになりました。

## ASM Electric Components デモモデルの変更

### バッテリー電気自動車とハイブリッド車のデモモデル

車両充電制御 ECU は現在、充電ステーションのケーブル接続と高電圧バッテリー側を制御するために、2 つの電源リレーをシミュレートしています。

### AURELION Manager プロジェクト

アニメーションの対象となるデモモデルには、AURELION Manager プロジェクトが含まれるようになりました。これにより、デモモデルの AURELION アニメーションを簡単に開始することができます。

## ASM Electric Components Blockset 2023-B への移行

### ELECTRIC\_MACHINE\_BASIC ブロック

モデルは新しいパラメータで拡張されています。

## 関連トピック

### 基本概要

[ASM モデルの移行 \(『ASM ユーザガイド』\)](#)

# ASM Fuel Cell

## 次のステップ

## 本章の内容

ASM Fuel Cell Blockset 2023-B の新機能.....	48
ASM Fuel Cell デモモデルの変更.....	49
ASM Fuel Cell Blockset 2023-B への移行.....	49

## ASM Fuel Cell Blockset 2023-B の新機能

### ELECTROCHEMISTRY\_ PHYSICAL ブロック

このブロックでは、内部電流をモデル化するために短絡抵抗を設定できるようになりました。さらに、標準化された計測データからブロックパラメータを計算する処理機能が最適化され、新しい膜短絡抵抗を含むように拡張されました。最新の燃料電池シャットダウンレチンは、低いセル電圧をしきい値として使用するため、このブロックでは、半経験的モデリングアプローチにより、0.5V 未満のセル電圧に対する分極曲線のシミュレーションが可能になりました。

### GAS\_DIFFUSION\_LAYER ブロック

このブロックは現在、水素クロスオーバー（膜を透過する水素の拡散）をサポートしています。酸素が利用可能であれば、水素クロスオーバー全体が酸素と完全に反応し、カソード触媒層で水になると想定されます。そうでない場合は、水素はさらにカソードガス流路に拡散します。

### MAPPING\_MDOT\_HDOT ブロック

このブロックは現在、水素クロスオーバー（膜を透過する水素の拡散）をサポートしています。酸素が利用可能であれば、水素クロスオーバー全体が酸素と完全に反応し、カソード触媒層で水になると想定されます。そうでない場合は、水素はさらにカソードガス流路に拡散します。

### MEMBRANE ブロック

このブロックは現在、水素クロスオーバー（膜を透過する水素の拡散）をサポートしています。酸素が利用可能であれば、水素クロスオーバー全体が酸素と完全に反応し、カソード触媒層で水になると想定されます。そうでない場合は、水素はさらにカソードガス流路に拡散します。

### STACK\_ASSEMBLY ブロック

ブロック機能は、伝導によるスタック内の熱交換と、放射および自由対流による環境との熱交換をモデル化するために拡張されています。



## ASM Fuel Cell デモモデルの変更

充電電源リレー	車両充電制御 ECU は、充電ステーションのケーブル接続と高電圧バッテリー側を制御するための 2 つの電源リレーをシミュレートするようになりました。
スタックレイアウトの自動生成	<p>ASM Fuel Cell デモモデルには、さまざまなセル構成のスタックレイアウトを作成する Python スクリプトが付属しています。Python スクリプトは、デモモデルの ControlDesk プロジェクトの一部です。</p> <p>スクリプトが実行されると、その時点でプラットフォーム上で動作しているリアルタイムまたは VEOS アプリケーションから、実際の設定に必要な情報が読み込まれます。</p>

## ASM Fuel Cell Blockset 2023-B への移行

ELECTROCHEMISTRY_PHYSICAL ブロック	ブロックのライブラリリンクは ELECTROCHEMISTRY_PHYSICAL_2_0 に変更されません。
GAS_DIFFUSION_LAYER ブロック	<p>値が 0 の Constant ブロック 4 個が新しい入力ポートに接続され、Terminator ブロック 3 個が新しい出力ポートに接続されています。</p> <p>ガス拡散層を通過する物質フラックスの一次ダイナミクスのために、時定数パラメータがブロックに追加されています。</p>
MEMBRANE ブロック	<p>値が 0 の Constant ブロック 6 個が新しい入力ポートに接続され、Terminator ブロック 8 個が新しい出力ポートに接続されています。</p> <p>膜中の水素の拡散係数が、新しいパラメータとしてブロックに追加されています。</p>
MAPPING_MDOT_HDOT ブロック	値 0 の Constant ブロック 5 個が新しい入力ポートに接続されています。

### 関連トピック

基本概要

[ASM モデルの移行 \(『ASM ユーザガイド』\)](#)

# ASM Gasoline Engine

## ASM Gasoline Engine Blockset 2023-B への移行

### MASS\_FRACTION ブロック

すべてのエンジン流量が 0 の場合、未燃焼空気の質量分率は 1 に設定されます。0 に設定する以前の動作は、移行中に復元されます。

### 関連トピック

#### 基本概要

[ASM モデルの移行 \(『ASM ユーザガイド』\)](#)

# ASM Traffic

## ASM Traffic デモモデルの変更

---

**AURELION Manager プロジェクト**

デモモデルには、AURELION Manager プロジェクトが含まれるようになりました。これにより、デモモデルの AURELION アニメーションを簡単に開始することができます。

# ASM Trailer

## 次のステップ

## 本章の内容

ASM Trailer Blockset 2023-B の新機能.....	52
ASM Trailer デモモデルの変更.....	53

## ASM Trailer Blockset 2023-B の新機能

**SUSPENSION\_FORCE\_KINEMATICS\_TRAILER\_FRONT** ブロック  
最後のブロックの出力ポートは、gF\_dq\_Rear\_dt[2x2][][]に名称変更されました。

**SUSPENSION\_FORCE\_KINEMATICS\_TRAILER\_REAR** ブロック  
最後のブロックの出力ポートは、gF\_dq\_Rear\_dt[2x2][][]に名称変更されました。

**SUSPENSION\_FORCE\_KINEMATICS\_TRAILER\_REAR\_2ND** ブロック  
最後のブロックの出力ポートは、gF\_dq\_Rear\_dt[2x2][][]に名称変更されました。

**SUSPENSION\_FORCE\_KINEMATICS\_TRAILER\_REAR\_3RD** ブロック  
最後のブロックの出力ポートは、gF\_dq\_Rear\_dt[2x2][][]に名称変更されました。

**TIRE\_MODEL\_MAGIC\_FORMULA** ブロック  
無負荷タイヤ半径と一定の垂直タイヤ剛性がすべてのタイヤモデルのASMSignalBus に追加されました。

**TIRE\_MODEL\_TMEASY** ブロック  
無負荷タイヤ半径と一定の垂直タイヤ剛性がすべてのタイヤモデルのASMSignalBus に追加されました。

## ASM Trailer デモモデルの変更

---

**AURELION Manager プロジェクト** デモモデルには、AURELION Manager プロジェクトが含まれるようになりました。これにより、デモモデルの AURELION アニメーションを簡単に開始することができます。

## ASM Truck

### 次のステップ

### 本章の内容

ASM Truck Blockset 2023-B の新機能.....	54
ASM Truck デモモデルの変更.....	54

## ASM Truck Blockset 2023-B の新機能

**SUSPENSION\_FORCE\_KINEMATICS\_REAR\_2ND** ブロック  
最後のブロックの出力ポートは、gF\_dq\_Rear\_dt[2x2][][]に名称変更されました。

**SUSPENSION\_FORCE\_KINEMATICS\_REAR\_3RD** ブロック  
最後のブロックの出力ポートは、gF\_dq\_Rear\_dt[2x2][][]に名称変更されました。

**TIRE\_MODEL\_MAGIC\_FORMULA** ブロック  
無負荷タイヤ半径と一定の垂直タイヤ剛性がすべてのタイヤモデルの ASMSignalBus に追加されました。

**TIRE\_MODEL\_TMEASY** ブロック  
無負荷タイヤ半径と一定の垂直タイヤ剛性がすべてのタイヤモデルの ASMSignalBus に追加されました。

## ASM Truck デモモデルの変更

**AURELION Manager** プロジェクト  
デモモデルには、AURELION Manager プロジェクトが含まれるようになりました。これにより、デモモデルの AURELION アニメーションを簡単に開始することができます。

**Traffic デモモデルが付属する新しい Truck Trailer**  
ASM Truck ライブラリに Traffic デモモデルが付属する Truck Trailer が追加されました。ASM Traffic コンポーネントは、Truck Trailer デモを拡張して、周辺車両や、ASM 車両に装備された複数のセンサモデルの動きもシミュレートすることができるようになりました。

他の ASM デモモデルと同様に、新しいデモモデルには以下のプロジェクトが含まれています。

- ASM
- ModelDesk
- ControlDesk
- AURELION Manager

## ASM Vehicle Dynamics

### 次のステップ

### 本章の内容

ASM Vehicle Dynamics Blockset 2023-B の新機能.....	56
ASM Vehicle Dynamics デモモデルの変更.....	57
ASM Vehicle Dynamics Blockset 2023-B への移行.....	57

## ASM Vehicle Dynamics Blockset 2023-B の新機能

### ADAMS2ASM コンバータ

ADAMS2ASM コンバータでは、ADAMS シミュレーションを使用した車両の自動パラメータ化のための新しいユーザインターフェースを利用できるようになりました。Adams Car®とシナリオのデータを ModelDesk に自動的に変換し、シミュレーション結果を検証できるようになりました。

### SUSPENSION\_FORCE\_ KINEMATICS\_FRONT ブロック

最後の出力ポートは、gF\_dq\_Front\_dt[2x2][ ]に名称変更されました。

### SUSPENSION\_FORCE\_ KINEMATICS\_REAR ブロック

最後の出力ポートは、gF\_dq\_Rear\_dt[2x2][ ]に名称変更されました。

### TORQUE\_INTERVENTION\_ CONTROL ブロック

シミュレーション中の最初のギア要求が、連続するギア要求に比べて 1 ステップサイズ遅延する問題が修正されました。

### TIRE\_MODEL\_MAGIC\_ FORMULA ブロック

無負荷タイヤ半径と一定の垂直タイヤ剛性がすべてのタイヤモデルの ASMSignalBus に追加されました。

### TIRE\_MODEL\_TMEASY ブロック

無負荷タイヤ半径と一定の垂直タイヤ剛性がすべてのタイヤモデルの ASMSignalBus に追加されました。



## ASM Vehicle Dynamics デモモデルの変更

**AURELION Manager プロジェクト** デモモデルには、AURELION Manager プロジェクトが含まれるようになりました。これにより、デモモデルの AURELION アニメーションを簡単に開始することができます。

## ASM Vehicle Dynamics Blockset 2023-B への移行

**TORQUE\_INTERVENTION\_CONTROL ブロック** ブロックが移行されるため、以前の動作は変更されません。新しいブロック機能を使用するには、移行ブロックを削除し、個々のポートを接続します。

### 関連トピック

#### 基本概要

[ASM モデルの移行 \(『ASM ユーザガイド』\)](#)



# Bus Manager (スタンドアロン)

## 次のステップ

## 本章の内容

Bus Manager (スタンドアロン) 2023-B の新機能.....	59
Bus Manager (スタンドアロン) 2023-B への移行.....	62
以前のバージョンで作成されたプロジェクトとアプリケーションの 移行.....	62
Bus Manager (スタンドアロン) の廃止.....	65

## Bus Manager (スタンドアロン) 2023-B の新機能

### 新たにサポートされた AUTOSAR リリース

Bus Manager (スタンドアロン) では、AUTOSAR Classic Platform Release R22-11 に基づく AUTOSAR システムデスクリプションファイルがサポートされるようになりました。

### 設定されたバス通信をバス設定に転送

Bus Manager (スタンドアロン) では、設定されたバス通信を 1 つのバス設定から別のバス設定に簡単に転送できるようになりました。これを行うには、バス設定全体をコピーするか、または個々のバス設定要素をコピーまたはカットして、Bus Configurations テーブルの 1 つまたは複数のバス設定に貼り付けることができます。

詳細については、「[バス設定要素のコピー、切り取り、および貼り付け](#)」(『Bus Manager (スタンドアロン) 実装ガイド』[☞](#))を参照してください。

### PDU ゲートウェイのシミュレーション (プレビュー版)

Bus Manager (スタンドアロン) では、PDU ゲートウェイをシミュレートすることができます。PDU ゲートウェイを使用すると、ECU は 1 つまたは複数の通信クラスタから他の通信クラスタに PDU のデータをルーティングできます。PDU ゲートウェイは、通信マトリクスで指定する必要があります。必要に応じて、通信マトリクスを変更することによって、ユーザ定義の PDU ゲートウェイを指定することができます。ただし、

PDU ゲートウェイをシミュレートするには、バス設定の PDU ゲートウェイサポートを有効にする必要があります。

詳細については、「[PDU ゲートウェイのシミュレーション\(プレビューバージョン\)](#)」(『[Bus Manager\(スタンドアロン\)実装ガイド](#)』[📖](#))を参照してください。

#### 注記

PDU ゲートウェイをシミュレートする現在のバージョンはプレビュー版です。Bus Manager プレビュー版の機能の詳細については、「[Bus Manager の概要](#)」(『[Bus Manager\(スタンドアロン\)実装ガイド](#)』[📖](#))を参照してください。

#### LIN PDU の欠落している ECU を生成

Bus Manager(スタンドアロン)は、LIN PDU の欠落している ECU を生成できるようになりました。ECU は、通信マトリクスが送信 ECU または受信 ECU を指定していない場合、または指定された ECU が通信マトリクスの一部でない場合に生成されます。

詳細については、「[通信マトリクスの使用](#)」(『[Bus Manager\(スタンドアロン\)実装ガイド](#)』[📖](#))を参照してください。

#### 通信マトリクスの IPDU にイベント制御タイミングエレメントを追加

Bus Manager(スタンドアロン)では、通信マトリクス内の ISignal IPDU および拡張多重化 IPDU に、欠落したイベント制御タイミングエレメントを追加できるようになりました。これによって、通信マトリクスでまだ指定されていない場合でもイベント制御伝送を指定することができます。

詳細については、「[IPDU へのイベントコントロールタイミングエレメントの追加](#)」(『[Bus Manager\(スタンドアロン\)実装ガイド](#)』[📖](#))を参照してください。

#### 強化された J1939-22 のサポート

Bus Manager(スタンドアロン)では、J1939-22 が次のように拡張されました。

**J1939-22 準拠の IPDU の操作:** Bus Manager(スタンドアロン)では、ペイロード長が 60 バイトまでの J1939-22 準拠の IPDU、つまり Multi-PG プロトコルを使用して送信される IPDU を操作できるようになりました。このような IPDU とその ISignal をバス設定の Manipulation 部分に割り当て、PDU User Code、ISignal Overwrite Value、ISignal Offset Value の操作機能を追加することができます。

さらに、Bus Manager(スタンドアロン)では、新しい Suspend PDU Transmission 操作機能を提供しています。この機能により、Multi-PG プロトコルを使用して送信される J1939-22 準拠の IPDU の送信を一時停止することができます。

- これらの機能の概要については、「[バス設定機能の基礎](#)」(『[Bus Manager\(スタンドアロン\)実装ガイド](#)』[📖](#))を参照してください。
- Suspend PDU Transmission 機能の詳細については、「[PDU 送信の中断](#)」(『[Bus Manager\(スタンドアロン\)実装ガイド](#)』[📖](#))を参照してください。

**保証データのサポート** Bus Manager(スタンドアロン)は、J1939-22 準拠 IPDU の Multi-PG および Broadcast Announce Message (BAM)プロトコルに従って保証データをサポートするようになりました。つまり、Bus Manager(スタンドアロン)では、保証データの送受信が可能になりました。

詳細については、「[J1939 プロトコルのさまざまな側面](#)」(『[Bus Manager\(スタンドアロン\)実装ガイド](#)』[📖](#))を参照してください。

**J1939-22 準拠 IPDU の設定可能な通信マトリクス設定** Bus Manager(スタンドアロン)では、J1939-22 準拠の IPDU のサービスタイプ、トレーラフォーマット、保証データタイプ、保証データサイズを変更できるようになりました。

詳細については、「PDU の指定可能な設定」(『Bus Manager(スタンドアロン)実装ガイド』[📖](#))を参照してください。

#### 多重化 IPDU で利用できる PDU Enable 機能

Bus Manager(スタンドアロン)で、多重化 IPDU に PDU Enable 機能を追加できるようになりました。

詳細については、「PDU の送信の有効化と無効化」(『Bus Manager(スタンドアロン)実装ガイド』[📖](#))を参照してください。

#### Counter Signal 機能が検査に利用可能

Bus Manager(スタンドアロン)では、バス設定の Inspection 部分に割り当てられた ISignal の Counter Signal 機能を利用できるようになりました。

詳細については、「カウンタ信号の使用」(『Bus Manager(スタンドアロン)実装ガイド』[📖](#))を参照してください。

#### パディングバイトの動的レイアウトを持つコンテナ IPDU の未使用ビットパターン

Bus Manager(スタンドアロン)は、関連フレームのパディングバイトの値として、ダイナミックレイアウトのコンテナ IPDU の未使用ビットパターンを使用できるようになりました。必要に応じて、パディングバイトがフレームに追加され、フレーム長が有効な CAN FD 長に延長されます。

詳細については、「サポートされているさまざまな CAN バス機能の側面」(『Bus Manager(スタンドアロン)実装ガイド』[📖](#))を参照してください。

#### Restbus Configuration Exclude List の機能強化

Bus Manager(スタンドアロン)では、ドラッグアンドドロップで ECU を Restbus Configuration Exclude List に追加できるようになりました。この場合、ECU はリスト内でセミコロンで区切られ、レストバス設定の作成時に除外されます。

詳細については、「レストバス設定の作成」(『Bus Manager(スタンドアロン)実装ガイド』[📖](#))を参照してください。

#### [Redesigned Options]ダイアログ

Bus Manager(スタンドアロン)で、完全に設計変更された[Options]ダイアログを利用できるようになりました。[Options]ダイアログでは、カテゴリベースのナビゲーションと、入力中に直ちに反応する検索バーによって、オプションをすばやく見つけることができます。さらに、[Options]ダイアログには新しい Search Path Editor が追加され、Bus Manager(スタンドアロン)で使用されるアーティファクトの検索パスをグラフィカルに設定できるようになりました。

詳細については、「オプション」(『ConfigurationDesk ユーザーインターフェースリファレンス』[📖](#))を参照してください。

#### ツール自動化インターフェースの新機能

ConfigurationDesk の自動化インターフェースにより、追加機能がサポートされています。詳細については、「Release 2023-B における自動化インターフェースの新機能と変更点」(『ConfigurationDesk - ツール操作の自動化』[📖](#))を参照してください。

## Bus Manager(スタンドアロン) 2023-B への移行

TRC ファイルと A2L ファイルの並び順が変更されました。

TRC ファイルおよび A2L ファイルのグループ、ブロック、変数がアルファベット順にソートされるようになりました。その結果、これらのファイルに基づくカスタムツールを調整する必要がある場合があります。

Simulink ルートレベルの入力ポートと出力ポートの新しい ID 動作への移行

Model Interface Package for Simulink は、ルートレベルの Simulink® In、Out、Bus Element In、および Bus Element Out ブロックに新しい ID を割り当てます。その結果、これらの Simulink ブロックに関連するモデルポートへの接続は、ConfigurationDesk アプリケーションでそれぞれの SIC ファイルを更新した後は未解決になります。新しい ID を持つモデルポートを再接続する必要があります。

### ヒント

MATLAB の dsmpb\_pref() API コマンドの新しい UseClassicIdConventionForRootPorts パラメータを True に設定することで、既存のプロジェクトの接続を維持するために古い ID スキーマを復元することができます。

```
dsmpb_pref('Set', 'UseClassicIdConventionForRootPorts', true);
```

ツール自動化インターフェースの変更起因するコードの誤動作

ツール自動化インターフェースの一部の変更により、データモデルに影響が生じ、以前のリリースのコードで誤動作が起きる可能性があります。詳細については、「[Release 2023-B における自動化インターフェースの新機能と変更点](#)」(『ConfigurationDesk - ツール操作の自動化』[📖](#))を参照してください。

## 以前のバージョンで作成されたプロジェクトとアプリケーションの移行

概要

現在の Bus Manager(スタンドアロン)バージョンのプロジェクトを開くのと同一方法で、以前の Bus Manager(スタンドアロン)バージョンで作成したプロジェクト、アプリケーションファイルおよびバックアップを開くことができます。

### 注記

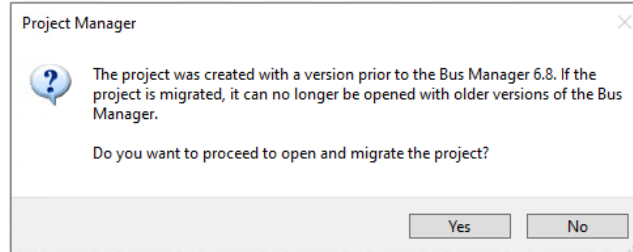
dSPACE Release 2021-A 以降、Bus Manager(スタンドアロン)は、Bus Manager の以前の 7 バージョンのいずれかのバージョンで最後に保存されたプロジェクトに限り、直接インポートをサポートします。

2 つの異なる移行シナリオがあります。

- dSPACE Release 2021-A 以前で作成されたプロジェクトを開く
- dSPACE Release 2021-B 以降で作成されたプロジェクトを開く

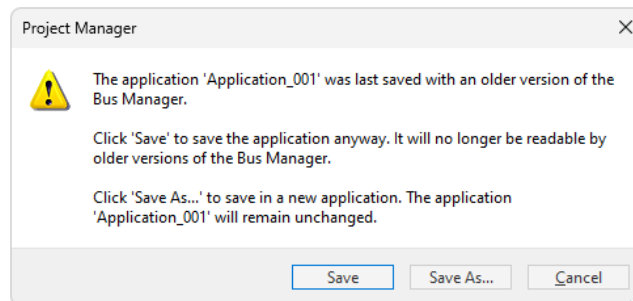
**dSPACE Release 2021-A 以前で作成されたプロジェクトを開く**

Bus Manager (スタンドアロン) 6.8 (dSPACE Release 2021-B) では、プロジェクト構造やプロジェクトとアプリケーションのファイル形式が変更されました。そのため、Bus Manager (スタンドアロン) 6.8 以前で作成されたプロジェクトは、それ以降の Bus Manager (スタンドアロン) バージョンでは使用できません。そのようなプロジェクトを開くと、Bus Manager は以下のように移行の確認を促します。



- Yes をクリックすると、プロジェクトとそのすべてのアプリケーションが新しいプロジェクト構造と新しいファイル形式に移行されます。その後、以前のバージョンの Bus Manager では開くことができなくなります。
- 新しいプロジェクトファイルとアプリケーションファイルは、それぞれプロジェクトフォルダとアプリケーションフォルダに格納されます。これらのファイルは GUID で識別されます。ファイルの移動や名前の変更はしないでください。

Bus Manager (スタンドアロン) 2023-A では、移行したアプリケーションを保存する前に警告ダイアログが表示されるようになりました。次のサンプル図を参照してください。



- [Save] をクリックして、アプリケーションを新しいバージョンの Bus Manager に移行します。その後、以前のバージョンの Bus Manager ではアプリケーションを開くことができなくなります。
- [Save As] をクリックすると、アプリケーションが新しい名前で新しいバージョンの Bus Manager に保存されます。旧バージョンで作成したアプリケーションは、プロジェクトフォルダ内にそのまま残ります。
- 移行したアプリケーションファイルは、それぞれのアプリケーションフォルダに格納されます。これらのファイルは GUID で識別されます。ファイルの移動や名前の変更はしないでください。

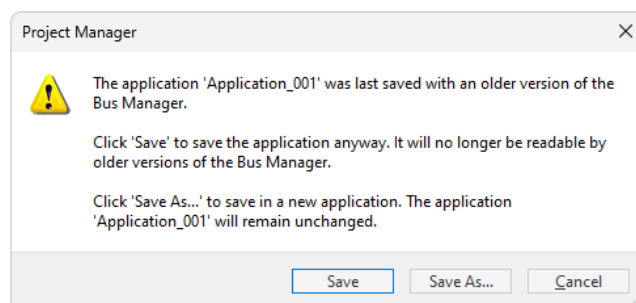
**バージョン管理ソフトウェアの使用** ConfigurationDesk プロジェクトにバージョン管理ソフトウェアを使用している場合、ローカルコピーにチェックインしたときに、古いプロジェクトとアプリケーションのファイルが削除され、新しいファイルがバージョン管理ソフトウェアに自動的に認識されない可能性があることに注意してください。移行後にチェックインしてローカルコピーを削除した場合、古いプロジェクトとアプリケーションファイルを含むプロジェクトを再度チェックアウトして移行を繰り返すことで、新しいプロジェクトとアプリケーションファイルを再作成することができます。プロ

ジェクトファイルとアプリケーションファイルのみが置き換えられ、他の移行手順は繰り返されません。

#### dSPACE Release 2021-B 以降で作成されたプロジェクトを開く

Bus Manager (スタンドアロン) 6.8 (dSPACE Release 2021-B) 以降で作成したプロジェクトを開くと、Bus Manager で警告ダイアログが表示され、移行を確認するよう促されます。

Bus Manager (スタンドアロン) 2023-A では、移行したアプリケーションを保存する前に警告ダイアログが表示され、アプリケーションを開くときには表示されないようになりました。次のサンプル図を参照してください。



- [Save]をクリックして、アプリケーションを新しいバージョンの Bus Manager に移行します。その後、以前のバージョンの Bus Manager ではアプリケーションを開くことができなくなります。
- [Save As]をクリックすると、アプリケーションが新しい名前で新しいバージョンの Bus Manager に保存されます。旧バージョンで作成したアプリケーションは、プロジェクトフォルダ内にそのまま残ります。
- 移行したアプリケーションファイルは、それぞれのアプリケーションフォルダに格納されます。これらのファイルは GUID で識別されます。ファイルの移動や名前の変更はしないでください。

**バージョン管理ソフトウェアの使用** ConfigurationDesk プロジェクトにバージョン管理ソフトウェアを使用している場合、ローカルコピーにチェックインしたときに、古いアプリケーションファイルが削除され、新しいファイルがバージョン管理ソフトウェアに自動的に認識されない可能性があることに注意してください。

移行後にチェックインしてローカルコピーを削除した場合、古いアプリケーションファイルを含むプロジェクトを再度チェックアウトして移行を繰り返すことで、新しいアプリケーションファイルを再作成することができます。アプリケーションファイルのみが置き換えられ、他の移行手順は繰り返されません。



### 移行後のキャッシュフォルダに関する注意事項

#### 注記

Bus Manager (スタンドアロン) 22.1 (dSPACE Release 2022-A) で導入されたキャッシュフォルダについて、以下の点にご注意ください。

- プロジェクトやアプリケーションの要素のパスは 260 文字以内にしてください。「[プロジェクトおよびアプリケーションに関する制限事項](#)」(『[ConfigurationDesk リアルタイム実装ガイド](#)』[📖](#))を参照してください。
- プロジェクト移行時に新しいアプリケーションキャッシュフォルダに移動される要素は、後になって文字数制限を超える可能性があります。これを回避するには、アプリケーションキャッシュのルートフォルダのパスを短く変更する必要があります。詳細については、「[プロジェクト/アプリケーションオプション](#)」(『[ConfigurationDesk ユーザーインターフェースリファレンス](#)』[📖](#))を参照してください。

## Bus Manager (スタンドアロン) の廃止

### ConfigurationDesk アプリケーションの Excel エクスポートの廃止

アクティブな ConfigurationDesk アプリケーションの設定データを Microsoft® Excel™ ファイル (XLSX ファイル) にエクスポートするための [Export Configuration] コマンドは廃止されました。

対応するツール自動化 API コマンド `ICaAlgorithms:ExportConfiguration` を使用すると、No more supported 例外が発生します。



# ConfigurationDesk

**概要** ConfigurationDesk では、SCALEXIO ハードウェア、MicroAutoBox III ハードウェア、または MicroLabBox II ハードウェア用にリアルタイムアプリケーションを実装することができます。

## 次のステップ

### 本章の内容

ConfigurationDesk 2023-B の新しい一般機能.....	67
I/O 機能に関する新機能とハードウェアサポート.....	69
ConfigurationDesk の Bus Manager の新機能.....	74
サポートされるコンテナファイルバージョン.....	76
ConfigurationDesk 2023-B への移行.....	79
以前のバージョンで作成されたプロジェクトとアプリケーションの移行.....	83
ConfigurationDesk の廃止.....	86

## ConfigurationDesk 2023-B の新しい一般機能

### FMU サポートの強化

**バイナリ入出力変数を提供する FMU のサポート** ConfigurationDesk は、FMI 3.0 規格に従ってバイナリ入出力変数を提供する FMU をサポートするようになりました。このような FMU を ConfigurationDesk で使用する場合、ConfigurationDesk はバイナリ入出力変数ごとに可変サイズのポートを作成します。これらのポートは、他の可変サイズのモデルポートや、可変サイズのシグナルをサポートするファンクションポートにマッピングすることができます。

**注記**

UDP Transmit、UDP Receive、TCP ファンクションブロックの Data vector ファンクションポートが、可変サイズのシグナルをサポートするようになりました。「[I/O 機能に関する新機能とハードウェアサポート](#)」(69 ページ)を参照してください。

**ターミナルを提供する FMU のサポート** ConfigurationDesk では、FMI 3.0 および FMI 2.0.4 規格に準拠したターミナルを提供する FMU を使用できるようになりました。ConfigurationDesk アプリケーションに構造化された入力または出力変数を持つターミナルを提供する FMU を追加すると、ConfigurationDesk は各ターミナルのルートエレメントに構造化データの入力ポートまたは構造化データの出力ポートを作成します。これらの構造化ポートは、他の FMU、SIC ファイル、Simulink モデル、BSC ファイル、またはファンクションブロックの構造化ポートとマッピングすることができます。

FMU サポートの詳細については、「[Functional Mock-up Unit \(FMU ファイル\) の使用](#)」(『[ConfigurationDesk リアルタイム実装ガイド](#)』[📖](#))を参照してください。

**マルチコアアプリケーションプロセスのライセンスチェック**

ConfigurationDesk では、1 つより多く(複数)のプロセッサコアを使用するアプリケーションプロセスに対して CFD\_MC ライセンスが必要になりました(必要なプロセッサコア数  $\geq 2$ )。詳細については、「[ライセンスの概要](#)」(『[ConfigurationDesk - 取扱説明書 - 入門編](#)』[📖](#))を参照してください。

**SCALEXIO と MicroLabBox II 間のモデルコンテナの移植性**

新たにサポートされた MicroLabBox II と SCALEXIO は、同じプロセッサアーキテクチャと Linux64 ベースのリアルタイム OS を搭載しています。これにより、両プラットフォームのアプリケーションで同じ SIC、FMU、BSC ファイルを再利用することができます。

**ECU インターフェース処理のための XCP タイムスタンプのサポート強化**

ConfigurationDesk は、ECU インターフェース処理のために XCP タイムスタンプのサポートを強化しました。XCP サービスが統合された ECU アプリケーションがタイムスタンプをサポートしている場合、ECU Interface Configuration ファンクションブロックは、関連する EIC ファイルをインポートするときに新しい TimestampOffset ファンクションポートを提供します。このファンクションポートは、タイムスタンプソースの時刻とリアルタイムハードウェアのシステム時刻の差を提供します。TimestampOffset と関連する TimestampRead ファンクションポートを使用すると、XCP のタイムスタンプをシステム時刻と同期させることができます。

詳細については、「[基本機能の設定 \(ECU Interface Configuration\)](#)」(『[ConfigurationDesk I/O ファンクションインプリメンテーションガイド](#)』[📖](#))を参照してください。

**バス設定とバスシミュレーションコンテナのバスアクセス用の新しいマッピングファイル**

特定の ConfigurationDesk アプリケーションに応じて、ConfigurationDesk はビルドプロセス中に EXPSWCFG.MAPPING ファイルを生成するようになりました。このファイルは、バス設定またはバスシミュレーションコンテナのバスアクセスを指定するために使用される CAN および/または LIN ファンクションブロックを含むアプリケーションプロセスごとに生成されます。各 CAN/LIN ファンクションブロックについて、

EXPSWCFG.MAPPING ファイルには、ファンクションブロックがバスアクセスを指定するすべてのバスアクセス要求へのマッピングが含まれています。

#### ヒント

ControlDesk エクスperimentを作成するとき、ControlDesk は EXPSWCFG.MAPPING ファイルを使用して Bus Navigator ツリーの構造を設定します。その結果、バス設定とバスシミュレーションコンテナで指定されたバス通信について、Bus Navigator ツリーの違いはなくなりました。

#### FPGA コンテナのインポートの改善

FPGA コンテナファイルの場合、FPGA カスタムファンクションをプロジェクトに直接インポートするために、FPGA Custom Functions フォルダがプロジェクトフォルダに自動的に追加されます。

FPGA コンテナをインポートすると、ConfigurationDesk は選択された検索パス内の同じコンテナ ID を持つ既存の FPGA ユーザ関数を自動的に置き換えるようになりました。つまり、検索パスでは最新のバージョンしか利用できなくなります。コンテナ ID は、FPGA Programming Blockset で定義した FPGA アプリケーションの名前です。

#### [Redesigned Options]ダイアログ

完全に再設計された[Options]ダイアログでは、カテゴリベースのナビゲーションと、入力中に直ちに反応する検索バーによって、オプションをすばやく見つけることができます。さらに、[Options]ダイアログに Search Path Editor が追加され、ConfigurationDesk で使用されるアーティファクト(カスタムファンクション、FPGA コンテナ)の検索パスをグラフィカルに設定できるようになりました。

詳細については、「[オプション](#)」(『[ConfigurationDesk ユーザインターフェースリファレンス](#)』[📖](#))を参照してください。

#### ツール自動化インターフェースの新機能

ConfigurationDesk の自動化インターフェースで、ConfigurationDesk の追加機能がサポートされています。詳細については、「[Release 2023-B における自動化インターフェースの新機能と変更点](#)」(『[ConfigurationDesk - ツール操作の自動化](#)』[📖](#))を参照してください。

## I/O 機能に関する新機能とハードウェアサポート

#### 新しくサポートされるハードウェア

**MicroLabBox II** ConfigurationDesk は新しい MicroLabBox II をサポートしています。

MicroLabBox II をサポートするファンクションブロックタイプの概要については、「[ファンクションブロックタイプへの MicroLabBox II チャンネルタイプの割り当て](#)」(『[ConfigurationDesk I/O ファンクションインプリメンテーションガイド](#)』[📖](#))を参照してください。

MicroLabBox の RTI モデルの再利用については、「[Migrating MicroLabBox RTI Models](#)」(『[MicroLabBox II - Getting Started](#)』[📖](#))を参照してください。

**SCALEXIO** ConfigurationDesk は、次の新しい SCALEXIO ハードウェアをサポートしています。

- DS6242 D/A Board  
DS6242 D/A Board は、シングルスロットの SCALEXIO I/O ボードです。電圧出力 (1 チャンネル最大 20 V、3 チャンネル直列最大 60 V) および電流シンク (1 チャンネル最大 40 mA、10 チャンネル並列最大 400 mA) 用に 10 個のガルバニック絶縁アナログチャンネルを提供します。
- DS6365-PE Automotive Ethernet Board  
DS6365-PE Automotive Ethernet Board は、最大 10 Gbit/s の Automotive Ethernet 接続経路で外部デバイスを SCALEXIO プロセッサユニットに接続するための独立した 2 つのポートを備えた PCIe 3.0 x4 ボードです。
- 新しい障害シミュレーションハードウェア：
  - SCALEXIO FSX ユニット  
SCALEXIO FSX ユニットは、FSX ボード用の筐体です。SCALEXIO ラックに組み込むことができ、FSX ボード用のスロット 10 個を備えています。
  - DS6801 Fault Insertion Board  
DS6801 Fault Insertion Board は、IOCNET 100M 通信インターフェースを備えた FSX ボードです。チャンネルあたりの公称負荷スイッチング電流が最大 5A のリレーベースの故障挿入機能を 32 チャンネル備えています。
  - DS6805 Fault Insertion Board  
DS6805 Fault Insertion Board は、IOCNET 100M 通信インターフェースを備えた FSX ボードです。DS6805 は最大 15 A のリレーベースの故障挿入機能を 10 チャンネル備えています。また、故障を切り替えるための中央半導体スイッチも備えています (現在、ソフトウェアではサポートされていません)。

**MicroAutoBox III** ConfigurationDesk は、MACsec 付き Ethernet コントローラを提供する新しい DS1521 Bus Board をサポートしています。MACsec (IEEE802.1AE) は、データリンク層 (レイヤー 2) における Ethernet フレームの認証と暗号化のための規格です。

MACsec の実装については、「Ethernet ポートの MACsec 機能の設定」(『ConfigurationDesk I/O ファンクションインプリメンテーションガイド』[📖](#))を参照してください。

### 拡張されたファンクションブロックタイプ

**UDP および TCP** UDP Transmit、UDP Receive、TCP ファンクションブロックの Data vector ファンクションポートが、可変サイズのシグナルをサポートするようになりました。可変サイズ信号の使用により、データベクトル内の有効バイト数を指定するためのファンクションポートを追加する必要がなくなります。

詳細については、次の項を参照してください。

- 「基本機能の設定 (TCP)」(『ConfigurationDesk I/O ファンクションインプリメンテーションガイド』[📖](#))
- 「基本機能の設定 (UDP Receive)」(『ConfigurationDesk I/O ファンクションインプリメンテーションガイド』[📖](#))
- 「基本機能の設定 (UDP Transmit)」(『ConfigurationDesk I/O ファンクションインプリメンテーションガイド』[📖](#))

**CAN** CAN ファンクションブロックでは、ボーレートとサンプルポイントを指定するソースを選択できるようになりました。ソースとしてファンクションブロックまたはビヘイビアモデルを選択することができます。モデルを選択すると、ボーレートとサン

ルポイントを指定するためのファンクションポートが利用可能になります。この場合、ボーレートとサンプルポイントは実行時に変更することができます。

詳細については、「基本機能の設定 (CAN)」(『ConfigurationDesk I/O ファンクションインプリメンテーションガイド』[📖](#))を参照してください。

#### 新しいファンクションブロックタイプ

**Angular Wavetable Current Sink** Angular Wavetable Current Sink  
ファンクションブロックタイプでは、角度符号化された値でテーブルを参照して、電流信号シーケンスを出力することができます。

詳細については、「Angular Wavetable Current Sink」(『ConfigurationDesk I/O ファンクションインプリメンテーションガイド』[📖](#))を参照してください。

#### 既存のファンクションブロックタイプ向けの新しいチャンネルタイプ

次の表に、新しいチャンネルタイプまたは新しいハードウェアをサポートするファンクションブロックを示します。

ファンクションブロック	新しいチャンネルタイプ	サポートされるハードウェア	詳細情報
Voltage In	Analog In 23	MicroLabBox II	「Voltage In」(『ConfigurationDesk I/O ファンクションインプリメンテーションガイド』 <a href="#">📖</a> )
	Analog In 24		
	Analog In 25		
Current Sink	Analog Out 21	DS6242 D/A Board (SCALEXIO)	「Current Sink」(『ConfigurationDesk I/O ファンクションインプリメンテーションガイド』 <a href="#">📖</a> )
Voltage Out	Analog Out 19	MicroLabBox II	「Voltage Out」(『ConfigurationDesk I/O ファンクションインプリメンテーションガイド』 <a href="#">📖</a> )
	Analog Out 20	DS6242 D/A Board (SCALEXIO)	
	Analog Out 21		
Multi Bit In	Digital In/Out 14	MicroLabBox II	「Multi Bit In」(『ConfigurationDesk I/O ファンクションインプリメンテーションガイド』 <a href="#">📖</a> )
	Digital In/Out 15		
Trigger In	Digital In/Out 14	MicroLabBox II	「Trigger In」(『ConfigurationDesk I/O ファンクションインプリメンテーションガイド』 <a href="#">📖</a> )
	Trigger In 4	DS6242 D/A Board (SCALEXIO)	
Multi Bit Out	Digital In/Out 14	MicroLabBox II	「Multi Bit Out」(『ConfigurationDesk I/O ファンクションインプリメンテーションガイド』 <a href="#">📖</a> )
	Digital In/Out 15		
Digital Pulse Out	Digital In/Out 14	MicroLabBox II	「Digital Pulse Out」(『ConfigurationDesk I/O ファンクションインプリメンテーションガイド』 <a href="#">📖</a> )
PWM/PFM Out	Digital In/Out 14	MicroLabBox II	「PWM/PFM Out」(『ConfigurationDesk I/O ファンクションインプリメンテーションガイド』 <a href="#">📖</a> )

ファンクションブロック	新しいチャンネルタイプ	サポートされるハードウェア	詳細情報
Multi-Channel PWM Out	Digital In/Out 14	MicroLabBox II	「Multi-Channel PWM Out」(『ConfigurationDesk I/O ファンクションインプリメンテーションガイド』 <a href="#">📖</a> )
Digital Pulse Capture	Digital In/Out 14	MicroLabBox II	「Digital Pulse Capture」(『ConfigurationDesk I/O ファンクションインプリメンテーションガイド』 <a href="#">📖</a> )
PWM/PFM In	Digital In/Out 14	MicroLabBox II	「PWM/PFM In」(『ConfigurationDesk I/O ファンクションインプリメンテーションガイド』 <a href="#">📖</a> )
Digital Incremental Encoder In	Digital In/Out 14	MicroLabBox II	「Digital Incremental Encoder In」(『ConfigurationDesk I/O ファンクションインプリメンテーションガイド』 <a href="#">📖</a> )
	Digital In/Out 15		
Sine Encoder In	Analog In 24	MicroLabBox II	「Sine Encoder In」(『ConfigurationDesk I/O ファンクションインプリメンテーションガイド』 <a href="#">📖</a> )
	Analog In 25		
Hall Encoder In	Digital In/Out 14	MicroLabBox II	「Hall Encoder In」(『ConfigurationDesk I/O ファンクションインプリメンテーションガイド』 <a href="#">📖</a> )
Resolver in	Analog In 24	MicroLabBox II	「Resolver In」(『ConfigurationDesk I/O ファンクションインプリメンテーションガイド』 <a href="#">📖</a> )
	Analog In 25		
	Analog Out 20		
EnDat Master	Digital In/Out 15	MicroLabBox II	「EnDat Master」(『ConfigurationDesk I/O ファンクションインプリメンテーションガイド』 <a href="#">📖</a> )
SSI Master	Digital In/Out 14	MicroLabBox II	「SSI Master」(『ConfigurationDesk I/O ファンクションインプリメンテーションガイド』 <a href="#">📖</a> )
	Digital In/Out 15		
Voltage Signal Capture	Analog In 23	MicroLabBox II	「Voltage Signal Capture」(『ConfigurationDesk I/O ファンクションインプリメンテーションガイド』 <a href="#">📖</a> )
	Analog In 24		
	Analog In 25		
Block-Commutated PWM Out	Digital In/Out 14	MicroLabBox II	「Block-Commutated PWM Out」(『ConfigurationDesk I/O ファンクションインプリメンテーションガイド』 <a href="#">📖</a> )
Field-Oriented Control In/Out	Analog In 23 <sup>1)</sup>	MicroLabBox II	「Field-Oriented Control In/Out」(『ConfigurationDesk I/O ファンクションインプリメンテーションガイド』 <a href="#">📖</a> )
	Analog In 24 <sup>1)</sup>		
	Analog In 25 <sup>1)</sup>		
	Digital In/Out 14 <sup>2)</sup>		
Angular Clock Setup	Angle Unit Set	MicroLabBox II	「Angular Clock Setup」(『ConfigurationDesk I/O ファンクションインプリメンテーションガイド』 <a href="#">📖</a> )
SENT In	Digital In/Out 14	MicroLabBox II	「SENT In」(『ConfigurationDesk I/O ファンクションインプリメンテーションガイド』 <a href="#">📖</a> )
CAN	CAN 9	MicroLabBox II	「CAN」(『ConfigurationDesk I/O ファンクションインプリメンテーションガイド』 <a href="#">📖</a> )



ファンクションブ ロック	新しいチャネル タイプ	サポートされる ハードウェア	詳細情報
LIN	LIN 5	MicroLabBox II	「LIN」(『ConfigurationDesk I/O ファンクションインプリメンテーションガイド』 <a href="#">📖</a> )
Ethernet のセット アップ	Ethernet Adapter 1	MicroLabBox II DS6365-PE Automotive Ethernet Board (SCALEXIO)	「Ethernet Setup」(『ConfigurationDesk I/O ファンクションインプリメンテーションガイド』 <a href="#">📖</a> )
Data Streaming Ethernet Setup	Ethernet Adapter 1	MicroLabBox II DS6365-PE Automotive Ethernet Board (SCALEXIO)	「Data Streaming Ethernet Setup」(『ConfigurationDesk I/O ファンクションインプリメンテーションガイド』 <a href="#">📖</a> )
PTP Master	–	MicroLabBox II DS6365-PE Automotive Ethernet Board (SCALEXIO)	「PTP Master」(『ConfigurationDesk I/O ファンクションインプリメンテーションガイド』 <a href="#">📖</a> )
PTP Slave	–	MicroLabBox II DS6365-PE Automotive Ethernet Board (SCALEXIO)	「PTP Slave」(『ConfigurationDesk I/O ファンクションインプリメンテーションガイド』 <a href="#">📖</a> )
Shutdown Request	–	MicroLabBox II	「Shutdown Request」(『ConfigurationDesk I/O ファンクションインプリメンテーションガイド』 <a href="#">📖</a> )
System Shutdown	–	MicroLabBox II	「System Shutdown」(『ConfigurationDesk I/O ファンクションインプリメンテーションガイド』 <a href="#">📖</a> )
System Temperature Monitoring	–	MicroLabBox II	「System Temperature Monitoring」(『ConfigurationDesk I/O ファンクションインプリメンテーションガイド』 <a href="#">📖</a> )
USB Eject	–	MicroLabBox II	「USB Eject」(『ConfigurationDesk I/O ファンクションインプリメンテーションガイド』 <a href="#">📖</a> )
Domain Clock	–	MicroLabBox II	「Domain Clock」(『ConfigurationDesk I/O ファンクションインプリメンテーションガイド』 <a href="#">📖</a> )
Non-Volatile Memory Access	–	MicroLabBox II	「Non-Volatile Memory Access」(『ConfigurationDesk I/O ファンクションインプリメンテーションガイド』 <a href="#">📖</a> )
UART	UART 7	MicroLabBox II	「UART」(『ConfigurationDesk I/O ファンクションインプリメンテーションガイド』 <a href="#">📖</a> )

1) 入力

2) 出力

#### ハードウェアトポロジ用の新しい ファイルフォーマット

ConfigurationDesk は、ハードウェアトポロジを HTFX ファイルに保存し、トポロジのエクスポートとインポートを行うことができます。Release 2023-B では、HTFX ファイルのフォーマットが XML から JSON に変更されました。

設定では、以前の dSPACE Release との互換性を維持するため、従来の XML フォーマットもサポートされています。ハードウェアトポロジを従来のフォーマットでエクスポートするには、Hardware Topology File - Legacy Support ファイルタイプを選択します。

## ConfigurationDesk の Bus Manager の新機能

<p><b>新たにサポートされた AUTOSAR リリース</b></p>	<p>Bus Manager では、AUTOSAR Classic Platform Release R22-11 に基づく AUTOSAR システムデスクリプションファイルがサポートされるようになりました。</p>
<p><b>設定されたバス通信をバス設定に転送</b></p>	<p>Bus Manager では、設定されたバス通信を 1 つのバス設定から別のバス設定に簡単に転送できるようになりました。これを行うには、バス設定全体をコピーするか、または個々のバス設定エレメントをコピーまたはカットして、Bus Configurations テーブルの 1 つまたは複数のバス設定に貼り付けることができます。</p> <p>詳細については、「<a href="#">バス設定エレメントのコピー、切り取り、および貼り付け</a>」(『<a href="#">ConfigurationDesk Bus Manager 実装ガイド</a>』)を参照してください。</p>
<p><b>PDU ゲートウェイのシミュレーション(プレビュー版)</b></p>	<p>Bus Manager では、PDU ゲートウェイをシミュレートすることができます。PDU ゲートウェイを使用すると、ECU は 1 つまたは複数の通信クラスタから他の通信クラスタに PDU のデータをルーティングすることができます。PDU ゲートウェイは、通信マトリクスで指定する必要があります。必要に応じて、通信マトリクスを変更することによって、ユーザ定義の PDU ゲートウェイを指定することができます。ただし、PDU ゲートウェイをシミュレートするには、バス設定の PDU ゲートウェイサポートを有効にする必要があります。</p> <p>詳細については、「<a href="#">PDU ゲートウェイのシミュレーション(プレビューバージョン)</a>」(『<a href="#">ConfigurationDesk Bus Manager 実装ガイド</a>』)を参照してください。</p>
<p><b>注記</b></p> <p>PDU ゲートウェイをシミュレートする現在のバージョンはプレビュー版です。Bus Manager プレビュー版の機能の詳細については、「<a href="#">Bus Manager の概要</a>」(『<a href="#">ConfigurationDesk Bus Manager 実装ガイド</a>』)を参照してください。</p>	
<p><b>LIN PDU の欠落している ECU を生成</b></p>	<p>Bus Manager は、LIN PDU の欠落している ECU を生成できるようになりました。ECU は、通信マトリクスが送信 ECU または受信 ECU を指定していない場合、または指定された ECU が通信マトリクスの一部でない場合に生成されます。</p> <p>詳細については、「<a href="#">通信マトリクスの使用</a>」(『<a href="#">ConfigurationDesk Bus Manager 実装ガイド</a>』)を参照してください。</p>
<p><b>通信マトリクスの IPDU にイベント制御タイミングエレメントを追加</b></p>	<p>Bus Manager では、通信マトリクス内の ISignal IPDU および拡張多重化 IPDU に、欠落したイベント制御タイミングエレメントを追加できるようになりました。これによ</p>

て、通信マトリクスでまだ指定されていない場合でもイベント制御伝送を指定することができます。



詳細については、「[IPDU へのイベントコントロールタイミングエレメントの追加](#)」([『ConfigurationDesk Bus Manager 実装ガイド』](#) )を参照してください。

#### 強化された J1939-22 のサポート


Bus Manager では、J1939-22 が次のように拡張されました。

**J1939-22 準拠の IPDU の操作:** Bus Manager では、ペイロード長が 60 バイトまでの J1939-22 準拠の IPDU、つまり Multi-PG プロトコルを使用して送信される IPDU を操作できるようになりました。このような IPDU とその ISignal をバス設定の Manipulation 部分に割り当て、PDU User Code、ISignal Overwrite Value、ISignal Offset Value の操作機能を追加することができます。

さらに、Bus Manager では、新しい Suspend PDU Transmission 操作機能を提供しています。この機能により、Multi-PG プロトコルを使用して送信される J1939-22 準拠の IPDU の送信を一時停止することができます。

- これらの機能の概要については、「[バス設定機能の基礎](#)」([『ConfigurationDesk Bus Manager 実装ガイド』](#) )を参照してください。
- Suspend PDU Transmission 機能の詳細については、「[PDU 送信の中断](#)」([『ConfigurationDesk Bus Manager 実装ガイド』](#) )を参照してください。

**保証データのサポート** Bus Manager は、J1939-22 準拠 IPDU の Multi-PG および Broadcast Announce Message (BAM)プロトコルに従って保証データをサポートするようになりました。つまり、Bus Manager では、保証データの送受信が可能になりました。


詳細については、「[J1939 プロトコルのさまざまな側面](#)」([『ConfigurationDesk Bus Manager 実装ガイド』](#) )を参照してください。

**J1939-22 準拠 IPDU の設定可能な通信マトリクス設定** Bus Manager では、J1939-22 準拠の IPDU のサービスタイプ、トレーラフォーマット、保証データタイプ、保証データサイズを変更できるようになりました。

詳細については、「[PDU の指定可能な設定](#)」([『ConfigurationDesk Bus Manager 実装ガイド』](#) )を参照してください。

#### 多重化 IPDU で利用できる PDU Enable 機能

Bus Manager で、多重化 IPDU に PDU Enable 機能を追加できるようになりました。

詳細については、「[PDU の送信の有効化と無効化](#)」([『ConfigurationDesk Bus Manager 実装ガイド』](#) )を参照してください。

#### Counter Signal 機能が検査に利用可能

Bus Manager では、バス設定の Inspection 部分に割り当てられた ISignal の Counter Signal 機能を利用できるようになりました。

詳細については、「[カウンタ信号の使用](#)」([『ConfigurationDesk Bus Manager 実装ガイド』](#) )を参照してください。

#### パディングバイトの動的レイアウトを持つコンテナ IPDU の未使用ビットパターン

Bus Manager は、関連フレームのパディングバイトの値として、ダイナミックレイアウトのコンテナ IPDU の未使用ビットパターンを使用するようになりました。必要に応じ

て、パディングバイトがフレームに追加され、フレーム長が有効な CAN FD 長に延長されます。

詳細については、「サポートされているさまざまな CAN バス機能の側面」(『ConfigurationDesk Bus Manager 実装ガイド』[📖](#))を参照してください。

### Restbus Configuration Exclude List の機能強化

Bus Manager では、ドラッグアンドドロップで ECU を Restbus Configuration Exclude List に追加できるようになりました。この場合、ECU はリスト内でセミコロンで区切られ、レストバス設定の作成時に除外されます。

詳細については、「レストバス設定の作成」(『ConfigurationDesk Bus Manager 実装ガイド』[📖](#))を参照してください。

## サポートされるコンテナファイルバージョン

### サポートされる SIC ファイルバージョン

ConfigurationDesk 2023-B(23.2)では、以下にリストする SIC ファイルバージョンがサポートされます。

SIC ファイルを作成した製品 ...	MATLAB Release
dSPACE Release 2023-B: <ul style="list-style-type: none"> <li>Model Interface Package for Simulink 2023-B (23.2)</li> <li>TargetLink 2023-B (23.1)</li> </ul>	<ul style="list-style-type: none"> <li>R2023b</li> <li>R2023a</li> <li>R2022b</li> <li>R2022a</li> </ul>
dSPACE Release 2023-A: <ul style="list-style-type: none"> <li>Model Interface Package for Simulink 23.1</li> </ul>	<ul style="list-style-type: none"> <li>R2023a</li> <li>R2022b</li> <li>R2022a</li> <li>R2021b</li> </ul>
dSPACE Release 2022-B: <ul style="list-style-type: none"> <li>Model Interface Package for Simulink 2022-B (22.2)</li> <li>TargetLink 5.3</li> </ul>	<ul style="list-style-type: none"> <li>R2022b</li> <li>R2022a</li> <li>R2021b</li> <li>R2021a</li> </ul>
dSPACE Release 2022-A: <ul style="list-style-type: none"> <li>Model Interface Package for Simulink 22.1</li> </ul>	<ul style="list-style-type: none"> <li>R2022a</li> <li>R2021b</li> <li>R2021a</li> <li>R2020b</li> </ul>
dSPACE Release 2021-B: <ul style="list-style-type: none"> <li>Model Interface Package for Simulink 4.6</li> <li>TargetLink 5.2</li> </ul>	R2020a
dSPACE Release 2021-A: <ul style="list-style-type: none"> <li>Model Interface Package for Simulink 4.5</li> </ul>	R2019b
dSPACE Release 2020-B: <ul style="list-style-type: none"> <li>Model Interface Package for Simulink 4.4</li> <li>TargetLink 5.1</li> </ul>	R2019a
dSPACE Release 2020-A: <ul style="list-style-type: none"> <li>Model Interface Package for Simulink 4.3</li> </ul>	R2018b
dSPACE Release 2019-B: <ul style="list-style-type: none"> <li>Model Interface Package for Simulink 4.2</li> </ul>	R2018a

SIC ファイルを作成した製品 ...	MATLAB Release
dSPACE Release 2019-A: ▪ Model Interface Package for Simulink 4.1	R2017b
dSPACE Release 2018-B: ▪ Model Interface Package for Simulink 4.0	R2017a
dSPACE Release 2018-A: ▪ Model Interface Package for Simulink 3.6	R2016b
dSPACE Release 2017-B: ▪ Model Interface Package for Simulink 3.5	R2016a
dSPACE Release 2017-A: ▪ Model Interface Package for Simulink 3.4	R2015b

**SIC ファイルのターゲットプラットフォームの互換性** 以下の表は、SIC ファイルとターゲットプラットフォームの互換性を示し、SIC ファイルを生成する際にどのシステムのターゲットファイルを選択する必要があるかを示しています。

SIC ファイルを作成した製品	システムターゲットファイル	Target Platform
dSPACE Release 2022-A 以降	dsrt.tlc	<ul style="list-style-type: none"> <li>▪ SCALEXIO Linux 64 ビット</li> <li>▪ MicroLabBox II</li> <li>▪ MicroAutoBox III</li> </ul> ターゲットプラットフォームに応じて、SIC ファイルを生成するターゲットアーキテクチャを選択する必要があります。詳細については、「 <a href="#">Simulink インプリメンテーションコンテナの基礎</a> 」(『 <a href="#">Model Interface Package for Simulink - モデリングガイド</a> 』(00))を参照してください。
dSPACE Release 2021-B 以前	dsrt64.tlc (dSPACE Release 2019-B 以降で使用可能)	<ul style="list-style-type: none"> <li>▪ SCALEXIO Linux 64 ビット</li> <li>▪ MicroLabBox II</li> </ul>
	dsrt.tlc	MicroAutoBox III

**ConfigurationDesk シナリオでの以前の SIC ファイルバージョンに対する制約** 次の ConfigurationDesk シナリオでは、Model Interface Package for Simulink バージョン 3.4~4.1 で作成した SIC ファイルは、サポートされません。

- マルチモデルアプリケーションプロセス。
- Real-Time Testing を使用するリアルタイムアプリケーションをビルドする場合。

**Simulink ビヘイビアモデルの以前の基本 SIC ファイルバージョンの制約** Model Interface Package for Simulink バージョン 3.4~4.4 で作成した Simulink ビヘイビアモデルの基本 SIC ファイルには、次の制約が適用されます。

- Simulink ビヘイビアモデルには、次のブロックセットのブロックを含めないでください。
  - ASM のブロック  
このような SIC ファイルを含むリアルタイムアプリケーションは、ModelDesk と使用することはできません。
  - FPGA Programming Blockset
  - MotionDesk Blockset  
このような SIC ファイルを含むリアルタイムアプリケーションは、MotionDesk と使用することはできません。
  - dSPACE ソリューションのブロック

- SIC ファイルに含まれるコンパイル済みオブジェクトは、ターゲットプラットフォームと一致している必要があります。現在のターゲットプラットフォームが存在しない時期に作成されたコンパイル済みオブジェクトを含む SIC ファイルはサポートされていません。この場合、コンパイル済みオブジェクトを使用することはできません。

#### サポートされる BSC ファイルバージョン

ConfigurationDesk 2023-B(23.2)は、最新リリースの Bus Manager で生成された BSC ファイル(つまり BSC ファイルバージョン 1.12)または Ethernet Configuration Package で生成された BSC ファイルをサポートします。サポートされる Ethernet BSC ファイルのバージョンの詳細については、<http://www.dspace.jp/go/ethcpinfo> を参照してください。

#### サポートされる Functional Mock-Up Unit(FMU)

ConfigurationDesk 2023-B(23.2)は、以下のバージョンの FMI 規格に準拠した Functional Mock-Up Units(FMU)をサポートします。

- 2.0.x
- 3.0.x

#### サポートされる EIC ファイルバージョン

ConfigurationDesk 2023-B(23.2)は、以下にリストする EIC ファイルバージョンをサポートします。

EIC ファイルを作成した製品 ...	EIC のバージョン
dSPACE Release 2023-B (ECU Interface Manager 2023-B)	5.0.0
dSPACE Release 2023-A (ECU Interface Manager 2023-A)	5.0.0
dSPACE Release 2022-B (ECU Interface Manager 2022-B)	5.0.0
dSPACE Release 2022-A (ECU Interface Manager 2.11)	5.0.0
dSPACE Release 2021-B (ECU Interface Manager 2.10)	4.0.0
dSPACE Release 2021-A (ECU Interface Manager 2.9)	4.0.0
dSPACE Release 2020-B (ECU Interface Manager 2.8)	4.0.0
dSPACE Release 2020-A (ECU Interface Manager 2.7)	4.0.0
dSPACE Release 2019-B (ECU Interface Manager 2.6)	4.0.0
dSPACE Release 2019-A (ECU Interface Manager 2.5)	3.0.0
dSPACE Release 2018-B (ECU Interface Manager 2.4)	3.0.0
dSPACE Release 2018-A (ECU Interface Manager 2.3)	2.0.0

EIC ファイルを作成した製品 ...	EIC のバージョン
dSPACE Release 2017-B (ECU Interface Manager 2.2)	1.0.0
dSPACE Release 2017-A (ECU Interface Manager 2.1)	1.0.0
dSPACE Release 2016-B (ECU Interface Manager 2.0p1)	1.0.0

**注記**

- 64 ビットターゲットアーキテクチャでは、バージョン 5.0.0 以降の EIC ファイルのみサポートされます。
- MicroAutoBox III システムでは、バージョン 4.0.0 以降の EIC ファイルのみをサポートされます。

## ConfigurationDesk 2023-B への移行

### インストールパスの変更

dSPACE Release 2023-A では、ConfigurationDesk は、dSPACE ConfigurationDesk <ReleaseVersion>という名前の別のフォルダにインストールされます。つまり、ConfigurationDesk は RCP や HIL のインストールフォルダに含まれなくなりました。

ConfigurationDesk のインストール場所に依存するスクリプトがある場合は、そのスクリプトを調整する必要があります。

### TRC ファイルと A2L ファイルの並び順が変更されました。

TRC ファイルおよび A2L ファイルのグループ、ブロック、変数がアルファベット順にソートされるようになりました。その結果、これらのファイルに基づくカスタムツールを調整する必要がある場合があります。

### Simulink ルートレベルの入力ポートと出力ポートの新しい ID 動作への移行

Model Interface Package for Simulink は、ルートレベルの Simulink® In、Out、Bus Element In、および Bus Element Out ブロックに新しい ID を割り当てます。その結果、これらの Simulink ブロックに関連する ConfigurationDesk または VEOS プロジェクトのモデルポートへのすべての接続は、モデルの解析またはそれぞれの SIC ファイルの更新後に未解決になります。新しい ID を持つモデルポートを再接続する必要があります。

**ヒント**

MATLAB の `dsmpb_pref()` API コマンドの新しい `UseClassicIdConventionForRootPorts` パラメータを `True` に設定することで、既存のプロジェクトの接続を維持するために古い ID スキーマを復元できます。

```
dsmpb_pref('Set', 'UseClassicIdConventionForRootPorts', true);
```



## ターゲットアーキテクチャ設定の変更

SCALEXIO 32-bit Linux ターゲットプラットフォームのサポートは、ConfigurationDesk 2023-B (23.2)から廃止になりました。これにより、以下の変更があります。

- デフォルトターゲットアーキテクチャを指定するための SCALEXIO target architecture プロパティは、[Options]ダイアログの[Configuration]ページから削除されました。
- Global Build Settings の Target architecture プロパティは、読み取り専用になりました。割り当てられたプロセッサユニットに応じて、以下のいずれかの値が表示されます。
  - 64-bit (SCALEXIO および MicroLabBox II の場合)
  - 32-bit (MicroAutoBox III の場合)
  - 該当なし (プロセッサユニットが割り当てられていない場合)
- モデルコンテナのプリコンパイルおよび自動化コマンドで、SCALEXIO\_LNX ターゲットプラットフォーム識別子がサポートされなくなりました。


SCALEXIO 64-bit Linux ターゲットプラットフォーム (SCALEXIO\_LNX64) は、利用することができます。

### 注記

32 ビット互換のバイナリを含むモデルインプリメンテーションコンテナは、SCALEXIO Linux または MicroLabBox II 64 ビットターゲットアーキテクチャと互換性がありません。モデルインプリメンテーションコンテナは、Linux 64 ビット互換のバイナリで再度生成する必要があります。


SIC ファイルは、Simulink [Configuration Parameters]ダイアログで適切な [Target architecture] を設定するか、dsrt\_build() API コマンドを使用して再生成する必要があります。

## ツール自動化インターフェースの変更に関連するコードの誤動作

ツール自動化インターフェースの一部の変更により、データモデルに影響が生じ、以前のリリースのコードで誤動作が起きる可能性があります。詳細については、「[Release 2023-B における自動化インターフェースの新機能と変更点](#)」(『[ConfigurationDesk - ツール操作の自動化](#)』)を参照してください。

## カスタムファンクションフォルダに関する変更

dSPACE Release 2022-A 以降、プロジェクト固有のカスタムファンクションフォルダ <ProjectLocation>\<ProjectFolder>\CustomFunctions は新規プロジェクトに自動的に追加されません。プロジェクト固有のカスタムファンクションフォルダがない場合、カスタムファンクションをそのフォルダにコピーするオートメーションスクリプトの実行に影響を与える可能性があります。

新規プロジェクトでは、カスタムファンクションを追加するために検索パスを使用することをお勧めします。検索パスを使用すると、たとえばリポジトリやバージョン管理ソフトなどを利用することができます。詳細については、「[検索パスとユーザーセッションファイルの基礎](#)」(『[ConfigurationDesk カスタム I/O ファンクションインプリメンテーションガイド](#)』)を参照してください。

ただし、ConfigurationDesk は、プロジェクト固有のカスタムファンクションフォルダを引き続きサポートしています。



**デジタル出力信号生成の設定の変更**

dSPACE Release 2023-A では、[High reference potential]プロパティの設定が変更されています。旧バージョンの ConfigurationDesk で作成されたプロジェクトを開くと、次の表のように設定が移行されます。

関連するチャンネルタイプごとに、移行前の設定と、移行後に新たにマッピングされた設定を表に示します。設定値は、自動化のために整数値（括弧内）で表示されず。

**注記**

マッピングされた新しい設定が適用可能かどうかを確認する必要があります。必要な場合、お客様の要件に応じて設定を変更する必要があります。

以前の設定 チャンネルタイプ	Individual (1)	VBat (2)	Shared (4)	Shared 2 (5)
ハードウェアリソースの割り当てなし	External individual (1)	External shared (2)	External shared (2)	External shared 2 (3)
Flexible Out 1	External individual (1)	External individual (1)	External individual (1)	External individual (1)
Digital Out 1	External individual (1)	Internal shared (4)	External shared (2)	External shared (2)
Digital Out 2	External individual (1)	External individual (1)	External individual (1)	External individual (1)
Digital Out 3	External individual (1)	External shared (2)	External shared (2)	External shared 2 (3)
Digital Out 4	External individual (1)	External shared (2)	External shared (2)	External shared (2)
Digital Out 5	External individual (1)	External shared (2)	External shared (2)	External shared (2)
Digital Out 7	External individual (1)	External shared (2)	External shared (2)	External shared (2)
Digital Out 8	External individual (1)	External shared (2)	Internal shared (4)	Internal shared 2 (5)
Digital In/Out 1	External individual (1)	External individual (1)	External individual (1)	External individual (1)
Digital In/Out 3	External individual (1)	External shared (2)	External shared (2)	External shared 2 (3)
Digital In/Out 5	External individual (1)	External shared (2)	Internal shared (4)	Internal shared 2 (5)
Digital In/Out 6	External individual (1)	External shared (2)	Internal shared (4)	Internal shared (4)
Digital In/Out 8	External individual (1)	External shared (2)	Internal shared (4)	Internal shared (4)
Digital In/Out 9	External individual (1)	External shared (2)	Internal shared (4)	Internal shared 2 (5)
Digital In/Out 10	External individual (1)	Internal shared 2 (5)	Internal shared (4)	Internal shared 2 (5)

チャンネルタイプが選択した設定に対応していない場合は競合が発生し、それが表示されます。たとえば、Flexible Out 1 チャンネルタイプは、External individual 設定のみをサポートします。それ以外の設定の場合、競合が発生します。

**新しい設定の説明**

設定	サポートされるチャンネルタイプ	説明
External individual	<ul style="list-style-type: none"> <li>Flexible Out 1 (DS2621)</li> </ul>	Flexible Out 1 チャンネルタイプのチャンネルは、dSPACE リアルタイムハードウェアのグラウンド電位に対してガルバニック絶縁されています。各チャンネルには固有の電子接続があり、他のチャンネルから独立しています。これにより、電位がシフトした信号と接続することができます。 電位を High Reference シグナルポートに接続することができます。

設定	サポートされるチャンネルタイプ	説明
	<ul style="list-style-type: none"> <li>▪ Digital Out 2 (DS2690)</li> <li>▪ Digital In/Out 1 (DS2690)</li> </ul>	<p>チャンネルタイプの各チャンネルでは、独自の電気的高基準電位を提供します。そのため、各チャンネルは互いに独立しています。これにより、個別の任意の電位と接続することができます。</p> <p>電位を High Reference シグナルポートに接続することができます。</p>
External shared	<ul style="list-style-type: none"> <li>▪ Digital Out 1 (DS2680)</li> <li>▪ Digital Out 3 (DS6101)</li> <li>▪ Digital Out 4 (MircoAutoBox III)</li> <li>▪ Digital Out 5 (MircoAutoBox III)</li> <li>▪ Digital Out 7 (MircoAutoBox III)</li> <li>▪ Digital In/Out 3 (DS6201)</li> </ul>	<p>これらのチャンネルタイプでは、接続された電位をチャンネルセットに属する他のすべてのデジタル出力チャンネルと共有します。したがって、電位は I/O コネクタの単一のピンで利用可能です。</p> <p>電位を High Reference シグナルポートに接続することができます。</p>
External shared 2	<ul style="list-style-type: none"> <li>▪ Digital Out 3 (DS6101)</li> <li>▪ Digital In/Out 3 (DS6201)</li> </ul>	<p>これらのチャンネルタイプでは、接続された電位をチャンネルセットに属する他のすべてのデジタル出力チャンネルと共有します。したがって、電位は I/O コネクタの単一のピンで利用可能です。この電位 (External shared 2) を共有するコネクタピンは、External shared 設定で使用されたピンと同一ではありません。</p> <p>電位を High Reference シグナルポートに接続することができます。</p>
Internal shared	<ul style="list-style-type: none"> <li>▪ Digital Out 8 (DS6121)</li> <li>▪ Digital In/Out 5 (DS6202)</li> <li>▪ Digital /Out 6 (MircoAutoBox III)</li> <li>▪ Digital /Out 8 (MircoAutoBox III)</li> <li>▪ Digital In/Out 9 (DS6121)</li> <li>▪ Digital /Out 10 (MircoAutoBox III)</li> </ul>	<p>dSPACE リアルタイムハードウェアの 5V 内部電源は、高基準電位として使用されます。</p> <p>High Reference シグナルポートを使用することはできません。</p>
	<ul style="list-style-type: none"> <li>▪ Digital Out 1 (DS2680)</li> </ul>	<p>dSPACE リアルタイムシステムの内部 VBAT (シミュレートされるバッテリー電圧) は、高基準電位として使用されます。</p> <p>High Reference シグナルポートを使用することはできません。</p> <p>シミュレートされる SCALEXIO システムの電圧は、DS2680 に接続されたバッテリーシミュレーション電源ユニットから供給されます。電圧範囲は、使用する電源によって異なります。「<a href="#">バッテリーシミュレーション電源ユニット</a>」(『<a href="#">SCALEXIO ハードウェアの設置および設定</a>』( ))を参照してください。</p>
Internal shared 2	<ul style="list-style-type: none"> <li>▪ Digital Out 8 (DS6121)</li> <li>▪ Digital In/Out 5 (DS6202)</li> <li>▪ Digital In/Out 9 (DS6121)</li> </ul>	<p>dSPACE リアルタイムハードウェアの 3.3V 内部電源は、高基準電位として使用されます。</p> <p>High Reference シグナルポートを使用することはできません。</p>

設定	サポートされるチャンネルタイプ	説明
	<ul style="list-style-type: none"> <li>Digital In/Out 10 (MircoAutoBox III)</li> </ul>	<p>dSPACE リアルタイムハードウェアの内部 VBAT は、高基準電位として使用されます。</p> <p>High Reference シグナルポートを使用することはできません。</p> <p>MicroAutoBox III の内部 VBAT は、MicroAutoBox III の動作電圧に対応しません。値については、「一般特性」(『MicroAutoBox III ハードウェアの設置および設定』<a href="#">📖</a>)を参照してください。</p>

## 以前のバージョンで作成されたプロジェクトとアプリケーションの移行

### 概要

現在の ConfigurationDesk のバージョンのプロジェクトを開くのと同一方法で、以前の ConfigurationDesk のバージョンで作成したプロジェクト、アプリケーションファイルおよびバックアップを開くことができます。

#### 注記

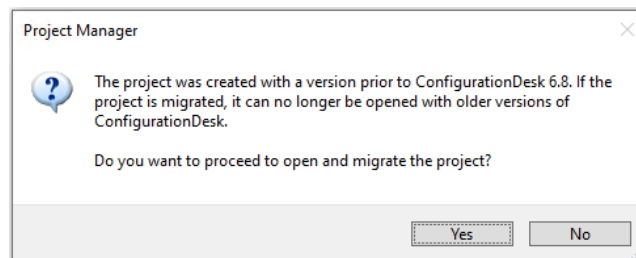
dSPACE Release 2021-A 以降、ConfigurationDesk は、ConfigurationDesk の以前の 7 バージョンのいずれかのバージョンで最後に保存されたプロジェクトのみの直接インポートをサポートします。

2 つの異なる移行シナリオがあります。

- dSPACE Release 2021-A 以前で作成されたプロジェクトを開く
- dSPACE Release 2021-B 以降で作成されたプロジェクトを開く

### dSPACE Release 2021-A 以前で作成されたプロジェクトを開く

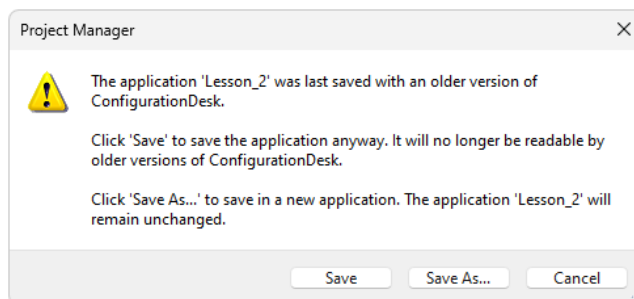
ConfigurationDesk 6.8 (dSPACE Release 2021-B) では、プロジェクト構造やプロジェクトとアプリケーションのファイル形式が変更されました。したがって、ConfigurationDesk 6.8 以前のバージョンで作成されたプロジェクトは、それ以降のバージョンの ConfigurationDesk では使用することができません。そのようなプロジェクトを開くと、ConfigurationDesk は以下のように移行の確認を促します。



- Yes をクリックすると、プロジェクトとそのすべてのアプリケーションが新しいプロジェクト構造と新しいファイル形式に移行されます。その後、以前のバージョンの ConfigurationDesk では開くことができなくなります。

- 新しいプロジェクトファイルとアプリケーションファイルは、それぞれプロジェクトフォルダとアプリケーションフォルダに格納されます。これらのファイルは GUID で識別されます。ファイルの移動や名前の変更はしないでください。

ConfigurationDesk 2023-A では、移行した ConfigurationDesk アプリケーションを保存する前に警告ダイアログが表示されます。次のサンプル図を参照してください。



- [Save]をクリックして、アプリケーションを新しいバージョンの ConfigurationDesk に移行します。その後、以前のバージョンの ConfigurationDesk ではアプリケーションを開くことができなくなります。
- [Save As]をクリックすると、アプリケーションが新しい名前で新しいバージョンの ConfigurationDesk に保存されます。旧バージョンで作成したアプリケーションは、プロジェクトフォルダ内にそのまま残ります。
- 移行したアプリケーションファイルは、それぞれのアプリケーションフォルダに格納されます。これらのファイルは GUID で識別されます。ファイルの移動や名前の変更はしないでください。

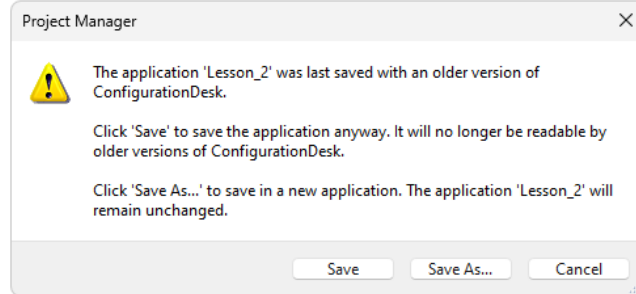
**バージョン管理ソフトウェアの使用** ConfigurationDesk プロジェクトにバージョン管理ソフトウェアを使用している場合、ローカルコピーにチェックインしたときに、古いプロジェクトとアプリケーションのファイルが削除され、新しいファイルがバージョン管理ソフトウェアに自動的に認識されない可能性があることに注意してください。

移行後にチェックインしてローカルコピーを削除した場合、古いプロジェクトとアプリケーションファイルを含むプロジェクトを再度チェックアウトして移行を繰り返すことで、新しいプロジェクトとアプリケーションファイルを再作成することができます。プロジェクトファイルとアプリケーションファイルのみが置き換えられ、他の移行手順は繰り返されません。

**dSPACE Release 2021-B 以降で作成されたプロジェクトを開く**

ConfigurationDesk 6.8(dSPACE Release 2021-B) 以降で作成したプロジェクトを開くと、ConfigurationDesk で警告ダイアログが表示され、移行を確認するよう促されます。

ConfigurationDesk 2023-A では、移行した ConfigurationDesk アプリケーションを保存する前に警告ダイアログが表示され、アプリケーションを開くときには表示されないようになりました。次のサンプル図を参照してください。



- [Save]をクリックして、アプリケーションを新しいバージョンの ConfigurationDesk に移行します。その後、以前のバージョンの ConfigurationDesk ではアプリケーションを開くことができなくなります。
- [Save As]をクリックすると、アプリケーションが新しい名前で新しいバージョンの ConfigurationDesk に保存されます。旧バージョンで作成したアプリケーションは、プロジェクトフォルダ内にそのまま残ります。
- 移行したアプリケーションファイルは、それぞれのアプリケーションフォルダに格納されます。これらのファイルは GUID で識別されます。ファイルの移動や名前の変更はしないでください。



**バージョン管理ソフトウェアの使用** ConfigurationDesk プロジェクトにバージョン管理ソフトウェアを使用している場合、ローカルコピーにチェックインしたときに、古いアプリケーションファイルが削除され、新しいファイルがバージョン管理ソフトウェアに自動的に認識されない可能性があることに注意してください。

移行後にチェックインしてローカルコピーを削除した場合、古いアプリケーションファイルを含むプロジェクトを再度チェックアウトして移行を繰り返すことで、新しいアプリケーションファイルを再作成することができます。アプリケーションファイルのみが置き換えられ、他の移行手順は繰り返されません。

#### 移行後のキャッシュフォルダに関する注意事項

##### 注記

ConfigurationDesk 22.1 (dSPACE Release 2022-A) で導入されたキャッシュフォルダについて、次の点に注意してください。

- プロジェクトやアプリケーションの要素のパスは 260 文字以内にしてください。「[プロジェクトおよびアプリケーションに関する制限事項](#)」(『[ConfigurationDesk リアルタイム実装ガイド](#)』)を参照してください。
- プロジェクト移行時に新しいアプリケーションキャッシュフォルダに移動されるビルドアーティファクトなどの要素は、後になって文字数制限を超える可能性があります。これを回避するには、アプリケーションキャッシュのルートフォルダのパスを短く変更する必要があります。詳細については、「[プロジェクト/アプリケーションオプション](#)」(『[ConfigurationDesk ユーザーインターフェースリファレンス](#)』)を参照してください。

## ConfigurationDesk の廃止

---

<b>SCALEXIO 32-bit Linux ターゲットプラットフォームの廃止</b>	SCALEXIO 32-bit Linux ターゲットプラットフォームのサポートは、ConfigurationDesk 2023-B (23.2)から廃止になりました。ターゲットアーキテクチャの設定に関する移行に関する注意事項については、「 <a href="#">ConfigurationDesk 2023-B への移行</a> 」(79 ページ)を参照してください。
<b>ConfigurationDesk アプリケーションの Excel エクスポートの廃止</b>	アクティブな ConfigurationDesk アプリケーションの設定データを Microsoft® Excel™ファイル (XLSX ファイル)にエクスポートするための[Export Configuration]コマンドは、ConfigurationDesk 2023-B (23.2)で廃止されました。  対応するツール自動化 API コマンド <code>ICaAlgorithms:ExportConfiguration</code> を使用すると、No more supported 例外が発生します。
<b>デバイストポロジを含む XLS ファイルの廃止</b>	ConfigurationDesk 2023-A (23.1)では、DTFX ファイルおよび Microsoft Excel™ XLSX ファイルからのみ外部デバイスのトポロジをインポートすることができます。
<b>カスタムデバイスプロパティの廃止</b>	カスタムデバイスプロパティをデバイストポロジの要素に追加する機能は、ConfigurationDesk 2023-B (23.2)で削除されました。
<b>外部ケーブルハーネスの計算の廃止予定</b>	外部ケーブルハーネスの表現を計算する機能は、ConfigurationDesk 2024-A で削除されます。
<b>SCALEXIO 向け RTI バス通信ブロックセットの廃止予定</b>	SCALEXIO 用の RTI CAN MultiMessage Blockset と RTI LIN MultiMessage Blockset のサポートは、ConfigurationDesk 2025-A で廃止される予定です。SCALEXIO 用の新しい ConfigurationDesk プロジェクトでは、これらのブロックセットのブロックを含むモデルを使用しないことをお勧めします。

---

# ControlDesk

---

## 次のステップ

## 本章の内容

ControlDesk 2023-B の新機能.....	88
ControlDesk 2023-B への移行.....	104

## ControlDesk 2023-B の新機能

### 次のステップ

### 本章の内容

プラットフォーム管理およびプラットフォーム／デバイスの新機能 (ControlDesk 2023-B).....	88
ControlDesk 2023-B のプラットフォーム管理およびプラットフォーム／デバイスの新機能について概要を説明します。	
新しい変数管理機能 (ControlDesk 2023-B).....	89
ControlDesk 2023-B の新しい変数管理機能について概要を説明します。	
新しい計器機能 (ControlDesk 2023-B).....	90
ControlDesk 2023-B の計器の新機能の概要について説明します。	
新しい Bus Navigator 機能 (ControlDesk 2023-B).....	93
ControlDesk 2023-B の新しい Bus Navigator 機能について概要を説明します。	
新しい電氣的欠陥シミュレーション機能 (ControlDesk 2023-B).....	101
ControlDesk 2023-B 以降の新しい電氣的欠陥シミュレーション機能の概要について説明します。	
新しいユーザインターフェース処理機能 (ControlDesk 2023-B).....	101
ControlDesk 2023-B の新しいユーザインターフェース処理機能について概要を説明します。	

## プラットフォーム管理およびプラットフォーム／デバイスの新機能 (ControlDesk 2023-B)

### 新しい MicroLabBox II のプラットフォームサポート

ControlDesk は、新しい MicroAutoBox II をサポートする MicroLabBox II プラットフォームを備えています。

「MicroLabBox II Platform」(『ControlDesk プラットフォーム管理』[📖](#))を参照してください。

### SCALEXIO プラットフォーム: DS6242 D/A Board および DS6365-PE Automotive Ethernet Board のサポート

ControlDesk では、次の新しい SCALEXIO ボードがサポートされるようになりました。

- DS6242 D/A Board
- DS6365-PE Automotive Ethernet Board



## GNSS デバイス:新しいステータス変数(使用中の衛星数の表示など)

ControlDesk 2023-B では、GNSS デバイスが、以下の図のように Status 変数グループを提供します。

Group	Description	Favorite	Var Co	Variable	Block	Platform/Device	Description	Unit	Type
All Variable Descriptions		<input type="checkbox"/>	<input type="checkbox"/>	Quality	Status	GNSS	Quality of the satellite fix.		Signed 64 bit
Calculated Variables		<input type="checkbox"/>	<input type="checkbox"/>	Satellites	Status	GNSS	Number of satellites in use.		Signed 64 bit
GNSS Data		<input type="checkbox"/>	<input type="checkbox"/>	Status	Status	GNSS	Status of the receiver.		Signed 64 bit
Date									
Position									
Speed									
Status									
Time									
VEOS_demo.sdf									

変数	目的
Quality	現在の衛星フィックス、つまり GNSS 受信機が位置を決定するために使用している信号のタイプや技術を表示します。したがって、衛星フィックスは、信号の品質、つまり報告される位置の正確さと信頼性を示しています。
Satellites	GNSS 受信機が現在使用している衛星の数を表示します。
Status	接続された GNSS 受信機が有効なデータを受信しているかどうかを表示します。

## ハードウェアトポロジのエクスポート

MicroAutoBox III、MicroLabBox II、および SCALEXIO プラットフォームでは、ControlDesk でハードウェアトポロジをエクスポートして、ConfigurationDesk アプリケーションで再利用できるようになりました。




「Export Hardware Topology」(『ControlDesk プラットフォーム管理』)を参照してください。

## 新しい変数管理機能 (ControlDesk 2023-B)

### 変数マッピングの改善

バージョン 2023-A で導入された ASAM XIL フレームワークマッピングコンセプトの識別子マッピングによる変数マッピングのための ControlDesk のサポートが改善されました。

**レイアウト上のマッピングと変数の置き換え(およびその逆の操作)** レイアウト上で計器に接続された解決可能なマッピングを関連する変数に置き換えること(およびその逆の操作)ができるようになりました。アクティブレイアウト、開いているすべてのレイアウト、実験のすべてのレイアウトに対して個別に実行することができます。

「Replace Variables with Mappings」(『ControlDesk レイアウト』)、  
「Replace Mappings with Variables」(『ControlDesk レイアウト』)、および  
「LayoutManagement / IXaLayoutManagement <<Interface>>」(『ControlDesk 自動化』)を参照してください。

**RT アプリケーションデモは変数マッピングを使用** ControlDesk のリアルタイムアプリケーションのデモプロジェクトでは、変数マッピングを使用するようになりました。デモ実験のレイアウトと計器は、フレームワークラベルに基づいています。

「リアルタイムアプリケーションのデモ」(『ControlDesk 概要』📖)を参照してください。

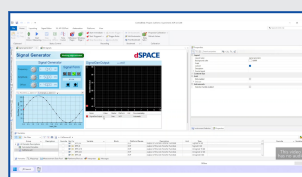
## 新しい計器機能(ControlDesk 2023-B)

**時間プロッタ、インデックスプロッタ: 凡例にシグナル名を表示**

ControlDesk では、時間プロッタおよびインデックスプロッタの凡例にシグナル名を表示できるようになりました。

### 時間プロッタ: 凡例に信号名を表示する

この動画では、時間プロッタの凡例に信号名を表示する方法を紹介します。インデックスプロッタの手順も同様です。



このビデオを見るには、以下のリンクをクリックするか、QR コードを読み取ってください。 <https://www.dspace.com/dspace-help/At2ub>

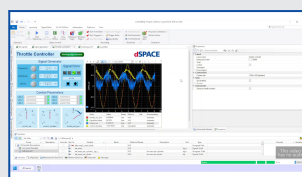
「Column Chooser (Time Plotter/Index Plotter)」(『ControlDesk 計器の操作』📖)を参照してください。

**時間プロッタ: 時間およびデータカーソルのツールチップに接続名を表示**

ControlDesk では、時間プロッタの時間およびデータカーソルのツールチップに接続された変数の名前を表示できるようになりました。

### 時間プロッタ: 時間およびデータカーソルのツールチップに接続名を表示する

この動画では、時間プロッタの時間およびデータカーソルのツールチップに接続名を表示する方法を紹介します。



このビデオを見るには、以下のリンクをクリックするか、QR コードを読み取ってください。 <https://www.dspace.com/dspace-help/uahmq>

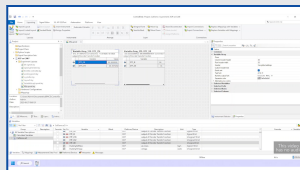
「Time Cursor Properties」(『ControlDesk 計器の操作』📖)および「Data Cursor Properties (Time Plotter/Index Plotter)」(『ControlDesk 計器の操作』📖)を参照してください。

### バリエブルアレイ:カスタムヘッダーテキストの指定

ControlDesk では、バリエブルアレイのヘッダーにカスタムテキストを指定できるようになりました。これはたとえば、変数のグループを複数のバリエブルアレイに分割し、すべての計器にグループ名を使用する場合などに便利です。

#### バリエブルアレイ:カスタムヘッダーテキストの指定

この動画では、バリエブルアレイのカスタムヘッダーテキストを[Properties]ペインと IVariableArrayInstrument 自動化インターフェースで指定する方法を紹介します。



このビデオを見るには、以下のリンクをクリックするか、QR コードを読み取ってください。 <https://www.dspace.com/dspace-help/a53Kp>

「Custom Header Text Property (Variable Array) (『ControlDesk 計器の操作』[🔗](#)) および「VariableArrayInstrument / IVariableArrayInstrument <<Interface>>」 (『ControlDesk 自動化』[🔗](#))を参照してください。

### 音声コントローラ:MP3 ファイルのサポート

ControlDesk 音声コントローラは、MP3 音声ファイルもサポートするようになりました。

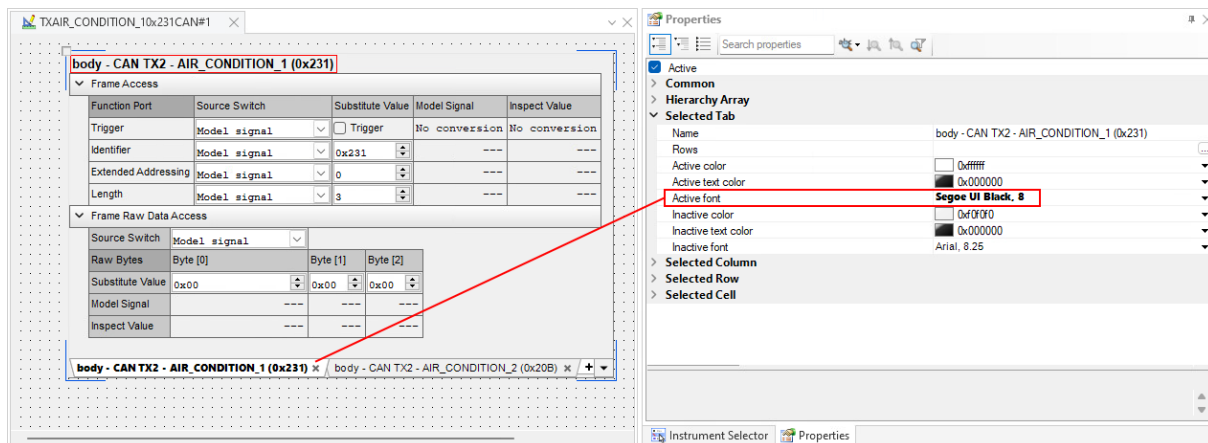
「Sound Controller Properties」 (『ControlDesk 計器の操作』[🔗](#))および「SoundController / IVariableArrayInstrument <<Interface>>」 (『ControlDesk 自動化』[🔗](#))を参照してください。

### 階層配列:ユーザビリティの改善

階層配列が改善されました。

**タブヘッダーテキストのフォントをカスタマイズする** タブヘッダーのテキストのフォントをカスタマイズできるようになりました。アクティブなタブと非アクティブなタブのヘッダーのフォントを個別にカスタマイズすることができます。

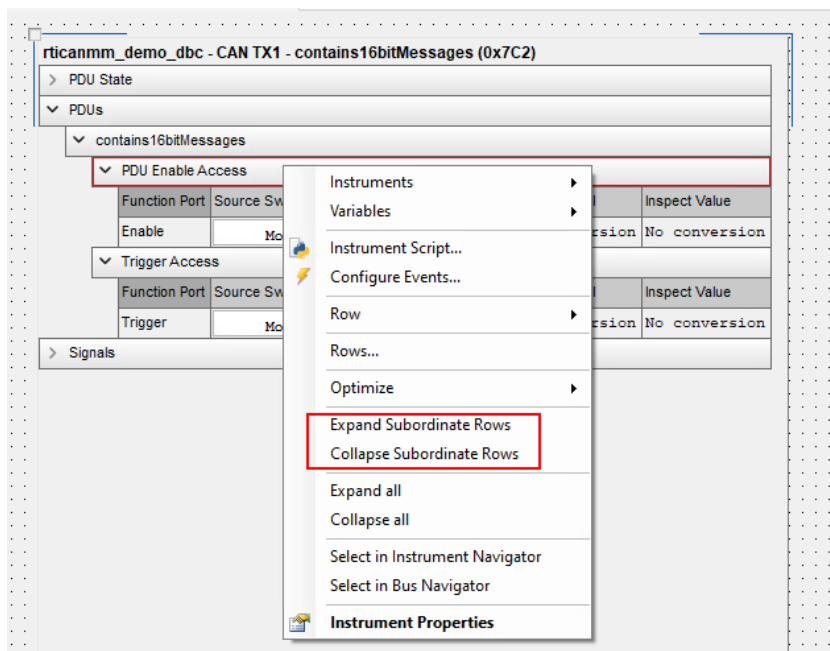
次の図では、2つのタブを使って階層配列を示します。現在アクティブなタブのヘッダーテキストフォントがカスタマイズされています。



「Tabs Properties (Hierarchy Array)」(『ControlDesk 計器の操作』)を参照してください。

**従属行の展開／折りたたみ** 特定のヘッダー行のすべての従属行を展開／折りたたむことができるようになりました。

次の図は、計器のコンテキストメニューの関連コマンドを示しています。アニメーショングラフィックについては、dSPACE Help を参照してください。



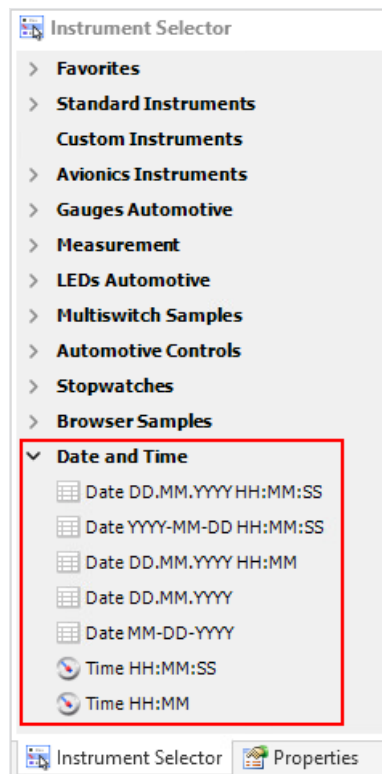
**タブヘッダーをドラッグして計器を移動** タブヘッダーをドラッグして階層配列を移動できるようになりました。

アニメーショングラフィックについては、dSPACE Help を参照してください。

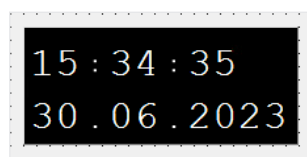
## 日付と時刻計器カテゴリ

ControlDesk Instrument Selector では、ホスト PC のシステム時刻や日付をさまざまなフォーマットでレイアウトに表示できるよう、あらかじめ設定された計器を備えた Date and Time カテゴリを利用できるようになりました。

次の図は計器のカテゴリを示しています。



次の図に、Date DD.MM.YYYY HH:MM:SS 計器の例を示します。



「Date and Time Library」(『ControlDesk 計器の操作』[📖](#))を参照してください。

## 新しい Bus Navigator 機能 (ControlDesk 2023-B)

### SOME/IP-SD メッセージのデコードに AUTOSAR ファイルのポート情報を使用

ControlDesk 2023-A までは、Manage SOME/IP-SD Ports ダイアログで関連ポートを指定することが、SOME/IP-SD メッセージをデコードするために必須でした。

ControlDesk 2023-B では、ControlDesk が関連する Ethernet バスモニタリングデバイスの AUTOSAR 変数記述ファイルから必要な情報を抽出できるため、この操作は必須ではなくなりました。

ただし、AUTOSAR 変数記述ファイルに一致するポート情報がない場合は、Provide Fallback Ports for SOME/IP ダイアログを使用してフォールバックポートを指定することができます。

詳細については、「Manage SOME/IP-SD Ports」(『ControlDesk Bus Navigator』)を参照してください。

## SOME/IP イベントパラメータのインポートとビジュアル表示(プレビュー機能)

ControlDesk 2023-B では、ControlDesk は関連する Ethernet バスモニタリングデバイスの AUTOSAR 変数記述ファイルから SOME/IP イベントパラメータをインポートすることができます。その後、レイアウト上でパラメータをビジュアル表示し、Decoding View でデコードされたイベントパラメータを見ることができます。

### 注記

ControlDesk 2023-B での SOME/IP イベントパラメータのインポートと使用は、**プレビュー機能**の一つです。プレビュー機能とは、初期の、しかし完全にテストされた開発段階で使用される機能です。最終的に通常の製品に統合される前に、ユーザからのフィードバックが製品の改善に役立ちます。ご意見、ご要望、不具合のご報告は、[support@dspace.de](mailto:support@dspace.de) までメールでご連絡ください。

SOME/IP イベントパラメータをインポートしてビジュアル表示するには、以下の手順を実行します。

1. [Preview Features] ページで、SOME/IP イベントパラメータのインポートを有効にします。
2. エクスペリメントに Ethernet バスモニタリングデバイスを追加し、デバイスに AUTOSAR 変数記述ファイルを追加します。

エクスペリメントに既に Ethernet バスモニタリングデバイスが含まれている場合、AUTOSAR 変数記述ファイルをリロードしてください。

[Variables] ペインには、インポートされた SOME/IP イベントパラメータが表示されます。次の図に例を示します。

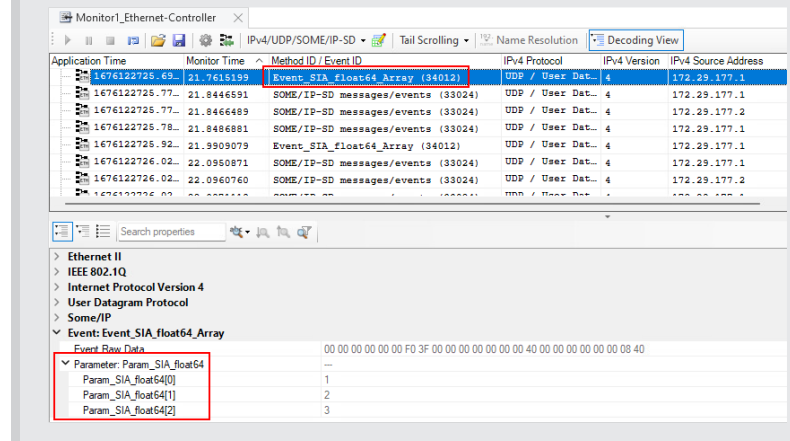
The screenshot shows the 'Variables' panel with a tree view on the left and a table on the right. The tree view shows a hierarchy of variables under 'ecu\_service1' and 'ServiceInterfaceSIA'. The table lists the imported variables with their details.

Favorite	Var Co	Variable	Block	Platform/Device Description	Unit	Type
<input type="checkbox"/>	<input type="checkbox"/>	Param_SIA_float64	Event_SIA_float64_Array	Ethernet		Double 64 bit
<input type="checkbox"/>	<input type="checkbox"/>	Param_SIA_float64[0]	Event_SIA_float64_Array	Ethernet		Double 64 bit
<input type="checkbox"/>	<input type="checkbox"/>	Param_SIA_float64[1]	Event_SIA_float64_Array	Ethernet		Double 64 bit
<input type="checkbox"/>	<input type="checkbox"/>	Param_SIA_float64[2]	Event_SIA_float64_Array	Ethernet		Double 64 bit

- 必要に応じて、SOME/IP イベントパラメータをレイアウト上にビジュアル表示します。

### ヒント

また、Decoding View は、デコードされた SOME/IP イベントパラメータも表示します。次の図に例を示します。



### 注記

これには次の制限事項があります。

- 基本データタイプを持つ SOME/IP イベントパラメータと固定長の 1 次元配列のみがサポートされています。
- Bus Navigator ツリーに SOME/IP イベントパラメータは表示されません。

### バス計器生成のカスタマイズ(プレビュー機能)

ControlDesk 2023-B では、階層配列に基づくバス計器の生成、つまり Bus Manager でモデル化されたバス通信用に作成されたバス計器をカスタマイズすることができます。

### 注記

ControlDesk 2023-B のバス計器生成のカスタマイズは、**プレビュー機能**の一つです。プレビュー機能とは、初期の、しかし完全にテストされた開発段階で使用される機能です。最終的に通常の製品に統合される前に、ユーザからのフィードバックが製品の改善に役立ちます。ご意見、ご要望、不具合のご報告は、[support@dSPACE.de](mailto:support@dSPACE.de) までメールでご連絡ください。

カスタムバス計器を作成するには、次の手順を実行する必要があります。

- カスタムバス計器のタイプごとに個別のレイアウト記述ファイルを作成します。
- ControlDesk の[Options]ダイアログの[Preview Features]ページでカスタムレイアウト記述ファイルの使用を有効にします。
- ControlDesk の[Options]ダイアログの[Bus Navigator - Layout Generation]ページでカスタムレイアウト記述ファイルを含むフォルダを指定します。

詳細については、「[カスタムバス計器の作成](#)」(『ControlDesk Bus Navigator』)を参照してください。

#### J1939-22: マルチ PG IPDU のためのコンパクトおよび操作バス計器の生成 (Bus Manager アプリケーションのみ)

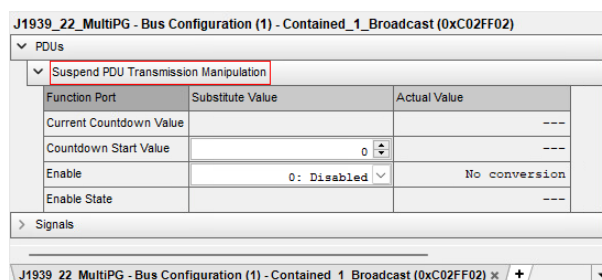
ControlDesk 2023-B では、ペイロード長が最大 60 バイトの J1939-22 準拠の IPDU、つまり Multi-PG プロトコルを使用して送信される IPDU 用のコンパクトバス計器と操作バス計器を生成することができます。

詳細については、以下を参照してください。

- 「[Compact Bus Instrument \(Bus Manager\)](#)」(『ControlDesk Bus Navigator』)
- 「[Manipulation Instrument for CAN \(Bus Manager\)](#)」(『ControlDesk Bus Navigator』)

#### J1939-22: Suspend PDU 送信操作サポート (Bus Manager アプリケーションのみ)

ControlDesk 2023-B では、Bus Manager を使ってモデル化されたバス通信用に生成された CAN 用のコンパクトバス計器および操作バス計器は、新しい Suspend PDU Transmission Manipulation 領域を表示することによって *Suspend PDU Transmission Manipulation* 機能をサポートします。この領域では、J1939-22 IPDU の送信を一時停止することができます。



詳細については、以下を参照してください。

- 「[Compact Bus Instrument \(Bus Manager\)](#)」(『ControlDesk Bus Navigator』)
- 「[Manipulation Instrument for CAN \(Bus Manager\)](#)」(『ControlDesk Bus Navigator』)

#### PDU ユーザポート操作サポート (Bus Manager アプリケーションのみ)

ControlDesk 2023-B では、Bus Manager を使ってモデル化されたバス通信用に生成された CAN 用の操作計器は、PDU User Code Manipulation 領域にユーザポートを表示することによって *PDU user port manipulation* 機能をサポートします。この領域では、ユーザポートの値を設定することができます。



CanCluster - BusConfig1 - IPduA1 (0x051)		
PDU User Code Manipulation		
Function Port	Substitute Value	Actual Value
Current Countdown Value		---
Countdown Start Value	0	---
Enable	0: Disabled	No conversion
Enable State		---
User Inport 0	0	---
User Inport 1	0	---
User Output 2		---
User Inport 3	0	---
User Inport 4	0	---
User Output 5		---
User Output 6		---
User Inport 7	0	---
User Inport 8	0	---
User Output 9		---
User Inport 10	0	---
User Inport 11	0	---
User Inport 12	0	---
User Inport 13	0	---
User Inport 14	0	---
User Inport 15	0	---
User Inport 16	0	---
User Inport 17	0	---
User Inport 18	0	---
User Inport 19	0	---

詳細については、「Manipulation Instrument for CAN (Bus Manager)」  
 (『ControlDesk Bus Navigator』)を参照してください。

### SecOC Freshness Overwrite Value 操作サポート (Bus Manager アプリケーションのみ)

ControlDesk 2023-B では、Bus Manager を使ってモデル化されたバス通信用に生成された CAN 用の操作バス計器は、新しい SecOC Freshness Overwrite Value Manipulation 領域を表示することによって *SecOC freshness overwrite value manipulation* 機能をサポートします。この領域では、選択したセキュア IPDU のフレッシュネス値を上書きすることができます。

CanCluster - BusConfiguration_TX - IPdu02_FVCounter_secured (0x011)		
PDU SecOC Freshness Overwrite Value Manipulation		
Function Port	Substitute Value	Actual Value
Current Countdown Value		---
Countdown Start Value	5	---
Enable	0: Disabled	No conversion
Enable State		---
Freshness Overwrite Value	0	---

詳細については、「Manipulation Instrument for CAN (Bus Manager)」  
 (『ControlDesk Bus Navigator』)を参照してください。

### LIN スケジュールテーブルアクセスサポート (Bus Manager アプリケーションのみ)

ControlDesk 2023-B では、Bus Manager でモデル化されたバス通信用に生成された LIN 用の TX 状態計器は、実行中にアクティブな LIN スケジュールテーブルを変

更できる新しい LIN Schedule Table Access 領域を表示することで、LIN schedule table 機能をサポートしています。

次の図は新しい計器領域を示しています。

LinDoorCluster - Restbus_BusConfiguration						
Communication Controllers						
Communication Controller Enable Access						
LIN Schedule Table Access						
ECU Name	Communication Controller Name	Function Port	Source Switch	Substitute Value	Model Signal	Actual Value
BodyControlEcu	LinDoorCommController	Schedule Index	Model signal	0	---	---
PDUs						

詳細については、「TX Status Instrument for LIN (Bus Manager)」(『ControlDesk Bus Navigator』)を参照してください。

### GTS 通信操作サポート(dSPACE FlexRay Configuration Package ベースのアプリケーションのみ)

ControlDesk 2023-B では、dSPACE FlexRay Configuration Package を使用してモデル化されたバス通信に生成された TX および RX 計器が、新しい Global Time Synchronization および Global Time Synchronization - Sent Values 計器領域を介してグローバル時間同期(GTS)の表示と操作をサポートしています。

- FlexRay 用の TX 計器:
  - Global Time Synchronization 領域では、FlexRay バスに送信する時間同期変数の値を操作できます。
  - Global Time Synchronization - Sent Values 領域には、FlexRay バスに送信される時間同期変数の値が表示されます。

FlexRay 用の RX 計器:

- Global Time Synchronization 領域には、FlexRay バスから受信した時間同期変数の値が表示されます。

以下の図は、FlexRay 用 TX 計器の新しい Global Time Synchronization と Global Time Synchronization - Sent Values 計器領域の例を示しています。

Bus Navigator Array: FlexRay - GTS_TestPdu_00						
PDU : GTS_TestPdu_00 AST : 0x046,0,4 Ch : FlexrayCluster_Domain1_Sub_1_PhysicalChannel						
Enable Bits						
Triggering						
Transmission Mode						
Global Time Synchronization						
Feature Name	Source Switch	Value	Dynamic Value	Countdown Value	Inspect Value	
CRC	SL	0	0	0	---	---
E2E Sequence Counter	SL	0	0	0	---	---
Time Gateway Synchronization Status	SL	0	0	0	---	---
Total Time	SL	0	0	0	---	---
User Byte [0]	SL	0	0	0	---	---
User Byte [1]	SL	0	0	0	---	---
Global Time Synchronization - Sent Values						
Name	Inspect Value					
Time Domain ID	---					
FlexRay Cycle Counter	---					
Synchronization Time (nanoseconds)	---					
Synchronization Time (seconds)	---					

詳細については、以下を参照してください。

- 「TX Instrument for FlexRay (Blockset)」(『ControlDesk Bus Navigator』)
- 「RX Instrument for FlexRay (Blockset)」(『ControlDesk Bus Navigator』)

### 領域ヘッダー内の PDU 状態の表示 (Bus Manager アプリケーションのみ)

ControlDesk 2023-B では、Bus Manager でモデル化されたバス通信に生成された TX 計器とコンパクトバス計器の PDU State 領域ヘッダーに PDU 状態が表示されるため、領域を拡張する必要がなくなりました。

以下の図は、送信が有効になっている PDU の PDU State 領域ヘッダーの例を示しています。



### 「ISignal Value」が有効なシグナルのビジュアル表示 (Bus Manager アプリケーションのみ)

レイアウト上の Bus Navigator ツリーからドラッグして *ISignal Value* (「*ISignal の値の使用*」(『*Bus Manager (スタンドアロン) 実装ガイド*』[📖](#))) を参照機能が有効になっている TX および RX 信号 (TX および RX PDU の信号) をビジュアル表示できるようになりました。

アニメーショングラフィックについては、dSPACE Help を参照してください。

### R22-11 ARXML ファイルのサポート

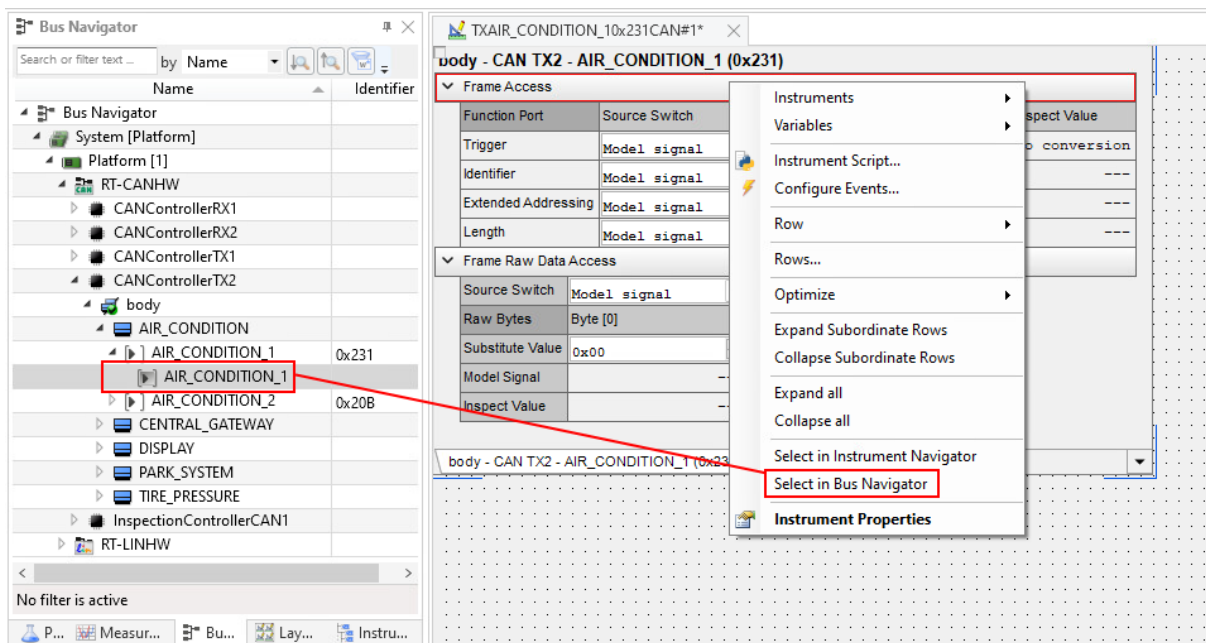
ControlDesk Bus Navigator は、CAN、LIN、および Ethernet バスモニタリングデバイスに関する AUTOSAR R22-11 Release に準拠した AUTOSAR システムデスク립ション (ARXML) ファイルもサポートするようになりました。

「ControlDesk がサポートする変数記述ファイル」(『ControlDesk 変数管理』[📖](#)) を参照してください。

### ユーザビリティの改善

**バス計器に関連する Bus Navigator エLEMENT の選択** Bus Manager でモデル化されたバス通信の場合、関連するバス計器は階層配列に基づいています。このような階層配列から、Bus Navigator ペインのツリーの関連ELEMENT にジャンプできるようになりました。

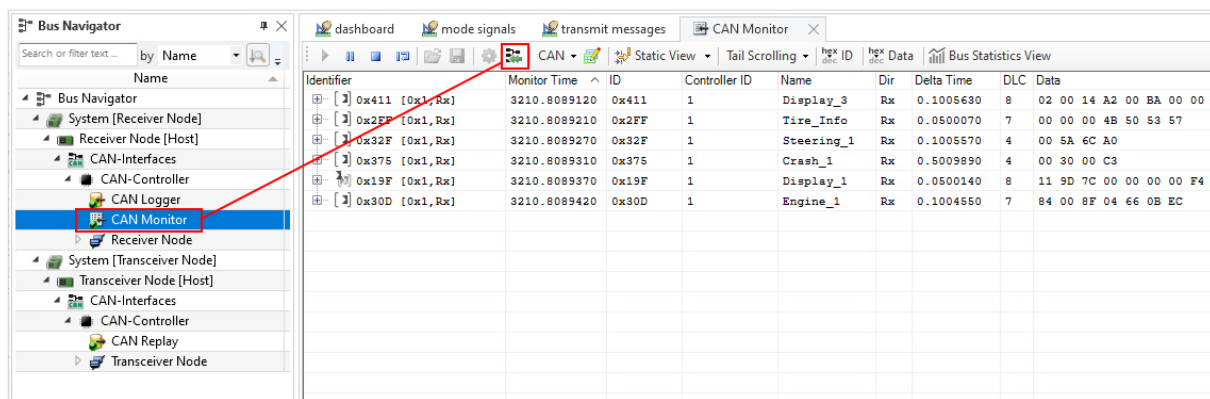
次の図を参照してください。アニメーショングラフィックについては、dSPACE Help を参照してください。



「Select in Bus Navigator」(『ControlDesk 計器の操作』)を参照してください。

**モニタリングリストに関連するモニターノードの選択** Bus Navigator ペインのツリーで、モニタリストから関連するモニターノードにジャンプできるようになりました。

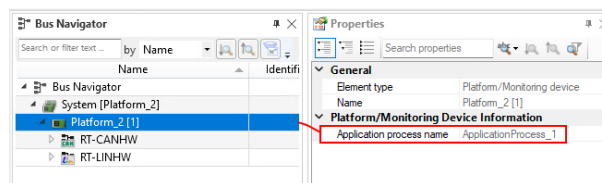
次の図を参照してください。アニメーショングラフィックについては、dSPACE Help を参照してください。



「Monitoring List」(『ControlDesk Bus Navigator』)を参照してください。

**プラットフォームのアプリケーションプロセス名の表示** Bus Navigator ツリーでプラットフォーム要素を選択すると、ControlDesk の Properties ペインに関連するアプリケーションプロセスの名前が表示されるようになりました。

次の図の例を参照してください。



**コンパクトバス計器: 命名マクロのサポート** ControlDesk 2023-B は、コンパクトバス計器レイアウトを生成する際にも、命名マクロをサポートするようになりました。命名マクロは、[Bus Navigator - Layout Generation] ページで指定することができます。

詳細については、「[レイアウト名を設定する方法](#)」(『ControlDesk Bus Navigator』)を参照してください。

#### 関連トピック

#### 参考文献

[Bus Navigator Page](#) (『ControlDesk Bus Navigator』)  
[\[Preview Features\] ページ](#) (『ControlDesk カスタマイズ』)

## 新しい電氣的欠陥シミュレーション機能 (ControlDesk 2023-B)

### 障害シミュレーション用の新しい SCALEXIO FSX ハードウェアのサポート

ControlDesk 2023-B は、障害挿入、シグナルコンディショニング、その他の拡張に関して新しい SCALEXIO FSX ハードウェアをサポートしています。FSX ハードウェアでは、ECU 配線の故障をシミュレートするための標準 I/O ボードを使用できます。

ControlDesk XIL API EESPort ユーザーインターフェースを使用すると、これらの故障を設定および切り替えることができます。

新しい FSX ハードウェアの詳細については、「[FSX ハードウェアによる障害シミュレーション](#)」(『SCALEXIO -ハードウェアおよびソフトウェアの概要』)を参照してください。

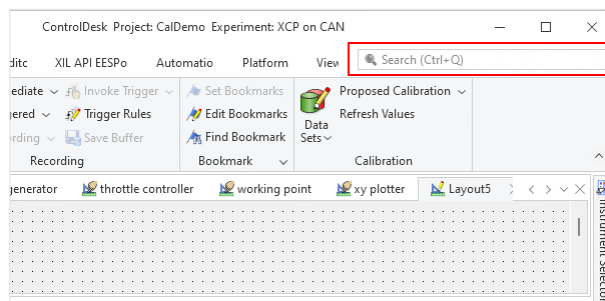
ControlDesk XIL API EESPort のユーザーインターフェースの詳細については、「[電氣的欠陥シミュレーションの基礎](#)」(『ControlDesk XIL API EESPort による電氣的欠陥シミュレーション』)を参照してください。

## 新しいユーザーインターフェース処理機能 (ControlDesk 2023-B)

### 変数とブロックグループの一元的な検索

ControlDesk 2023-B のリボンには、Variables ペインの変数とブロックグループを検索するための新しいフィールドがあります。

下図を参照してください。



Variables ペインで現在使用可能な変数の名前とパスが検索されます。

検索結果リストでは、以下のことを行うことができます。

- Variables ペインの変数リストで変数をクリックして選択します。
- Variables ペインのツリービューでブロックグループをクリックして選択します。
- 変数をレイアウトにドラッグして、計器でビジュアル表示します。

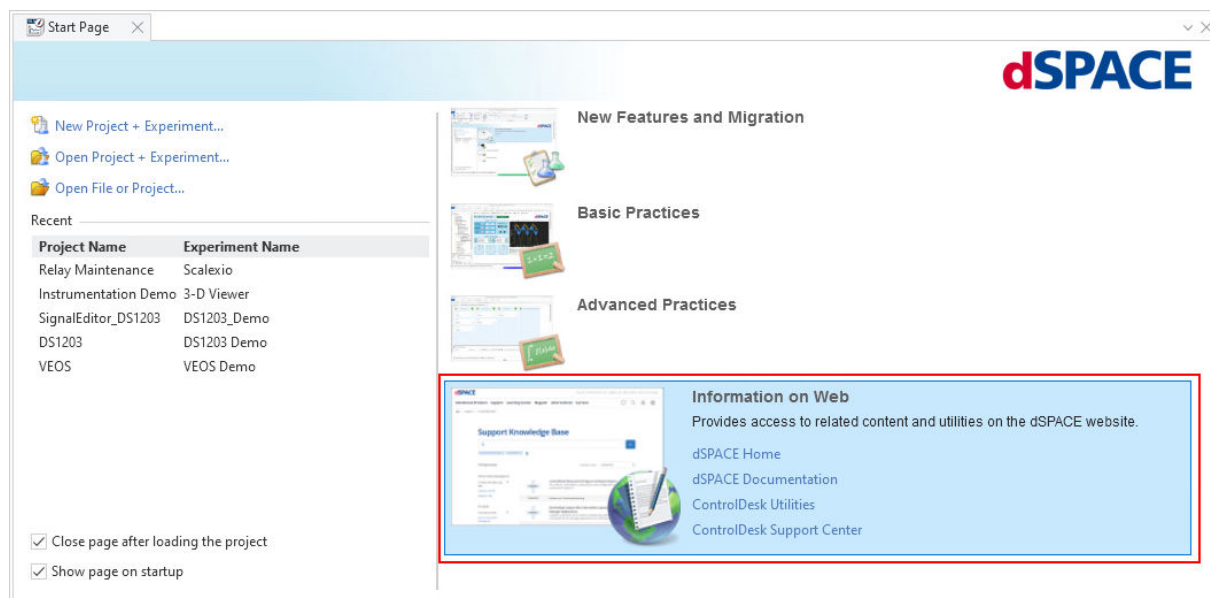
アニメーショングラフィックについては、dSPACE Help を参照してください。


「Search」(『ControlDesk ユーザーインターフェースの操作』)を参照してください。

dSPACE Web サイトのコンテンツ  
およびユーティリティへのアクセス

ControlDesk の Start ページから dSPACE Web サイトのコンテンツおよびユーティリティにアクセスできるようになりました。

次の図を参照してください。



「Start Page」(『ControlDesk ユーザーインターフェースの操作』)を参照してください。

## プレビュー機能

ControlDesk 2023-B では、ControlDesk でプレビュー機能が利用できるようになりました。プレビュー機能とは、初期の、しかし完全にテストされた開発段階で使用される機能です。最終的に通常の製品に統合される前に、ユーザからのフィードバックが製品の改善に役立ちます。

新しい Preview Features ページでプレビュー機能を個別に有効にすることができます。

ご意見、ご要望、不具合のご報告は、[support@dspace.de](mailto:support@dspace.de) までメールでご連絡ください。

### 注記

その後の機能開発によって、プレビュー機能とその機能の最初の安定版との間に互換性のない変更が行われる可能性があります。

## 関連トピック

### 参考文献

[\[Preview Features\]ページ](#) (『ControlDesk カスタマイズ』📖)

# ControlDesk 2023-B への移行

## ControlDesk 2023-B への移行

### 概要

ControlDesk 2023-A から ControlDesk 2023-B に移行して既存のエクスペリメントを再利用するには、次の移行手順が必要になる場合があります。

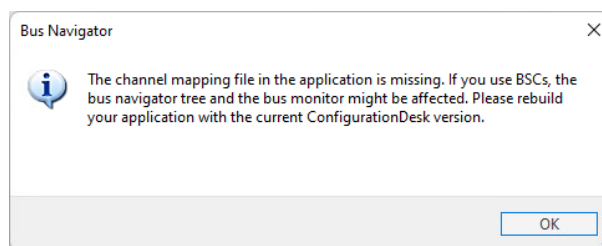
#### 注記

2023-A より前のバージョンから ControlDesk 2023-B に移行するには、その間の ControlDesk バージョンの移行手順も実行する必要があります。

### バス設定によるアプリケーションの再ビルド

dSPACE Release 2023-B では、ConfigurationDesk および VEOS は、Bus Manager ベースのアプリケーションのビルドプロセス中に EXPSWCFG.MAPPING ファイルを作成します。ControlDesk 2023-B 以降では、一貫した Bus Navigator ツリー構造を作成し、このようなアプリケーションのバスモニタリングをサポートするために、このファイルが必要となります。

EXPSWCFG.MAPPING ファイルがない場合に、たとえば、dSPACE Release 2023-A 以前のバージョンで生成された SDF ファイルに含まれるバス設定をロードする場合、そしてバス構成がバスシミュレーションコンテナ (BSC) ファイルに基づいている場合、ControlDesk は Bus Navigator ツリーを正しく作成することができず、次のメッセージが表示されます。



この場合、dSPACE Release 2023-B 以降の ConfigurationDesk または VEOS を使用してアプリケーションを再構築する必要があります。

### A2L ファイルのインポートの変更

ControlDesk 2023-B では、MATRIX\_DIM キーワードを持つ要素を含むバージョン 1.6.1 以前の A2L ファイルのインポートが変更されました。これは、計測配列、値ブロックおよび構造体配列に影響を及ぼします。

- ControlDesk 2023-A までは、このような A2L ファイルをインポートすると、ControlDesk は、MATRIX\_DIM キーワードの次元値が 1 であっても、これらの要素に次元を割り当てます。
- ControlDesk 2023-B では、このような A2L ファイルをインポートすると、ControlDesk は、MATRIX\_DIM キーワードの次元値 1 をすべて無視します。この動作は、ASAM MCD-2 MC 規格の仕様に準拠しています。



その結果、ControlDesk 2023-B では、ControlDesk 2023-A 以前に比べて、計測配列と値ブロックの次元が少なくなります。このことは、自動化スクリプトや ControlDesk レイアウトなどの変数アクセスに影響を及ぼします。

例 CalDemo プロジェクトの CalDemo.a21 ファイルには、MeasureArray 変数が含まれています。この変数には以下の MATRIX\_DIM キーワードが含まれていません。

MATRIX\_DIM 10 1 1

- 次の図は、ControlDesk 2023-A 以前のバージョンで A2L ファイルをインポートした後の [Variables] ペインの MeasureArray 変数を示しています。

MeasureArray
MeasureArray[0][0][0]
MeasureArray[1][0][0]
MeasureArray[2][0][0]
MeasureArray[3][0][0]
MeasureArray[4][0][0]
MeasureArray[5][0][0]
MeasureArray[6][0][0]
MeasureArray[7][0][0]
MeasureArray[8][0][0]
MeasureArray[9][0][0]

次の図は、MeasureArray 変数のデータタイププロパティを示しています。

Data Type	
Data type	Array_UWORD_DIRECT_...
Byte order	MSB Last (Intel)
Range of values	[0, 65535]
Size in bytes	20
Dimensions	[10][1][1]

以下のリストは、ControlDesk 2023-A 以前のバージョンで、MeasureArray 変数の特定の元素にアクセスする方法を示しています。

```
...ActiveVariableDescription.Variables.ItemByPath('XCP()://MeasureArray[5][0][0]')
```

- 次の図は、ControlDesk 2023-B 以降のバージョンで A2L ファイルをインポートした後の [Variables] ペインの MeasureArray 変数を示しています。

MeasureArray
MeasureArray[0]
MeasureArray[1]
MeasureArray[2]
MeasureArray[3]
MeasureArray[4]
MeasureArray[5]
MeasureArray[6]
MeasureArray[7]
MeasureArray[8]
MeasureArray[9]

次の図は、MeasureArray 変数のデータタイププロパティを示しています。

Data Type	
Data type	Array_UWORD_DIRECT_Row...
Byte order	MSB Last (Intel)
Range of values	[0, 65535]
Size in bytes	20
Dimensions	[10]

以下のリストは、ControlDesk 2023-B 以降のバージョンで、MeasureArray 変数の特定の元素にアクセスする方法を示しています。

```
...ActiveVariableDescription.Variables.ItemByPath('XCP()://MeasureArray[5]')
```

## ツール自動化の変更

**Python 3.11 への移行** dSPACE Release 2023-B 以降、dSPACE 製品は Python 3.9 をサポートしなくなりました。その代わりに、Python 3.11 がサポートされます。

ControlDesk 固有の移行上の注意点はありません。

詳細については、「[Python 3.9 から Python 3.11 への Python スクリプトの移行](#)」(31 ページ)を参照してください。

**ライセンスの取り扱いとツールの自動化の変更** ControlDesk 2023-B では、ライセンスの取り扱いが変更されました。

その結果、ControlDesk のメインバージョンの有効なライセンスがない場合、Application / IXaApplication <<Interface>>インターフェースの以下のプロパティとメソッドのみを使用することができます。

- CommandLineArguments
- Environment
- Execute(...)
- Log
- MainWindow
- Name
- Quit(...)
- UserSettings
- Version
- VersionInformation

**Table Editor 自動化インターフェースの変更点** ControlDesk 2023-B では:

- Header プロパティが「TableEditorArrayInstrument / IViTableEditorArrayInstrument <<Interface>>」([『ControlDesk 自動化』](#))から削除されました。
- 「Views / IViViews <<Interface>>」([『ControlDesk 自動化』](#)) インターフェースから InfoHeader プロパティが削除されました。このプロパティは、「TableEditorInstrument / IViTableEditorInstrument <<Interface>>」([『ControlDesk 自動化』](#))によって返されます

## 以前の ControlDesk バージョンからの移行

以前の ControlDesk バージョンから移行して既存のエクスペリメントを再利用するには、追加の移行手順が必要な場合があります。移行手順の詳細については、「[以前のバージョンの ControlDesk からの移行](#)」([『ControlDesk 新機能と移行手順』](#))を参照してください。

## 関連トピック

### 基本概要

[CalDemo](#) ([『ControlDesk 概要』](#))  
[以前のバージョンの ControlDesk からの移行の基礎](#) ([『ControlDesk 新機能と移行手順』](#))

### 参考文献

[Application / IXaApplication <<Interface>>](#) ([『ControlDesk 自動化』](#))  
[Variables / IXaVariables <<Collection>>](#) ([『ControlDesk 自動化』](#))

# DCI Configuration Tool

## DCI Configuration Tool 2023-B の新機能

### DCI-GSI2 インターフェースの ファームウェアバージョン

DCI-GSI2 インターフェースでは、ファームウェアバージョン 1.5.8 が DCI Configuration Tool 2023-B で提供されます。

#### 注記

DCI Configuration Tool で提供されるファームウェアバージョンは、必ずしも使用可能な最新のファームウェアバージョンではありません。問題が発生する場合は、新しいファームウェアバージョンが使用可能かどうかを dSPACE サポートにお問い合わせください。




# dSPACE ECU Flash Programming Tool

## dSPACE ECU Flash Programming Tool 2023-B の新機能

---

### CAN\_MAX\_DLC 値を使用

XCP on CAN ECU インターフェースが選択されたフラッシュプロジェクトを設定する際に、dSPACE ECU Flash Programming Tool で、CAN/CAN FD メッセージに指定された最大データ長コード(DLC)値を使用するか、実際に使用される長さを使用するかを指定できるようになりました。

「[Configure Interface Settings]ダイアログ」(『ECU フラッシュプログラミング』)を参照してください。



# dSPACE FlexRay Configuration Package

## dSPACE FlexRay Configuration Package 2023-B の新機能

### FlexRay Configuration Package

**AUTOSAR Classic Platform R22-11 のサポート** FlexRay Configuration Package では、FlexRay ネットワークの記述に、AUTOSAR Classic Platform Release R22-11 に基づいた AUTOSAR システムテンプレートの形式がサポートされています。ただし、AUTOSAR Classic Platform Release R22-11 の新機能はサポートされていません。

「設定に使用できるコミュニケーションクラスタファイル」(『FlexRay Configuration Tool ガイド』[📖](#))を参照してください。

### FlexRay Configuration Tool

**[Create Corresponding ECU]コマンドの機能強化** FlexRay Configuration Tool 2023-A までは、Create Corresponding ECU コマンドの実行は、Configuration ビューにまだ追加されていない ECU に対してのみ可能でした。

FlexRay Configuration Tool 2023-B では、既にシミュレーション設定に含まれている ECU に対してレストバス設定を作成することもできます。[General Properties] ダイアログで許可を与えた場合、FlexRay Configuration Tool では Create Corresponding ECU コマンドを、既に Configuration ビューに表示されている ECU に対しても呼び出すことができます。

「レストバスシミュレーションの設定方法」(『FlexRay Configuration Tool ガイド』[📖](#))を参照してください。





# dSPACE Installation Manager

## 次のステップ

## 本章の内容

dSPACE Installation Manager 23.2 の新機能.....	113
dSPACE Installation Manager 23.2 への移行.....	114

## dSPACE Installation Manager 23.2 の新機能

### 概要

dSPACE Installation Manager 23.2 では、ユーザに関連する新機能はありません。ただし、バージョン 23.1 以降、以下のような廃止項目があることに注意してください。

### dSPACE Installation Manager による dongle 移行の廃止

dSPACE Release 2023-A 以降、dSPACE Installation Manager では、古い dongle (dSPACE Release 2017-A 以前に納品) から現在納品されている CmDongle への移行をサポートしなくなります。

dSPACE Release 2017-B 以降を使用するために移行が必要な古い dongle をまだお持ちの場合は、次のようになります。

- WibuKey dongle (タイプ 1) を使用しているユーザの場合、今回のリリースから移行ができなくなりました。dSPACE による dongle の交換が必要です。詳細については、<http://www.dspace.jp/go/DongleReplacement> を参照してください。
- 第一世代の CmDongle (タイプ 2、2-xxxx で始まるシリアル番号の白色のコネクタ) を使用しているユーザの場合、今回のリリースから移行ができなくなりました。dSPACE による dongle の交換が必要です。詳細については、<http://www.dspace.jp/go/DongleReplacement> を参照してください。
- シリアル番号が 3-xxxx で始まり、古いファームコードを使用した CmDongle (タイプ 3) を使用しているユーザの場合、dSPACE Installation Manager の旧バージョン (5.0 から 22.3 まで) を使用して移行する必要があります。dSPACE Installation Manager 22.3 のダウンロードについては、<http://www.dspace.jp/go/IM> を参照してください。

## レガシーライセンステクノロジーの廃止

dSPACE Release 2023-A 以降、dSPACE Installation Manager はレガシーライセンステクノロジー( dongle ライセンス用の Wibu-Systems 社の WibuKey、フローティングネットワークライセンス用の Flexera 社の FlexNet)をサポートしなくなりました。これには、以下のような意味があります。

- dSPACE Installation Manager 23.1 (dSPACE Release 2023-A に付属)以降では、dSPACE Release 2017-B で導入された CodeMeter ライセンステクノロジーのみサポートします。
- dSPACE Installation Manager 23.1 以降では、CodeMeter ライセンスに基づくレガシーライセンス用ファイルダウンロード (keys.dsp および license.dsp) を提供しなくなります。
- dSPACE Release 2023-A 以降を dSPACE Release 2017-A 以前と同じ PC にインストールすることは、サポートが終了しているため、推奨できません。
- dSPACE Release 2023-A 以降、dSPACE License Manager (Legacy) は dSPACE Installation Manager と共に出荷もインストールもされなくなりました。関連するコマンドラインユーティリティ(特に IMLicUtil.exe および InstallationReporter.exe ユーティリティなど)も廃止されています。
- それでもレガシーライセンス方式を使用する必要がある場合は、このテクノロジーをサポートする dSPACE Installation Manager の最新バージョンを使用する必要があります。今回のリリースでは、dSPACE Installation Manager 22.3 が最新バージョンです。dSPACE Installation Manager 22.3 は、<http://www.dspace.jp/go/IM> からダウンロードすることができます。

## dSPACE Installation Manager 23.2 への移行

### CmDongles の使用

dSPACE Installation Manager 23.2 と組み合わせて、CmDongle でのライセンスのアクティブ化、非アクティブ化、アップデートなどを実行したい場合は、dongle のファームウェアはバージョン 4.10 以上が必要です。

dSPACE Release 2019-A より前のリリースに付属する CmDongle を使用するには、ファームウェアのアップデートが必要です。dSPACE Release 2019-A 以降に付属の CmDongle には、必要な最小ファームウェアバージョンが含まれています。

dSPACE Installation Manager は、接続されたdongleのファームウェアが必要な最小ファームウェアバージョンと一致するかどうかをチェックし、アップデートが必要かどうかを表示します。

ファームウェアアップデートの手順については、「[CmDongle のファームウェアをアップデートする方法](#)」(『CodeMeter ライセンステクノロジーの使用』)を参照してください。

### Microsoft .NET6 フレームワークへの移行

dSPACE Installation Manager 23.1 以降では、dSPACE Installation Manager は Microsoft .NET6 フレームワークに移行しました。下位互換性を実現するために、dSPACE Installation Manager は .NET4.8 フレームワークも dSPACE Release 2026-B までサポートします。したがって、dSPACE Installation Manager の API は、.NET6 プロセスだけでなく .NET4.8 プロセスでも使用することができます。

dSPACE Installation Manager の API の使用に関する注意事項:.NET4.8(APIバージョン 2.0)は、後のリリースでサポートが終了する予定です。その場合、スクリプトとプログラムを API バージョン 3.0 に移行する必要があります。したがって、新規のスクリプトには、今のうちから API バージョン 3.0 を使用してください。



# dSPACE Python Extensions

## dSPACE Python Extensions 2023-B の新機能

---

### 一般的な機能強化

**Python 3.11 のサポート** dSPACE Python Extensions は、Python 3.11 と付属の pythonnet 3.0 パッケージをサポートしています。詳細については、「[Python 3.9 から Python 3.11 への Python スクリプトの移行](#)」(31 ページ)を参照してください。



# dSPACE XIL API .NET

## 次のステップ

## 本章の内容

dSPACE XIL API .NET 2023-B の新機能.....	119
dSPACE XIL API .NET 2023-B への移行.....	120

## dSPACE XIL API .NET 2023-B の新機能

### 一般的な機能強化

**Python 3.11 のサポート** dSPACE XIL API .NET は、Python 3.11 と付属の pythonnet 3.0 パッケージをサポートしています。必要な移行手順については、「[移行に関する製品固有の情報](#)」(35 ページ)を参照してください。

### MAPort 機能の拡張

**新しいシミュレーションプラットフォームのサポート** dSPACE XIL API .NET の MAPort 実装では、シミュレーションプラットフォームとして新しい MicroLabBox II の使用をサポートしています。新しい MicroLabBox II の詳細については、「[一般的な機能拡張および変更](#)」(13 ページ)を参照してください。

詳細については、『[dSPACE XIL API MAPort インプリメンテーション](#)』📖を参照してください。

### EESPort 機能の拡張

**新しい EES ハードウェアのサポート** dSPACE XIL API .NET の EESPort 実装は、障害シミュレーション用の新しい SCALEXIO FSX ハードウェアの使用をサポートしています。

詳細については、『[dSPACE XIL API EESPort インプリメンテーション](#)』📖を参照してください。

### ECUPort 機能の拡張

**ECUPort から MF4 へのキャプチャ** V-ECU からデータをキャプチャし、ホスト PC 上の MF4 ファイルに保存できるようになりました。RAM または MF4 へ

の保存の最新の実装は、トリガを使用しないキャプチャの直接開始／停止のみをサポートしています。

詳細については、『[dSPACE XIL API ECU Port Implementation](#)』を参照してください。

## dSPACE XIL API .NET on Linux

**Python クライアントの廃止** dSPACE XIL API .NET on Linux には、dSPACE 固有の Python クライアントが含まれなくなりました。Linux での Python アクセスは、pythonnet を介した Windows でのアクセスに合わせて調整されています。必要な移行手順については、『[dSPACE XIL API .NET 2023-B への移行](#)』(120 ページ)を参照してください。

## dSPACE XIL API .NET 2023-B への移行

### dSPACE XIL API .NET on Linux の移行に関する問題

Python クライアントが廃止されたため、Python ベースの XIL API アプリケーションを pythonnet に移行する必要があります。pythonnet パッケージを使用する場合には、戻り値の取り扱いを調整する必要があります。戻り値の扱いは、以前のバージョンでは Python オブジェクトでしたが、現在のバージョンでは .NET オブジェクトになりました。

**pythonnet パッケージの使用** 次の例では、必要な clr モジュールをインポートする方法と、ASAM アセンブリをロードする方法を示します。

```
import pythonnet
pythonnet.load("coreclr")
import clr

# Load ASAM assemblies
clr.AddReference("ASAM.XIL.Implementation.TestbenchFactory,
Version=2.2.0.0, Culture=neutral, PublicKeyToken=f9604847d8afbfb")
clr.AddReference("ASAM.XIL.Interfaces, Version=2.2.0.0,
Culture=neutral, PublicKeyToken=f9604847d8afbfb")
```

Linux での pythonnet の使用とアセンブリのロードは、以下の項目を除いて Windows での使用と同様になりました。

- `pythonnet.load("coreclr")` を呼び出してランタイムを設定する必要があります。
- ASAM は Linux オペレーティングシステム用のアセンブリを提供していません。したがって、PublicKeyToken 引数に `f9604847d8afbfb` を指定して、dSPACE 固有のアセンブリを参照する必要があります。

**戻り値の取り扱い** pythonnet を使用すると、XIL API のメソッドやプロパティの戻り値は、Python のデータタイプ `PyList` や `PyDict` ではなく、`IList` や `IDictionary` などの .NET オブジェクトになります。



戻り値のタイプが変更されたため、たとえば、次のようにスクリプトを移行する必要があります。

- dSPACE Release 2023-A 以前の XIL API .NET での Python データタイプの使用

```
# Example for PyList
variableNames = CaptureResult.GetVariableNames(...)
numVariables = len(variableNames)
variableNames.append(newValue)
variableNames.sort()
# Example for PyDict
watcherDefines = Watcher.Defines
numDefines = len(watcherDefines)
```

- dSPACE Release 2023-B 以降の XIL API .NET での .NET オブジェクトの使用

```
# Example for IList
variableNames = CaptureResult.GetVariableNames(...)
numValues = variableNames.Count
variableNames.Add(newValue)
variableNames.Sort()
# Example for IDictionary
watcherDefines = Watcher.Defines
numDefines = watcherDefines.Count
```

以下に、影響を受ける XIL API メソッドとプロパティの例を示します。

メソッド/プロパティ	戻り値 データタイプ	
	Release 2023-A 以前	Release 2023-B 以降
Capture.Variables	PyList	IList<DataType>
Capture.Variables2	PyList	IList<DataType>
CaptureResult.GetVariableNames	PyList	IList<DataType>
MAPort.TaskInfos	PyList	IList<DataType>
VectorValue.Value	PyList	IList<DataType>
CaptureResult.MetaData	PyDict	IDictionary<DataType>
Watcher.Defines	PyDict	IDictionary<DataType>



# ECU Interface Manager

## 次のステップ

## 本章の内容

<a href="#">ECU Interface Manager 2023-B の新機能</a> .....	123
ECU Interface Manager 2023-B の新機能の概要。	
<a href="#">ECU Interface Manager 2023-B の互換性</a> .....	124
ECU Interface Manager 2023-B の互換性に関する情報を提供します。	
<a href="#">ECU Interface Manager 2023-B への移行</a> .....	124
ECU Interface Manager 2023-B への移行方法に関する情報。	

## ECU Interface Manager 2023-B の新機能

### XCP タイムスタンプ同期

XCP サービスが統合された ECU アプリケーションがタイムスタンプをサポートしている場合、つまり ECU アプリケーションの A2L ファイルに `TIMESTAMP_SUPPORTED` エントリが含まれている場合、XCP タイムスタンプを ECU に接続された dSPACE リアルタイムハードウェアのシステム時刻と同期させることができます。また、XCP 経由で計測した ECU データと、dSPACE ハードウェアに接続されている他のソースからのデータを時間相関させることもできます。

「[XCP タイムスタンプと ECU データの計測](#)」(『[ECU Interface Manager マニュアル](#)』[123](#))を参照してください。

### 新しい MicroLabBox II のサポート

ECU Interface Manager を使用すると、ConfigurationDesk および新しい MicroLabBox II を使用して外部 ECU インターフェース処理を次のように準備することができます。

- MicroLabBox II の ECU ファンクションへのアクセスを設定することができます。
- データアクセスと動的データアクセスを設定することによって、MicroLabBox II の ECU 変数への読み取り/書き込みアクセスを設定することができます。
- ECU Interface Manager では、ConfigurationDesk および MicroLabBox II を使用して ECU 適合ページへのアクセスを設定することができます。

MicroAutoBox II は、次の ECU インターフェースをサポートしています。

- DCI-GSI2
- XCP on CAN
- XCP on Ethernet

## ECU Interface Manager 2023-B の互換性

### 一般的な互換性

dSPACE では、同一の dSPACE Release のソフトウェア製品のみ使用することをお勧めしています。これにより、最大限のランタイム互換性が保証されます。

### EIC ファイルと ConfigurationDesk の互換性

次の表は、EIC ファイルと ConfigurationDesk の互換性を示しています。

	EIC ファイルの作成に使用した ECU Interface Manager のバージョン			
	バージョン 2.11 <sup>1)</sup>	バージョン 2022-B <sup>2)</sup>	バージョン 2023-A <sup>3)</sup>	バージョン 2023-B <sup>4)</sup>
ConfigurationDesk 2023-B <sup>4)</sup>	✓ <sup>5)</sup>	✓ <sup>5)</sup>	✓ <sup>5)</sup>	✓ <sup>5)</sup>
ConfigurationDesk 2023-A <sup>3)</sup>	✓ <sup>5)</sup>	✓ <sup>5)</sup>	✓ <sup>5)</sup>	✓ <sup>5)</sup>
ConfigurationDesk 2022-B <sup>2)</sup>	✓ <sup>5)</sup>	✓ <sup>5)</sup>	✓ <sup>5)</sup>	✓ <sup>5)</sup>
ConfigurationDesk 22.1 <sup>1)</sup>	✓ <sup>5)</sup>	✓ <sup>5)</sup>	✓ <sup>5)</sup>	✓ <sup>5)</sup>

1) dSPACE Release 2022-A

2) dSPACE Release 2022-B

3) dSPACE Release 2023-A

4) dSPACE Release 2023-B

5) 32 ビットおよび 64 ビットターゲットアーキテクチャの場合。

### EIC ファイルと RTMaps の互換性

RTMaps は、ECU Interface Manager 2.10 以降で作成された EIC ファイルをサポートします。

## ECU Interface Manager 2023-B への移行

### プロジェクトの自動移行

ECU Interface Manager 2.0 p1 以降でプロジェクトを最後に保存した場合、ECU Interface Manager 2023-B でプロジェクトを再利用することができます。

ECU Interface Manager 2023-B でプロジェクトを開くと、自動的に移行されます。

**注記**

ECU Interface Manager 2023-B では、ECU Interface Manager 2.0 以前で最後に保存したプロジェクトを再利用することはできません。

**追加の移行手順が必要な場合**

ECU Interface Manager 2023-A より前のバージョンから ECU Interface Manager 2023-B に移行するには、その間の ECU Interface Manager バージョンの移行手順も必要になる場合があります。



# Firmware Manager

## Firmware Manager 23.2 の新機能

### SCALEXIO と MicroAutoBox III のデフォルトのファームウェアアーカイブの変更

dSPACE Release 2023-A までは、SCALEXIO および MicroAutoBox III の [Available archive versions] ダイアログで事前に選択されたファームウェアアーカイブは、アクティブな dSPACE Release のファームウェアアーカイブでした。dSPACE Release 2023-B では、アクティブなリリースが Release 2021-B より新しい場合、そのリリースへの依存関係がなくなります。事前に選択されたファームウェアアーカイブは、インストールフォルダで使用可能な最新のアーカイブバージョンになります。

デフォルトの SCALEXIO ファームウェアアーカイブに、64 ビット Linux ベースのファームウェアコンポーネントが含まれるようになりました。32 ビット Linux ベースのファームウェアコンポーネントを読み込むには、32 ビットファームウェアアーカイブを明示的に選択する必要があります。

### プラットフォームサポートの強化

Firmware Manager では、MicroLabBox II のアップデートをサポートします。





# FPGA Programming Blockset

## 次のステップ

## 本章の内容

FPGA Programming Blockset 2023-B の新機能.....	129
FPGA Programming Blockset 2023-B への移行.....	131

## FPGA Programming Blockset 2023-B の新機能

**拡張 FPGA 設計ツールのサポート**      FPGA Programming Blockset が以下の製品をサポートするようになりました。

設計ツールのバージョン	MATLAB バージョン <sup>1)</sup>	オペレーティングシステム
AMD Vivado 2023.1 <sup>2)</sup>	MATLAB R2022a	<p>最新リリースの RCP および HIL ソフトウェアによってサポートされる Windows オペレーティングシステム。</p> <p>サポートされるオペレーティングシステムのリストについては、「<a href="#">オペレーティングシステム</a>」(263 ページ)を参照してください。</p> <ul style="list-style-type: none"> <li>リストに記載されている Windows Server オペレーティングシステムは、AMD の公式のサポートはありませんが、dSPACE によってテスト済みです。</li> <li>AMD Vitis Model Composer は、Windows 11 を正式にサポートしていませんが、dSPACE ではテスト済みです。</li> </ul>

<sup>1)</sup> RTI FPGA Programming Blockset の Processor Interface サブライブラリは、MATLAB R2022b、R2023a、および R2023b もサポートしています。

<sup>2)</sup> 一般的に、Vivado ML Enterprise エディションと AMD Vitis Model Composer が必要です。AMD Vitis Model Composer 付属の Vivado ML Standard エディションは、DS2655 (7K160) および DS6601 FPGA Base Board でも使用することができます。AMD Vitis Model Composer は、FPGA Programming Blockset を使用して FPGA アプリケーションをモデリングする場合にのみ必要です。

FPGA プログラミング用のソフトウェアツールの詳細については、「[FPGA Programming Blockset を使用するためのソフトウェアツール](#)」(『[FPGA Programming Blockset ガイド](#)』[📖](#))を参照してください。

**MATLAB の互換性** MATLAB の互換性に制限事項があるのは、AMD Design Tools の要件と FPGA アプリケーション作成におけるビルドプロセスへの影響を考慮しているためです。

FPGA のビルドプロセスとは対照的に、プロセッサアプリケーションのモデリングとビルドのためのブロックセットとツールは、dSPACE Release の標準的な MATLAB 互換性をサポートしています。そのため、FPGA アプリケーションとは別個にプロセッサアプリケーションをモデリングしてビルドすると、MATLAB の互換性を拡張することができます。唯一の要件は、FPGA アプリケーションとプロセッサアプリケーションのビルドに同じ dSPACE Release を使用することです。

例:FPGA アプリケーションは、現在の dSPACE Release、Vivado 2023.1、および MATLAB R2022a を含むインストールセットで作成されています。生成された FPGA アプリケーションを、現在の dSPACE Release と MATLAB R2023b を含むインストールセットに転送およびインポートして、ConfigurationDesk または FPGA Programming Blockset の Processor Interface サブライブラリを使用して作業を継続することができます。

## FPGA Programming Blockset の一般的な機能強化

**インクリメンタルビルド** FPGA Programming Blockset では、Vivado のインクリメンタルビルド機能を有効にすることで、一貫して高い FPGA 使用率でビルドを成功させることができます。既知の良好なリファレンスビルドが実装の出発点として使用されるため、新しいビルドでも FPGA コンフィギュレーションの動作を前提にして、実際の変更点以外は実装する必要はありません。インクリメンタルビルドは、FPGA ビルドが成功する可能性を高め、FPGA モデルのわずかな変更にかかるビルド時間を短縮することができます。

「[FPGA アプリケーションをインクリメンタルビルドする方法](#)」(『[FPGA Programming Blockset ガイド](#)』[📖](#))を参照してください。

**dSPACE Timing Analyzer** 新しい dSPACE Timing Analyzer を使用すると、ビルドプロセス中に検出されたタイミングエラーを修正できます。

タイミングアナライザは、次のような機能であなたをサポートします。

- 使用するブロックライブラリ( AMD Vitis Model Composer の HDL ライブラリや標準 Simulink ライブラリ(HDL Coder サブシステム))に依存せず、ビルドされた FPGA アプリケーションの最もクリティカルな FPGA 信号パスを表示します。  
FPGA 信号パスの数は、1,000 ビットの信号に制限されています。たとえば、20 ビット幅のデータパスは 50 本を表示することができます。
- タイミングが収束していない信号パス内で使用されているブロックの詳細情報を提供します。
- FPGA モデルでタイミングが収束していない信号パスの選択されたブロックを開きます。これにより、エラーを修正するための迅速なアクセスが可能になります。
- 選択した信号パスの詳細情報をタイミングレポートに表示します。これらの詳細情報は、詳細な分析が必要なエラーの修正に役立ちます。

詳細については、「[dSPACE Timing Analyzer の機能](#)」(『[FPGA Programming Blockset ガイド](#)』[📖](#))を参照してください。

**メッセージ処理** FPGA Programming Block は、可能性のある問題や今後の処置について通知する新しいメッセージ処理機能を提供します。これらのメッセージは、MATLAB コマンドウィンドウやダイアログに表示されます。メッセージには、詳細説

明が記載された新しい『FPGA Programming Blockset Message Reference』の関連トピックを開くリンクやボタンが含まれています。

### 新しい MicroLabBox II フレームワーク

MicroLabBox II は、DS1303 Multi-I/O Board に FPGA を搭載しており、FPGA Programming Blockset を使用してプログラムすることができます。FPGA Programming Blockset で構築された FPGA カスタムファンクションブロックタイプは、I/O ボードの標準 I/O 機能に取って代わります。FPGA カスタムファンクションブロックタイプを使用するか、ConfigurationDesk が提供するファンクションブロックタイプを使用して I/O 機能を実装します。

提供される I/O 機能の概要については、「Block Settings for the MicroLabBox II Framework」(『FPGA Programming Blockset - FPGA Interface Reference』)を参照してください。

FPGA アプリケーションをハンドコーディングする場合は、提供されている Tcl スクリプトを使用して FPGA アプリケーションをビルドします。「FPGA アプリケーションをビルドする方法」(『FPGA Programming Handcode Framework Guide』)を参照してください。

#### 注記

このフレームワークは、サービスパックのインストール後に利用可能になります。このサービスパックは、MicroLabBox II と一緒にリリースされる予定です。

### SCALEXIO および MicroAutoBox III フレームワークの機能拡張

これらのフレームワークは現在、ユーザクロックドメインを使用するサブシステム向けにクロック信号と追加情報を提供しています。

- FPGA のベースレートを持つクロック信号。
- ユーザクロックドメインを使用する他のサブシステムのクロック信号。
- FPGA アプリケーション、プロセッサアプリケーション、およびオプションの MGT モジュールに関するシステム情報。

詳細については、「同じサブシステムにおける異なるクロックドメインのクロック信号の使用」(『FPGA Programming Blockset ガイド』)を参照してください。

## FPGA Programming Blockset 2023-B への移行

### 以前の RTI FPGA Programming Blockset からの移行

FPGA Programming Blockset バージョン 1.1 以降で FPGA アプリケーションを実装しており、FPGA Programming Blockset 2023-B でそれを使用する場合、フレームワークは最新のフレームワークバージョンに自動的に更新されます。

この更新では、モデル/サブシステム内のすべてのサブシステムが影響を受けません。最新のフレームワークバージョンにアップデートしても、ブロックのパラメータは変更されません。

たとえば、FPGA モデルなしで RTI プロセッサモデルを移行するために、スクリプトを使用してプロセッサインターフェースブロックを移行することもできます。詳細について

では、「MigrateToModelPortBlocks」(『FPGA Programming Software Script Interface Reference』)を参照してください。

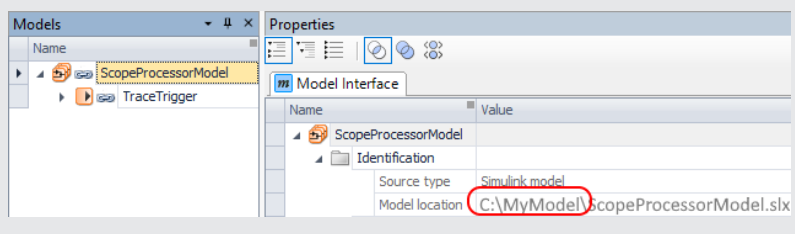
**FPGA スコープラスタの生成プロセスの変更** ConfigurationDesk で RTI FPGA Programming Blockset 3.13 (Release 2022-A) を使用してビルドされた FPGA カスタムファンクションを使用する場合、ControlDesk で使用される FPGA スコープラスタを生成する必要があります。

FPGA Programming Blockset 2022-B では、FPGA スコープファンクションを含む FPGA カスタムファンクションブロックで ControlDesk 用の FPGA スコープラスタが自動生成されます。

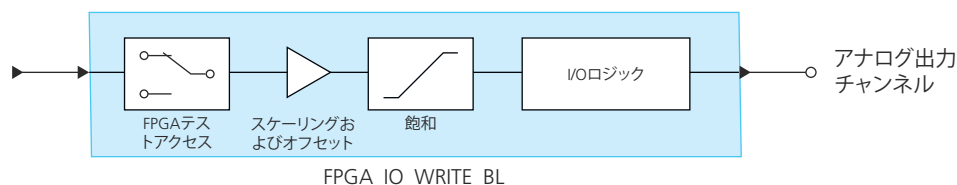
既存の ConfigurationDesk プロジェクトを使用して、Release 2022-A の FPGA カスタムファンクションを新しい FPGA カスタムファンクションに更新すると、既存の FPGA スコープラスタ(<application>\_usr.trc) はエラーになります。

#### 注記

新しい FPGAC ファイルを使用して ConfigurationDesk の既存の FPGA カスタムファンクションを更新する前に、モデルフォルダ内の <application>\_usr.trc ファイルを削除してください。ConfigurationDesk でモデルの位置が表示されます。



**アナログ出力信号を変更するためのインターフェースの変更** FPGA スケーリングにより、アナログ出力信号を変更することができます。Release 2022-B では、FPGA テストアクセスに置き換えられた信号値をインターフェースでスケールします。次の図は、アナログ出力インターフェースを示しています。FPGA テストアクセスおよびスケールリングの詳細については、「FPGA テストアクセスおよびスケールリングの基礎」(『FPGA Programming Blockset ガイド』)を参照してください。



Release 2022-A までは、最初にインターフェースで信号値をスケールし、その後、スケールされた値を FPGA テストアクセスに置き換えることができました。

**Buffer64 ブロックから Bus In ブロックへの置き換え** RTI FPGA Programming Blockset 3.12 以降では、Buffer64 In/Buffer64 Out ブロックの転送モードは、MicroAutoBox III/SCALEXIO フレームワーク用の Bus In/Bus Out ブロックに置き換えられます。アップデートの過程で、バス転送モードを使用する Buffer64 In/Buffer64 Out ブロックは、自動的に Bus In/Bus Out ブロックに置き換えられません。

**バスデータ送信メソッドの名前変更** FPGA Programming Blockset 23-A では、Synchronous to Read\_Req メソッドは、Synchronous to task メソッドという名称になっています。この機能に変更はありません。

**MGT In ブロック移行時に必要となる手直し** Release 2022-B までは、MGT In ブロックは Opto\_Ready ポートを提供します。Release 2023-A からは、Opto\_Ready ポートの信号をユーザクロックドメインで提供できないため、MGT In Opto Ready ブロックが Opto\_Ready ポートを提供します。

移行後は、旧フレームワークの MGT In ブロックを、現フレームワークの MGT In Opto Ready ブロックと MGT In ブロックに手作業で置き換える必要があります。

---

## 関連トピック

### 基本概要

[既存の FPGA モデルの移行と更新 \(『FPGA Programming Blockset ガイド』\)](#)



# MicroAutoBox III のファームウェア

## MicroAutoBox III Firmware 23.2 の新機能

### DS1521 Bus Board

DS1521 Bus Board のファームウェアが、MACsec をサポートするようになりました (Ethernet コントローラがこの機能をサポートしている場合)。

MicroAutoBox III の底面に貼られているタイプラベルには、Ethernet コントローラが MACsec をサポートしているかどうかは脚注に示されています。

### System Information

<b>MicroAutoBox III</b>		
Type	1403/1521 <sup>1)</sup>	1) MACsec
Serial	1007747	
Host-MAC	64:4D:70:00:E2:10	





# Model Compare

## 次のステップ

## 本章の内容

Model Compare 2023-B の新機能.....	137
Model Compare 2023-B への移行.....	138

## Model Compare 2023-B の新機能

### モデルの比較を促進

Model Compare 2023-B は、純粋なモデルファイルを直接解析し、比較結果を即座に表示する、ファイルベースの新しい比較方法を提供します。

#### 関連ドキュメント

- 「[New Session: Select Method]ダイアログ」(『Model Compare リファレンス』[📖](#))
- 「[New Session: Specify Model Files]ダイアログ」(『Model Compare リファレンス』[📖](#))
- 「高速ファイルベース比較手法を使用した比較セッションの作成方法」(『Model Compare ガイド』[📖](#))

### TargetLink サポートを内蔵した Textconv ドライバ

Model Compare 2023-B では、Model2TextConverter(textconv ドライバ)が、TargetLink ブロックのプロパティとパラメータを詳細にサポートしています。

#### 関連ドキュメント

- 「Model Compare の Textconv ドライバとしての設定」(『Model Compare ガイド』[📖](#))

### マッチングアルゴリズムの改善

Model Compare 2023-B では、マッチングアルゴリズムがさらに改良されています。特に、プロパティの変更が少なく、位置の変更が大きいモデル要素では、マッチングがさらにうまく機能するようになり、手作業によるアラインメントの必要性が少なくなりました。

### ユーザマニュアルの改善

Model Compare 2023-B では、Model Compare のワークフロー統合 (Git の使用など) に関するユーザマニュアルが改善されています。

#### 関連ドキュメント

- 「一般的なワークフローでの Model Compare の使用」(『Model Compare ガイド』[📖](#))

## Model Compare 2023-B への移行

### 質問

ModelCompare.exe の -textconv オプションは、DS.MOC.Model2TextConverter.CLI.exe に置き換えられます。そのため、ModelCompare.VCSTextConv.cmd を使用することをお勧めします。既に使っている場合は、調整の必要はありません。

#### 注記

このバージョンの Model Compare で作成された XML ダンプファイルのみを使用することをお勧めします。それ以外の場合、マージ操作と Model Compare の新機能の一部が使用できなくなります。

# Model および Sensor Interface Blockset

## 次のステップ

## 本章の内容

[Model and Sensor Interface Blockset 2023-B の新機能](#)..... 139

[Model and Sensor Interface Blockset 2023-B への移行](#)..... 139

## Model and Sensor Interface Blockset 2023-B の新機能

### Model and Sensor Interface Blockset

Model and Sensor Interface Blockset 2023-B には新機能はありません。

## 関連トピック

## 基本概要

[Model and Sensor Interface Blockset の機能](#) (『Model and Sensor Interface Blockset マニュアル』)

## Model and Sensor Interface Blockset 2023-B への移行

### MicroAutoBox III

MicroAutobox III プラットフォームは、Model and Sensor Interface Blockset ではサポートされていません。

MicroAutobox III でブロックセットを使用する場合の詳細については、dSPACE サポートにお問い合わせください。

### 前のリリースからの移行

**前のリリースから dSPACE Release 2020-B への移行** dSPACE Release 2020-B でバージョン 1.0 以降の Model and Sensor Interface Blockset の以前のリリースを使用して作成したモデルは、開いたときに自動的に移行されます。移行の正常完了または失敗を確認するために、MATLAB コマンドウィンドウと移行ログにメッセージが表示されます。

**Model and Sensor Interface Blockset Solution からの移行** dSPACE Release 2020-B 以前の Model and Sensor Interface Blockset Solution で作成したモデルは、手作業で移行する必要があります。ブロックセットソリューションからのブロックを使用するモデルを開くと、Model and Sensor Interface Blockset の最新のリリースのブロックを使用するためにモデルを移行する手順が記載されたメッセージが表示されます。

# ModelDesk

## 次のステップ

## 本章の内容

ModelDesk 2023-B の新機能.....	141
ModelDesk 2023-B への移行.....	142

## ModelDesk 2023-B の新機能

### 全般

ModelDesk のコアテクノロジーが変更されました。

- 将来の使いやすさが確保された最新のテクノロジーです。
- アプリケーションと、ユーザインターフェースの一部が更新されています。
- 最新のオペレーティングシステムと、.NET フレームワークの新しいバージョンに対応しています。
- 該当する部分ではユーザアクセスが統一されています。

### Road Generator

#### OpenDRIVE ファイル

- OpenDRIVE フォーマットのバージョン 1.6 の道路をインポートできます。
- OpenDRIVE フォーマットの道路のエクスポート品質が向上しています。

### 関連トピック

#### 基本概要

[OpenDRIVE 形式での道路の使用 \(『ModelDesk 道路の作成』📖\)](#)

## ModelDesk 2023-B への移行

---

<b>プロジェクトの移行</b>	ModelDesk 2023-B 以降では、ModelDesk バージョン 5.2 (dSPACE Release 2019-B) より後のバージョンで作成したプロジェクトのみ移行することができます。
<b>移行したプロジェクトでの Release 2020-B の ASM の使用</b>	ModelDesk プロジェクトや対応する ASM モデルを移行しても、レイアウトでプロットに使用したすべての信号は接続されたままになります。移行した ASM モデルを Release 2020-B で作成した ASM モデルに置き換える場合は、信号を再接続する必要があります。Release 2020-B で作成した ASM モデルは、モデルの各部で使用できる ASMSignalInterface ブロックを使用します。
<b>ASM プロジェクト</b>	ASM プロジェクトに基づく新規 ModelDesk プロジェクトの作成は、DS1006 ターゲットプラットフォームでサポートされなくなりました。ただし、DS1006 プラットフォームは今後も ModelDesk によってサポートされます。DS1006 プラットフォームで動作するリアルタイムアプリケーションをパラメータ化することも引き続き可能です。

---

# Model Interface Package for Simulink

## 次のステップ

## 本章の内容

Model Interface Package for Simulink 2023-B の新機能..... 143

Model Interface Package for Simulink 2023-B への移行..... 144

## Model Interface Package for Simulink 2023-B の新機能

### Simulink root In および Simulink root Out ブロックのサ ポート強化

Model Interface Package for Simulink は、ブロックおよび信号 ID をモデルのすべての外部インターフェースに割り当てます。<sup>Simulink®</sup> In、Out、Bus Element In、および Bus Element Out ポートでは、これらの ID は改良されたスキーマを使用して作成されるようになりました。このため、次のような機能強化があります。

- これらのブロックに接続されている Simulink バス信号は、ConfigurationDesk または VEOS プロジェクトの接続を失うことなく変更することができます。
- 入力ポートと出力ポートに同一の名前を使用することで、モデル通信の際に一致するポートを簡単に特定できます。

### 注記

- これらの機能強化は、VEOS にも有効です。
- 移行の詳細については、「[Model Interface Package for Simulink 2023-B への移行](#)」(144 ページ)を参照してください。

### 新しい MicroLabBox II プラット フォームのサポート

MicroLabBox II と SCALEXIO は、同じプロセッサアーキテクチャと Linux64 ベースのリアルタイム OS を搭載しています。したがって、SCALEXIO Linux 64 ビットターゲットアーキテクチャ用に生成された Simulink モデルと SIC ファイル(プラットフォーム識別子: SCALEXIO\_LNX64)は、MicroLabBox II の ConfigurationDesk アプリケーションでも使用できるようになりました。

---

新しい VEOS_LINUX32_GCC プラットフォーム識別子	Model Interface Package for Simulink が、Windows 32 ビット上の VEOS で使用される SIC ファイル用の新しい VEOS_LINUX32_GCC プラットフォーム識別子を提供するようになりました。
無制限配列変数のサポート	Model Interface Package for Simulink は、Simulink モデル内の無制限配列変数をサポートします。ただし、モデルポートブロックおよび Simulink ルートレベルの Inport または Outport ブロックでは、無制限の配列変数を使用することはできません。
MATLAB の互換性	Model Interface Package for Simulink 2023-B は、以下の MATLAB®リリースと互換性があります。 <ul style="list-style-type: none"><li>▪ MATLAB R2023b</li><li>▪ MATLAB R2023a</li><li>▪ MATLAB R2022b</li><li>▪ MATLAB R2022a</li></ul>

---

## Model Interface Package for Simulink 2023-B への移行

---

Simulink ルートレベルの入力ポートと出力ポートの新しい ID 動作への移行	Model Interface Package for Simulink は、ルートレベルの Simulink® In、Out、Bus Element In、および Bus Element Out ブロックに新しい ID を割り当てます。その結果、これらの Simulink ブロックに関連する ConfigurationDesk または VEOS プロジェクトのモデルポートへのすべての接続は、モデルの解析またはそれぞれの SIC ファイルの更新後に未解決になります。新しい ID を持つモデルポートを再接続する必要があります。
---	---

### 注記

MATLAB の `dsmptb_pref()` API コマンドの新しい `UseClassicIdConventionForRootPorts` パラメータを `True` に設定することで、既存のプロジェクトの接続を維持するために古い ID スキーマを復元できます。

```
dsmptb_pref('Set', 'UseClassicIdConventionForRootPorts', true);
```

---

サポートされていたプラットフォームの廃止	SCALEXIO 32-bit Linux ターゲットプラットフォーム (SCALEXIO_LNX) のサポートは、Model Interface Package for Simulink 2023-B で廃止されました。SCALEXIO 64-bit Linux ターゲットプラットフォーム (SCALEXIO_LNX64) は引き続き使用可能です。
----------------------	--



# MotionDesk

## 次のステップ

## 本章の内容

MotionDesk 2023-B の新機能.....	145
MotionDesk 2023-B への移行.....	145

## MotionDesk 2023-B の新機能

### MotionDesk

MotionDesk 2023-B には新機能はありません。

## MotionDesk 2023-B への移行

### センサシミュレーションの廃止

MotionDesk Release 2023-A より、MotionDesk に基づくセンサシミュレーションは、すべてのセンサタイプではサポートされなくなりました。

**AURELION** dSPACE AURELION は、カメラ、レーダー、および LiDAR センサを使用した ADAS/AD シミュレーションで運転機能やセンサ出力の検証を行うためのツールで、シミュレーションアニメーション用の MotionDesk と Sensor Simulation の後継にあたります。

詳細および AURELION への移行の準備については、<https://www.dspace.com/en/pub/home/support/kb/faqs/faq433.cfm> を参照するか、dSPACE サポートにお問い合わせください。

### 廃止の予定

**MotionDesk** MotionDesk および MotionDesk Blockset は、リリース 2023-B 以降に廃止されます。廃止までの間、MotionDesk はカメラセンサのシミュレーションをサポートします。MotionDesk を使用したレーダーセンサと LiDAR センサはサポー

ト対象外となりました。詳細については、<http://www.dspace.jp/go/elamd> を参照してください。

---

#### 無限の地表と水平線の使用

MotionDesk 4.0 以前では、風景の仮想世界は地表およびドーム 3D オブジェクトでビルドされていました。

古いシーンの無限の地表と無限の空を使用する場合は、まずこれらの 3-D オブジェクトを削除する必要があります。

高度な照明モードでは、ドームに使用される静的オブジェクトは仮想世界のビルドに適していません。無限の空を使用する必要があります。

---

#### カスタムオブジェクト VRML ファイルの移行

VRML2 形式の 3-D カスタムオブジェクトを MotionDesk にインポートする場合は、最初に、Release 2022-A までの MotionDesk を使用して VRML2 ファイルを COLLADA (\*.dae) 形式に変換する必要があります。その後、Release 2022-B 以降で COLLADA ファイルをインポートし、使用することができます。

このファイルは、Library Manager Tool の Import Objects Dialog で変換できません。

---

#### 前のリリースからの移行

MotionDesk は、4 バージョン前の MotionDesk の実験までサポートしています。

# Real-Time Testing

## 次のステップ

## 本章の内容

Real-Time Testing 2023-B の新機能.....	147
Real-Time Testing 2023-B への移行.....	147

## Real-Time Testing 2023-B の新機能

### 新しいプラットフォームのサポート

Real-Time Testing は、以下をサポートしています。

## Real-Time Testing 2023-B への移行

### シミュレーションアプリケーションの変数

TRC ファイルには、Real-Time Testing でアクセス可能なシミュレーションアプリケーションの変数が含まれています。有理スケールリングが含まれている変数は、Real-Time Testing でサポートされなくなりました。

### 互換性のない BCG ファイル

Windows PC で生成した BCG ファイルは、Linux PC で使用することはできません。逆も同じです。リアルタイムテストを管理するために使用するオペレーティングシステムで BCG ファイルを生成します。

SCALEXIO では Real-Time Testing 4.4 以降、VEOS では Real-Time Testing 4.2 以降にのみ適用されます。内部 Python インタープリタの内部バージョンが 2.7.11 から 3.6.4 に変更されました。Python 2.7.11 の構文の古いスクリプトを使用するには、スクリプトを Python 3.6.4 の構文に移行する必要があります。Python スクリプトの移行の詳細については、<http://www.dspace.jp/go/Python36Migration> を参照してください。



# RTI/RTI-MP および RTLib

## 次のステップ

## 本章の内容

RTI/RTI-MP および RTLib の新機能.....	149
RTI/RTI-MP および RTLib の移行上の注意点.....	149

## RTI/RTI-MP および RTLib の新機能

### RTI/RTI-MP の新機能

RTI および RTI-MP には、以下の新機能があります。

- MATLAB® R2023b のサポート。

## RTI/RTI-MP および RTLib の移行上の注意点

### 以降の MATLAB バージョンで修正された機能

**以降の MATLAB バージョンへの切り替え** 新しい MATLAB バージョンをインストールする場合、以前インストールした MATLAB バージョンから設定の一部が引き継がれます。新しい MATLAB バージョンまたは dSPACE Release に切り替える場合、Simulink モデルの予期しない動作を防ぐために、モデルの使用を開始する前に MATLAB および Simulink のプリファレンスを必ずデフォルト値にリセットしてください。

MATLAB バージョンまたは dSPACE Release を変更する場合、以前のバージョンの MAT ファイルに保存されたコンフィギュレーションセットによって問題が発生することがあります。そのため、Release バージョンを変更する場合はこれらのコンフィギュレーションセットを再度作成することをお勧めします。



# RTI Bypass Blockset

## RTI Bypass Blockset 2023-B への移行

### 追加の移行手順が必要な場合

RTI CAN MultiMessage Blockset 2023-A より前のバージョンから RTI Bypass Blockset 2023-B に移行し、既存のモデルを再利用するには、追加の移行手順が必要な場合があります。移行の詳細については、「[移行手順の変更点](#)」(『RTI Bypass Blockset リファレンス』[📖](#))を参照してください。

### 以前の RTI Bypass Blockset バージョン 2023-A、2022-B、3.x および 2.x のモデルの使用

最新の Release には、以前のブロックセットバージョン 2023-A、2022-B、3.x および 2.x と互換性のある RTI Bypass Blockset 2023-B が含まれています。ただし、いくつかの注意事項があります。

- *RTI Bypass Blockset 2.5 以前のモデルを使用する場合*

データ管理が以前の RTI Bypass Blockset バージョンから変更されています。RTI Bypass Blockset 2.5 以前でビルドされた Simulink モデルを RTI Bypass Blockset 2023-B で開くと、古い Data Dictionary ファイル(ファイル名の拡張子.dd)は、Setup ブロックに格納されている情報を使用して新しい Data Dictionary ファイル(.vdb)に置き換えられます。この手順は、Setup ブロックダイアログを開いて[OK]をクリックして閉じるか、または Read/Write/Upload/Download ブロックダイアログを開いて[Variables]ページの[Fill Variable Selector]をクリックすると自動的に行われます。

RTI Bypass Blockset 2023-B で保存されたモデルを RTI Bypass Blockset 2.5 以前のバージョンで使用する場合、バージョン 2.5 以前のブロックセットに必要なモデルの Data Dictionary ファイル(ファイル名の拡張子.dd)が作成されます。この手順は、Setup ブロックで A2L ファイルを更新するか、または Read/Write/Upload/Download ブロックを開いて[Variables]ページの[Fill Variable Selector]をクリックしたときに行われます。RTI Bypass Blockset 2023-A で作成された Data Dictionary ファイル(.vdb)は、ディスク上にそのまま残ります。

RTI Bypass Blockset で Data Dictionary を再作成するには、Setup ブロックで指定されたデータベースファイルが変更されておらず、指定された場所でアクセス可能なことが必須条件となります。

- *RTI Bypass Blockset 2.6 から RTI Bypass Blockset 2023-A までのモデルを使用する場合*

RTI Bypass Blockset 2.6 から RTI Bypass Blockset 2023-A まででビルドされた Simulink モデルを RTI Bypass Blockset 2023-B で開くと、古い Data Dictionary

ファイルが新しい Data Dictionary ファイルに置き換えられます。ただし、新しい Data Dictionary ファイルを以前のバージョンの RTI Bypass Blockset で使用することはできません。RTI Bypass Blockset 2.6 から RTI Bypass Blockset 2023-A まででビルドされたモデルを再利用するには、Setup ブロックで指定されているデータベースファイル (A2L ファイル) を再インポートして、以前のバージョンの RTI Bypass Blockset で適切なデータベースを作成する必要があります。



# RTI CAN MultiMessage Blockset

## 次のステップ

## 本章の内容

RTI CAN MultiMessage Blockset 2023-B の新機能.....	153
RTI CAN MultiMessage Blockset 2023-B への移行.....	153

## RTI CAN MultiMessage Blockset 2023-B の新機能

### AUTOSAR Classic Platform R22-11 のサポート

RTI CAN MultiMessage Blockset は、CAN ネットワークの記述用として、AUTOSAR Classic Platform Release R22-11 に準拠した AUTOSAR System Template もサポートします。ただし、AUTOSAR Classic Platform Release R22-11 の新機能はサポートされていません。

「[General Settings]ページ (RTICANMM MainBlock)」(『RTI CAN MultiMessage Blockset リファレンス』[📖](#))を参照してください。

## RTI CAN MultiMessage Blockset 2023-B への移行

### 廃止の予定

RTI CAN MultiMessage Blockset for SCALEXIO は、Release 2024-B 以降、廃止される予定です。詳細については、<http://www.dspace.jp/go/elaRTIxxMMBS> を参照してください。

RTI CAN MultiMessage Blockset は、新しい SCALEXIO プロジェクトでは使用できないようになりました。新規プロジェクトでは、多数の機能を 1 つのツールにまとめた Bus Manager を使用することをお勧めします。

---

**RTI CAN MultiMessage Blockset の以前のバージョンで作成したモデルの使用**

RTI CAN MultiMessage Blockset の以前のバージョンで作成されたモデルを再利用するには、CAN の設定に変更を加える前に、すべての RTICANMM ブロックの S-function を更新して保存する必要があります。

モデル内のすべての RTICANMM ブロックに対して新しい S-function を一度に作成するには、モデルを開いた後で次のいずれかを実行します。

- MATLAB コマンドウィンドウに `rtimmsu_update('System', bdroot)` と入力します。

このコマンドおよびオプションの詳細を確認するには、MATLAB コマンドウィンドウに `help rtimmsu_update` と入力します。

- 「[RTICANMM GeneralSetup](#)」(『[RTI CAN MultiMessage Blockset リファレンス](#)』[📖](#))ブロックの[Options]メニューから[Create S-Function for all CAN Blocks]コマンドを選択します。

詳細については、「[RTICANMM に関する制限事項](#)」(『[RTI CAN MultiMessage Blockset リファレンス](#)』[📖](#))を参照してください。

---

**追加の移行手順が必要な場合**

RTI CAN MultiMessage Blockset 2023-A より前のバージョンから RTI CAN MultiMessage Blockset 2023-B に移行し、既存のモデルを再利用するには、追加の移行手順が必要な場合があります。移行の詳細については、「[移行手順の履歴](#)」(『[RTI CAN MultiMessage Blockset リファレンス](#)』[📖](#))を参照してください。

# RTI LIN MultiMessage Blockset

## 次のステップ

## 本章の内容

RTI LIN MultiMessage Blockset 2023-B の新機能.....	155
RTI LIN MultiMessage Blockset 2023-B への移行.....	155

## RTI LIN MultiMessage Blockset 2023-B の新機能

### AUTOSAR Classic Platform R22-11 のサポート

RTI LIN MultiMessage Blockset は、LIN ネットワークの記述用として、AUTOSAR Classic Platform Release R22-11 に準拠した AUTOSAR System Template もサポートします。ただし、AUTOSAR Classic Platform Release R22-11 の新機能はサポートされていません。

「[General Settings]ページ (RTILINMM MainSetup)」(『RTI LIN MultiMessage Blockset リファレンス』[📄](#))を参照してください。

## RTI LIN MultiMessage Blockset 2023-B への移行

### 廃止の予定

RTI LIN MultiMessage Blockset for SCALEXIO は、Release 2024-B 以降、廃止される予定です。詳細については、<https://www.dspace.jp/go/elaRTIxxMMBS> を参照してください。

RTI LIN MultiMessage Blockset は、新しい SCALEXIO プロジェクトでは使用できないようになりました。新規プロジェクトでは、多数の機能を 1 つのツールにまとめた Bus Manager を使用することをお勧めします。

**RTI LIN MultiMessage Blockset  
の以前のバージョンで作成したモデルの使用**

RTI LIN MultiMessage Blockset の以前のバージョンで作成されたモデルを再利用するには、LIN の設定に変更を加える前に、すべての RTILINMM ブロックの S-function を更新して保存する必要があります。

モデル内のすべての RTILINMM ブロックに対して新しい S-function を一度に作成するには、モデルを開いた後で次のいずれかを実行します。

- MATLAB コマンドウィンドウに `rtimmsu_update('System', bdroot)` と入力します。

このコマンドおよびオプションの詳細を確認するには、MATLAB コマンドウィンドウに `help rtimmsu_update` と入力します。

- 「RTILINMM GeneralSetup」(『RTI LIN MultiMessage Blockset リファレンス』[📖](#)) ブロックの[Options]メニューから[Create S-Function for all LIN Blocks]コマンドを選択します。

詳細については、「RTI LIN MultiMessage Blockset の制限事項」(『RTI LIN MultiMessage Blockset リファレンス』[📖](#))を参照してください。

# RTI Synchronized Time Base Manager Blockset

## RTI Synchronized Time Base Manager Blockset 2023-B の新機能

---

### MicroLabBox II のサポート

RTI Synchronized Time Base Manager Blockset は、新しい MicroAutoBox II をサポートしています。



# SCALEXIO Firmware

## 次のステップ

## 本章の内容

SCALEXIO Firmware 23.2 の新機能.....	159
SCALEXIO Firmware 23.2 への移行.....	160
SCALEXIO ファームウェアの廃止.....	161

## SCALEXIO Firmware 23.2 の新機能

### 新しくサポートされるハードウェア

SCALEXIO ファームウェアでは、次の新しいハードウェアをサポートします。

- DS6242 D/A Board

DS6242 D/A Board は、シングルスロットの SCALEXIO I/O ボードです。電圧出力 (1 チャンネル最大 20 V、3 チャンネル直列最大 60 V) および電流シンク (1 チャンネル最大 40 mA、10 チャンネル並列最大 400 mA) 用に 10 個のガルバニック絶縁アナログチャンネルを提供します。

- DS6365-PE Board

DS6365-PE Automotive Ethernet Board は、最大 10 Gbit/s の Automotive Ethernet 接続経由で外部デバイスを SCALEXIO プロセッサユニットに接続するための独立した 2 つのポートを備えた PCIe 3.0 x4 ボードです。

- SCALEXIO FSX ハードウェア

SCALEXIO では、SCALEXIO FSX と呼ばれる新しい製品ラインが開発されました。FSX の略語は、故障の挿入 (fault insertion)、シグナルコンディショニング (signal condition)、およびさらなる拡張 (extensions) を意味しています。SCALEXIO 標準 I/O ボードは、ECU 配線の故障のシミュレーションには使用することができません。これまでは、HighFlex ボード (DS2601 など) または MultiCompact ユニット (DS2680 など) を搭載した SCALEXIO システムでしか故障シミュレーションを行うことができませんでした。標準 I/O ボードを FSX ボードで拡張すれば、標準 I/O ボードで故障シミュレーションを行うことも可能になりました。最初のバージョンでは、FSX ボードはカスタマイズされた SCALEXIO システムでのみ使用することができます。

カスタマイズされた SCALEXIO システムとともに、信号リストが提供されます。この信号リストでは、切り替えを許可する障害クラスを設定することができます。信号リストの編集には、Signal List Editor が開発されています。故障は、XIL API EESPort を介して切り替えられます。これは、ControlDesk の Failure Simulation Package で使用され、自動化ツールでも使用することができます。

FSX ボードによる故障シミュレーションは、SCALEXIO の HighFlex および MultiCompact ハードウェアによる故障シミュレーションとは互換性はありません。

詳細については、「FSX ハードウェアによる障害シミュレーション」(『SCALEXIO – ハードウェアおよびソフトウェアの概要』[📖](#))を参照してください。

- DS6511 Connector Panel

SCALEXIO Connector Panel を使用すると、BNC コネクタを介して標準 I/O ボードの I/O チャンネルに簡単かつ便利にアクセスできます。このコネクタパネルは、50 ピンの D-Sub リボンケーブルを介して SCALEXIO I/O ボード (DS6221、DS6241) に接続されています。

## 関連トピック

### 基本概要

[DS6242 D/A Board](#) (『SCALEXIO ハードウェアの設置および設定』[📖](#))  
[DS6365-PE Automotive Ethernet Board](#) (『SCALEXIO ハードウェアの設置および設定』[📖](#))

## SCALEXIO Firmware 23.2 への移行

### 64 ビット Linux ベースのオペレーティングシステムへの移行

dSPACE Release 2022-A では、デフォルトの SCALEXIO ファームウェアは 64 ビット Linux ベースのディストリビューションです。dSPACE Release 2021-B 以前用に作成された次の項目は、SCALEXIO システムとの互換性がなくなったため、dSPACE Release 2022-A 以降に基づくソースコードから(再)ビルドする必要があります。

- リアルタイムアプリケーション
- モデルコンテナに含まれるバイナリファイル (SIC、BSC、FMU、および CTLGZ ファイルなど)
- SIC ファイルと BSC ファイル
- Simulink モデルによって参照されるバイナリライブラリ
- カスタムコード設定またはカスタム I/O ファンクションを介して ConfigurationDesk によって参照されるバイナリライブラリ

大多数の場合、ソースコードの再利用が可能であると見られています。C コードの調整が必要となるケースは、会社またはサードパーティサプライヤによって作成された特殊な機能 (OS 固有の機能など) を使用するカスタムコードに関係がありません。

### Hypervisor の拡張

新しい SCALEXIO Hypervisor Extension がインストールされている場合、dSPACE Release 2020-A 以前用に作成された Linux リアルタイムおよび非リアルタイムアプリケーションだけでなく、SCALEXIO リアルタイムアプリケーションも使用できなくなり



ます。SCALEXIO リアルタイムアプリケーションを再ビルドし、Linux リアルタイムおよび非リアルタイムアプリケーションに合わせて新しいハイパーバイザを調整する必要があります(特に、割り込み、共有メモリ、およびハードウェアリソースの割り当てに関する問題を解決する必要があります)。

## SCALEXIO ファームウェアの廃止

---

### プロセッサユニット

Intel® Core™ i7-860 プロセッサ搭載の SCALEXIO プロセッサユニット (Real-Time PC Version 1.0) は、バージョン 6.0 (dSPACE Release 2022-A) 以降の SCALEXIO ファームウェアでサポートされていません。

### 32 ビットサポートの廃止

64 ビット Linux ベースのリアルタイムオペレーティングシステム (RTOS) が、SCALEXIO システムおよび DS6001 Processor Board のデフォルト RTOS です。32 ビット Linux ベースの RTOS は、サポートされません。



# Signal List Editor

## 概要

Signal List Editor は、SCALEXIO システムを使用した故障シミュレーションシナリオで使用する新しい dSPACE ソフトウェア製品です。

故障シミュレーションは、配線問題が発生する場合、ECU の配線に電氣的故障を挿入し、システムのビヘイビアを解析することを意味します。たとえば、ECU ピンがグラウンドやバッテリー電圧へ短絡する、ECU ピンが接続されていない(断線、開回路)などの状況をシミュレートすることができます。

このようなシナリオでは、Signal List Editor を使用して各信号に対して許可される障害クラスを指定し、試験ソフトウェア(ControlDesk または dSPACE XIL API)を使用して故障をシミュレートします。

### 注記

Signal List Editor は、SCALEXIO FSX ボード(DS6801 Fault Insertion Board、DS6805 Fault Insertion Board など)のみをサポートしています。故障シミュレーションを提供する HighFlex および MultiCompact SCALEXIO ハードウェア(DS2680 I/O ユニットなど)は、Signal List Editor ではサポートされていません。このハードウェアでは、代わりに ConfigurationDesk を使用する必要があります。

## Signal List Editor 2023-B (23.1)の機能

### 主な機能

Signal List Editor の主な機能は、次のとおりです。

- 電氣的故障シミュレーションに必要な設定データを含む信号リストを表示します。
- 新しい外部デバイスや新しい信号を追加して、この信号リストを拡張します。
- ControlDesk などの試験ソフトウェアでシミュレーションする ECU 信号の許容障害クラスを指定します。
- 故障シミュレーションの場合に接続された負荷に対する負荷遮断動作を指定します。

詳細については、「Introduction to the Signal List Editor」(『Signal List Editor Manual』[📖](#))を参照してください。

# SYNECT

## SYNECT 2023-B への移行

### 次のステップ

### 本章の内容

SYNECT 2023-A からの移行.....	166
SYNECT 2023-A からの移行に必要な手順の一覧。	
データベースの移行.....	166
以前の SYNECT バージョンのデータを SYNECT 2023-B で使用するには、SYNECT のデータベースを移行する必要があります。	
SYNECT 2023-A から SYNECT 2023-B へのデータモデルの変更.....	167
SYNECT 2023-A から SYNECT 2023-B へのデータモデルの変更はありません。	

## SYNECT 2023-A からの移行

### Python 3.11 のサポート

SYNECT は、dSPACE Release 2023-B で配布されている Python 3.11 をサポートするようになりました。これにより、pythonnet を使用するスクリプトの移行で問題が発生する可能性があります。「SYNECT 固有の移行の問題(『SYNECT の使用』(38 ページ)を参照)」を参照してください。

## データベースの移行


### 概要

以前の SYNECT バージョンのデータを SYNECT 2023-B で使用するには、SYNECT のデータベースを移行する必要があります。

SYNECT バージョン 2.0 - 2023-A のデータベースを SYNECT 2023-B に移行するには、SYNECT 2023-B の Database Migrator を使用します。

### 注記

SYNECT 2.0 より前のバージョンに移行する必要がある場合は、dSPACE サポートまでお問い合わせください。  
[http://www.dspace.jp/go/jpn\\_supportrequest](http://www.dspace.jp/go/jpn_supportrequest) を参照してください。

データベースの移行に関する基本的な情報および手順については、『以前の SYNECT バージョンからのデータベースの移行』(『SYNECT Server ガイド』)を参照してください。

## SYNECT 2023-A から SYNECT 2023-B へのデータモデルの変更

---

### 概要

SYNECT 2023-A から SYNECT 2023-B へのデータモデルの変更はありません。





# SystemDesk

---

## 次のステップ

## 本章の内容

SystemDesk 2023-B の新機能.....	170
SystemDesk 2023-B への移行.....	174

# SystemDesk 2023-B の新機能

## 次のステップ

## 本章の内容

新しい一般機能.....	170
新しい V-ECU 生成機能.....	171
A2L ファイル処理の改善.....	171

## 新しい一般機能

### SystemDesk 2023-B での Classic Platform のサポート

**モデリングに使用する AUTOSAR リリース** SystemDesk では、AUTOSAR R22-11 Release に準拠したデータモデルを使用して、Classic Platform ソフトウェアおよびシステムアーキテクチャをモデリングすることができます。ただし、SystemDesk では、他の AUTOSAR リリースのデータとも交換することができます。

**データ交換のサポート** SystemDesk は、データ交換用に AUTOSAR R22-11、R21-11、R20-11、R19-11、4.4.0、4.3.1、4.3.0、4.2.2、4.2.1、4.1.3、4.1.2、4.1.1、4.0.3、および 4.0.2 をサポートしています。

### Adaptive Platform のサポート

SystemDesk は、Adaptive Platform ソフトウェアを開発するために AUTOSAR R22-11 のサポートを開始しました。データ交換では、AUTOSAR R22-11、R21-11、R20-11、R19-11、および R19-03 がサポートされています。

### Python のサポート

dSPACE Release 2023-B では、Python 3.11 が提供されるようになりました。これを、Python スクリプトの開発に使用して、COM API を介して SystemDesk を自動化することができます。

SystemDesk のデモスクリプトは、Python 3.11 に合わせて明示的に調整する必要はありませんでした。カスタムの自動化スクリプトについても同様であると想定されます。

API の変更により、以前のバージョンの SystemDesk の Python スクリプトを移行する必要があります。「[API スクリプトの移行](#)」(176 ページ)を参照してください。

## 新しい V-ECU 生成機能

### SystemDesk 2023-B で生成された V-ECU の互換性

SystemDesk では、ビルドおよびシミュレーションする V-ECU を以下のフォーマットでエクスポートすることができます。

- V-ECU インプリメンテーションコンテナ: バージョン 3.3

V-ECU インプリメンテーションコンテナは、dSPACE SIL ソリューションのすべての機能をサポートする dSPACE 固有の形式です。dSPACE VEOS 2023-B を使用してバージョン 3.3 V-ECU のビルドおよびシミュレートを行うことができます。

詳細な互換性情報については、「[VEOS 2023-B の互換性](#)」(255 ページ)を参照してください。

- V-ECU FMU: FMI 2.0/3.0

SystemDesk では、V-ECU FMU をエクスポートして、OEM メーカー／サプライヤ間シナリオ(すべてのパートナーが dSPACE SIL ソリューションを使用するとは限らない状況)をサポートすることができます。FMI 準拠のシミュレータで V-ECU FMU をビルドおよびシミュレートすることができます。

dSPACE FMI サポートに関する詳細情報および最新の互換性情報については、<http://www.dspace.jp/go/FMI-Compatibility> を参照してください。

### IronPython のサポート

SystemDesk 2023-B では、BSW モジュールスクリプト、カスタムモジュール変換スクリプト、およびカスタム検証ルールについて IronPython 3.4 がサポートされるようになりました。しかし、ほとんどの場合、以前の SystemDesk バージョンの IronPython 2.7 スクリプトを移行する必要はありません。

### fmi-ls-xcp レイヤー構造規格のサポート

SystemDesk では、fmi-ls-xcp レイヤー構造規格バージョン 1.0.0-RC1 に準拠する、XCP サービスを内部に実装した V-ECU FMU の生成がサポートされるようになりました。<https://github.com/modelica/fmi-ls-xcp> を参照してください。

## A2L ファイル処理の改善

### 概要

このバージョンの SystemDesk により、A2L ファイルの処理が改善されて、AUTOSAR ワークフローによる SWC の交換と、AUTOSAR 手法による A2L ファイルの生成のサポートが向上し、SWC 内部の計測／適合 (MC) 変数に関する制限が克服されました。

- V-ECU 用の A2L ファイルの生成時に SystemDesk で、AUTOSAR に準拠して、カーブやマップを含むスタティックメモリと定数メモリがサポートされるようになりました。これにより、AUTOSAR arxml ファイルで SWC 内部変数を完全に定義できます。SystemDesk と他の AUTOSAR ツールの間の SWC 記述を、A2L ファイルを追加することなく交換できるようになりました。「[A2L ファイル生成での、スタティックメモリおよび定数メモリのサポート](#)」(172 ページ)を参照してください。

- SystemDesk では、SWC 内部変数が A2L ファイルで記述されている SWC の統合が引き続きサポートされます。A2L ファイルのインポートで、AUTOSAR に準拠して、スタティックメモリと定数メモリが作成されるようになりました。インポートした SWC 内部変数を SystemDesk で編集し、A2L ファイルを追加することなく SWC

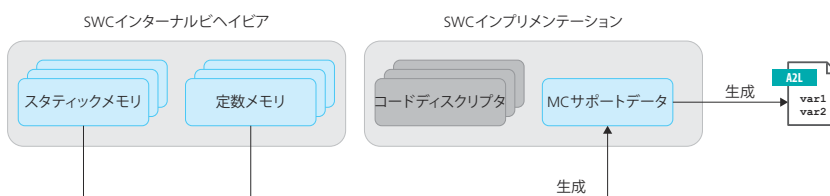
を引き渡すことができます。「[SWC 統合のための A2L ファイルのサポート](#)」(172 ページ)を参照してください。

- SystemDesk では、A2L ファイルのマージが引き続きサポートされます。SWC の実装で既存の A2L ファイルを参照でき、V-ECU では直接参照できます。これらのファイルは、SystemDesk によって生成された A2L フラグメントとマージされます。「[V-ECU A2L ファイルのエクスポート](#)」(173 ページ)を参照してください。

### A2L ファイル生成での、スタティックメモリおよび定数メモリのサポート

A2L ファイル生成時に SystemDesk で、AUTOSAR に準拠してスタティックメモリと定数メモリがサポートされるようになりました。

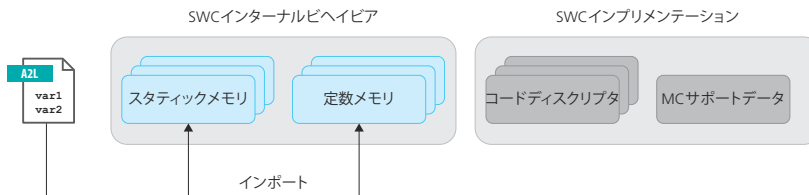
スタティックメモリと定数メモリを使用すると、RTE 機能を使用せずに、SWC の実装で AUTOSAR データタイプを使用できます。これにより、SWC 内部変数の MC サポートが可能になります。



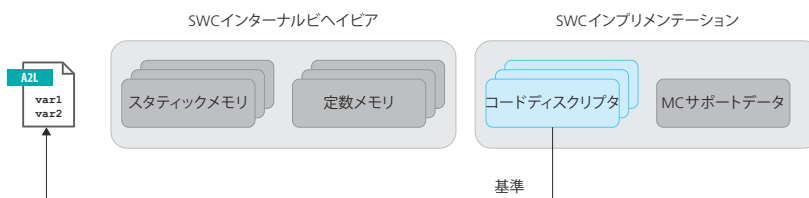
スタティックメモリと定数メモリからの A2L 変数記述の生成は、2 段階のプロセスです。まず、SWC 実装の一部である MC サポートデータが、AUTOSAR モデルのスタティックメモリと定数メモリから生成されます。次に、2 番目のステップとして、A2L 変数の記述が MC サポートデータから生成されます。SystemDesk では、両方のステップにジェネレータを提供しており、これが、V-ECU エクスポート時に自動的に使用されます。

### SWC 統合のための A2L ファイルのサポート

SystemDesk では、AUTOSAR モデルへの SWC の統合時に、SWC 内部変数を含む A2L ファイルのインポートが引き続きサポートされます。このインポートで、AUTOSAR に準拠して、スタティックメモリと定数メモリが作成されるようになりました。これにより、必要に応じて、測定値や特性を SystemDesk で編集できます。



SWC 内部変数の編集を望まない場合は、代わりに A2L ファイルを参照することができます。この場合は、SWC 実装でコードディスクリプタを作成する必要があります。次の図を参照してください。



**V-ECU A2L ファイルのエクスポート** MC ツールを使用して、VEOS シミュレーション中に ECU 変数にアクセスできます。V-ECU 実装コンテナのエクスポート時に SystemDesk は、関連するすべての変数記述をマージします。

**モデルベースの V-ECU** モデルベースの V-ECU を、アプリケーションソフトウェアをテストする場合に使用できます。AUTOSAR モデルのスタティックメモリと定数メモリを定義したり、アプリケーションレイヤー SWC に A2L ファイルをインポートしたり、SWC 実装で A2L ファイルを参照したりすることができます。RTE 用ジェネレータと、dSPACE が SIL テスト用に提供する基本ソフトウェアコンポーネントにより A2L フラグメントが追加されます。

SystemDesk で、以下の変数記述がマージされます。

- 定数メモリとスタティックメモリの MC サポートデータに基づいて SystemDesk で生成される A2L フラグメント。SystemDesk では、最初のステップで MC サポートデータが生成されます。
- SWC 実装で参照する A2L ファイル。
- RTE および基本ソフトウェアの MC サポートデータに基づいて SystemDesk で生成される A2L フラグメント。
- V-ECU で直接参照する A2L ファイル。

**コードベースの V-ECU** コードベースの V-ECU を、量産 BSW で ECU 統合が実行されている場合に使用できます。A2L ファイルを含む、関連するソフトウェアパッケージを V-ECU に追加します。dSPACE では、マイクロコントローラ抽象化レイヤー (MCAL) 用のモジュール構成を提供しており、これで、コードと A2L フラグメントを生成できます。

SystemDesk で、以下の変数記述がマージされます。

- ソフトウェアパッケージの A2L ファイル。
- MCAL モジュール用に SystemDesk で生成される A2L フラグメント。

## 移行

A2L ファイル処理の改善は、SystemDesk の一部ではなくなった変数用カスタムデータベースに代わるものです。

移行には、次のような側面があります。

- プロジェクトの移行は、新しいバージョンの SystemDesk でプロジェクトを開くと自動的に実行されます。インポートされた A2L ファイルはエクスポートされ、参照されます。詳細については、「[インポートされた A2L ファイルを含むプロジェクトの移行](#)」(175 ページ)を参照してください。
- SWC 内部変数にアクセスする API スクリプトを移行する必要があります。詳細については、「[SW インターナルビヘイビアの内部変数の処理](#)」(179 ページ)を参照してください。

## SystemDesk 2023-B への移行

### 次のステップ

### 本章の内容

dSPACE Release 2023-B の廃止.....	174
SystemDesk 2023-B への移行.....	175
API スクリプトの移行.....	176

## dSPACE Release 2023-B の廃止

### 廃止された部分的な AUTOSAR Import および Export コマンド

以下に示す、ソフトウェアコンポーネントの部分的な AUTOSAR Import および Export コマンドは、SystemDesk 2023-B で使用できなくなりました。

- Import – AUTOSAR、Update Component Ports
- Import – AUTOSAR、Update Internal Behavior
- Export – AUTOSAR、Ports Only
- Export – AUTOSAR、Internal Behavior Only

SystemDesk は、代わりに使用できるスプリッタブルのサポートを提供しています。AUTOSAR エレメントをマスターファイルに割り当て、ソフトウェアコンポーネントと内部動作を必要に応じて別のファイルに分割する必要があります。

### IronPython スクリプトのデバッグなし

SystemDesk では、次のような使用事例向けの IronPython がサポートされていません。

- *BSW* モジュールスクリプト: コンテキストメニューコマンドおよび[Auto Configure and Generate]コマンドを介して、アプリケーションソフトウェアに従って RTE および BSW モジュールを設定します。
- *カスタム変換スクリプト*: データを失うことなく、インポート/エクスポートシナリオ用のモジュールコンフィギュレーションを変換します。
- *カスタム妥当性確認ルール*: SystemDesk プロジェクト要素の妥当性確認のカスタムルールを指定します。

以前のバージョンの SystemDesk では、Microsoft Visual Studio を使用してスクリプトをデバッグすることができました。また、必要に応じて、スクリプトにブレイクポイントを追加してデバッグすることができました。しかし、SystemDesk 2023-B では、IronPython スクリプトのデバッグがサポートされなくなりました。

**ECU コンフィギュレーションに関するビューの廃止**

ECU コンフィギュレーションマネージャで、さまざまなビューがサポートされなくなりました。

AUTOSAR ビューと Dependency ビューのどちらに切り替えることもできません。ただし、ECU を設定するための SystemDesk のすべての機能は、デフォルトの Flat List ビューで使用できます。

## SystemDesk 2023-B への移行

**プロジェクトの自動移行**

SystemDesk 2023-B では、SystemDesk 5.6 および 2022-B の SDP プロジェクトファイルが起動時に自動的に移行されます。

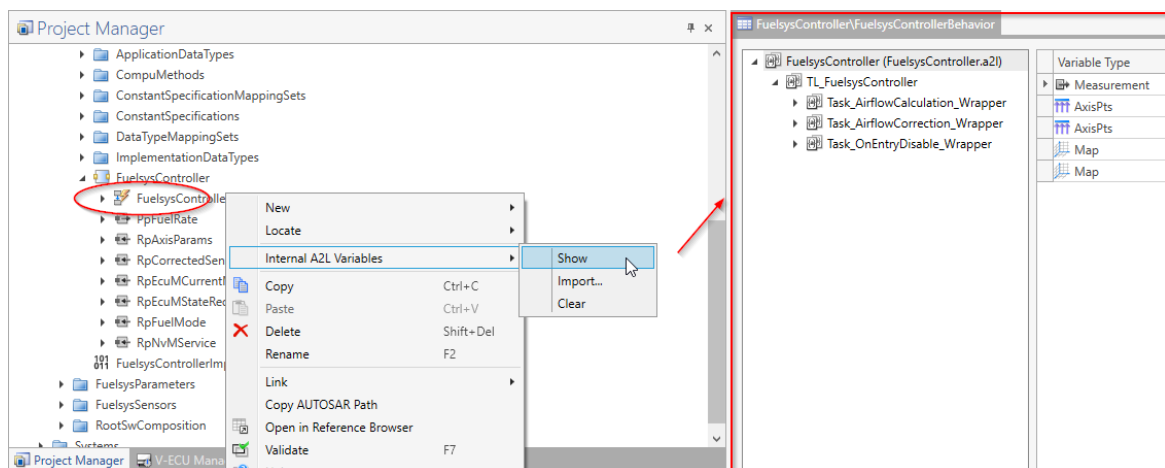
**注記**

SystemDesk 5.6 または 2022-B の最新パッチをインストールすることをお勧めします。その後、移行する SDP プロジェクトファイルを保存してから、SystemDesk 2023-B で開いてください。

**インポートされた A2L ファイルを含むプロジェクトの移行**

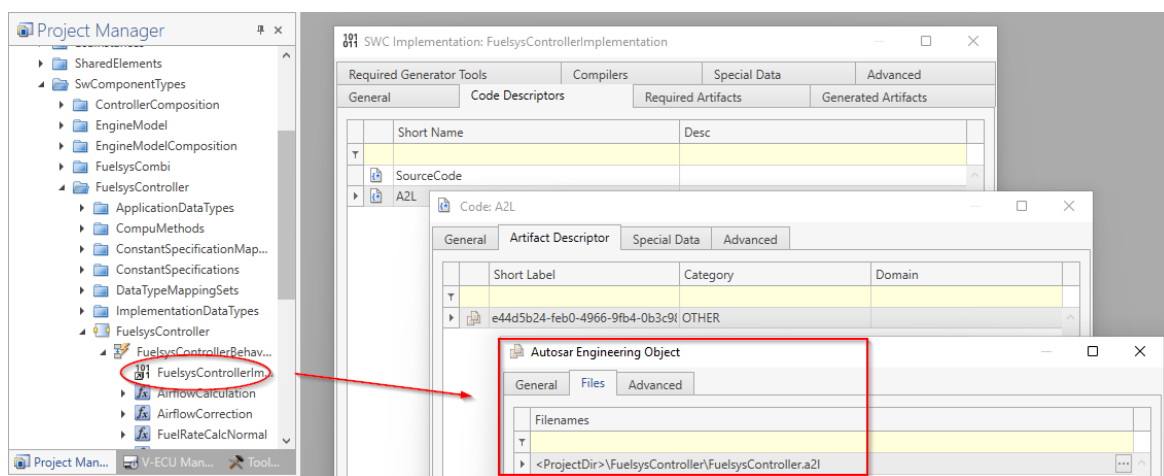
このバージョンの SystemDesk では、A2L ファイル処理が変更されました。これは、1 つ以上の A2L ファイルが SWC インターナルビヘイビアでインポートされた SystemDesk プロジェクトの移行に、次のような影響を及ぼします。

**古いプロジェクト** 以下の図は、SystemDesk 2022-B の AR Fuelsys デモの SWC インターナルビヘイビアでの内部変数を示しています。



**移行されたプロジェクト** SystemDesk2023-B では、SystemDesk 5.6 および 2022-B の SDP プロジェクトファイルが自動的に移行されます。SWC インターナルビヘイビアの内部変数は、SystemDesk プロジェクトファイルを含むフォルダのサブフォルダにある A2L ファイルに自動的にエクスポートされます。これらの A2L ファイルは、それぞれの SWC 実装にリンクされています。これは、SystemDesk 2023-B で参照されるようになった、V-ECU にインポートされた A2L ファイルにも適用されます。

以下の図は、SystemDesk 2023-B で移行された AR Fuelsys デモプロジェクトを示しています。



## API スクリプトの移行

### 文字列プロパティの処理

文字列プロパティの処理を調整する必要があります。Python の None キーワードは使用できなくなりました。代わりに pythoncom.Nothing プロパティを使用できません。

**古いスクリプト** 次のリストは、None キーワードの使用を示します。

```
activeProject = systemDeskApplication.ActiveProject
root = activeProject.RootAutosar
packages = root.ArPackages
for package in packages.Elements:
    package.SdDescription = None
```

**新しいスクリプト** 以下のリストは、pythoncom モジュールのインポートと、pythoncom.Nothing プロパティの使用を示します。

```
import pythoncom

activeProject = systemDeskApplication.ActiveProject
root = activeProject.RootAutosar
packages = root.ArPackages
for package in packages.Elements:
    package.SdDescription = pythoncom.Nothing
```

### 例外処理

- C#の使用時は例外処理を調整する必要があります。これは、現在、COM 自動化では、例外タイプに関して限られた情報のみが返されるためです。したがって、ValueNullException や AutomationLockedException などの特定の例外タイプには対応できません。ただし、例外メッセージを使用すると、特定の例外に対応できます。



- Python の使用時の例外処理は変わっていません。これは、特定の例外タイプが提供されることはなく、常に例外メッセージを使用して、特定の例外に対応する必要があったためです。

**古いリスト** 次のリストは、例外タイプを使用した例外処理を示しています。

```
try
{
    bool demo = runnable.CanBeInvokedConcurrently;
}
catch (ValueNullException e)
{
    // do something
}
```

**新しいリスト** SystemDesk では、すべての COM API の例外について System.Exception が返されます。次のリストは、例外メッセージを使用した例外処理を示しています。

```
try
{
    bool demo = runnable.CanBeInvokedConcurrently;
}
catch (System.Exception e)
{
    if (e.Message.Equals("The value is not set. ")) {
        // do something
    }
}
```

## SystemDesk を自動化するスクリプトの移行

SystemDesk 2023-B 以降、SystemDesk API が変更されました。SystemDesk 2023-B と比較していくつかのインターフェースが追加され、特定のインターフェースが変更されました。

詳細については、dSPACE Help でのみ参照できる『*SystemDesk API Reference*』の「*API Changes from SystemDesk 2022-B to SystemDesk 2023-B*」を参照してください。

**IPortInterface.IsService プロパティのタイプの変更** AUTOSAR R22-11 のサポートで導入された SystemDesk API の変更点の 1 つは、ソフトウェアコンポーネントテンプレートの IPortInterface の IsService プロパティに関するものです。インターフェース、または ISenderReceiverInterface や IClientServerInterface などの類似インターフェースを使用してスクリプトを調整する必要があります。以下のリストは、スクリプトに必要な調整を示しています。

**古いリスト** 以下のリストは、以前のバージョンの SystemDesk での IsService プロパティへのアクセスを示しています。

```
# Setting the IsService property
interface.IsService = True

# Testing if the IsService property is specified
interface.IsIsServiceDefined

# Getting the IsService property
v = interface.IsService
```

**新しいリスト** 以下のリストは、SystemDesk 2023-B での IsService プロパティへのアクセスを示しています。

```
# Setting the IsService property
interface.SetNewIsService().SetValue(True)

# Testing if the IsService property is specified
interface.IsService is not None and
interface.IsService.IsCalculatedValueDefined

# Getting the IsService property
v = interface.IsService.CalculatedValue
```

*IPortInterface*については、dSPACE ヘルプでのみ参照できる『*SystemDesk API Reference*』を参照してください。

## AUTOSAR エLEMENTの作成

AUTOSAR R22-11 で導入されたELEMENTでは、パッケージでの AUTOSAR ELEMENTの作成が変更されました。ELEMENTは、このバージョンの SystemDesk で追加された汎用メソッドでのみ生成できます。ただし、AUTOSAR R21-11 以前の AUTOSAR ELEMENTを作成するスクリプトを移行する必要はありません。

**古いスクリプト** IPackageableElements インターフェースは、AUTOSAR ELEMENTを作成するためのELEMENT固有のメソッドを提供します。ICompuMethod インターフェースの作成については、以下のリストを参照してください。

```
activeProject = systemDeskApplication.ActiveProject
root = activeProject.RootAutosar
packages = root.ArPackages
package = packages.Item("PackageableElements")
compuMethod = package.Elements.AddNewCompuMethod("myCompuMethod")
```

自動化インターフェースは、ELEMENT固有の以下のプロパティおよびメソッドを提供します。

- <Elements>
- AddNew<Element>(ShortName)
- AddNew<Element>Range(ShortName[])

**新しいスクリプト** 以下のリストは、このバージョンの SystemDesk で追加された汎用メソッドによる AUTOSAR ELEMENTの作成を示します。汎用メソッドは、すべての AUTOSAR ELEMENTに使用できます。

```
activeProject = systemDeskApplication.ActiveProject
root = activeProject.RootAutosar
packages = root.ArPackages
package = packages.Item("PackageableElements")
availableElements = package.Elements.AvailableElementTypes
for element in availableElements:
    myElement = package.Elements.AddNew(element, "my"+element)
```

自動化インターフェースで、次のメソッドが提供されるようになりました。

- AddNew(ElementType, ShortName)
- AddNewRange(ElementType, ShortName[])
- OfType(ElementType)

また、IPackageableElements インターフェースは、パッケージで作成できるすべてのエレメントのインターフェース名を返す AvailableElementTypes プロパティも提供します。

#### 注記

以下のエレメント名を使用して、AddNew および AddNewRange メソッドの ElementType パラメータを指定することができます。

- インターフェース名 (ICompuMethod など)
- クラス名 (CompuMethod など)
- XML 名 (COMPU-METHOD など)

**関連情報** 詳細については、dSPACE ヘルプでのみ参照できる『SystemDesk API Reference』の「IPackageableElements」を参照してください。

## SW インターナルビヘイビアの内部変数の処理

このバージョンの SystemDesk では、SWC インターナルビヘイビアにインポートした A2L 変数の処理が変更されています。A2L ファイルから、SWC インターナルビヘイビアの AUTOSAR スタティックメモリおよび定数メモリに変数をインポートできるようになりました。以前のバージョンの SystemDesk では、A2L ファイルの変数がカスタム変数形式でインポートされていました。

SWC インターナルビヘイビアのインポート済み内部変数へのアクセスを自動化するスクリプトを移行し、古い API を、AUTOSAR モデルの API に置き換える必要があります。

**古い API コマンド** 以下の API は SystemDesk から削除されました。

- *ISwcInternalBehavior::InternalVariables* プロパティ: SWC インターナルビヘイビアの内部変数コレクション<type *IInternalVariables*>を取得するためのプロパティが削除されました。
- *IInternalVariables*: すべてのプロパティとメソッドを含むコレクションが SystemDesk API から削除されました。
- *IInternalVariable*: すべてのプロパティとメソッドを含むインターフェースが SystemDesk API から削除されました。

**AUTOSAR モデル用 API** 次の API を AUTOSAR モデルに使用できます。

- *ISwcInternalBehavior::ConstantMemories* プロパティ: 特性<type *IParameterDataPrototypes*>のリストを取得します。
- *IParameterDataPrototype*: 特定の特性へのインターフェース。
- *ISwcInternalBehavior::StaticMemories* プロパティ: 計測<type *IVariableDataPrototypes*>のリストを取得します。
- *IVariableDataPrototype*: 特定の計測へのインターフェース。

dSPACE ヘルプでのみ参照できる『SystemDesk API Reference』を参照してください。



# TargetLink

## 新しいバージョン管理方式

### 注記

これまで dSPACE 製品で使用していたバージョン管理は、カレンダーベースのバージョン管理方式に変更されました。「[新しいバージョン管理方式](#)」(「[一般的な機能拡張および変更](#)」(13 ページ)) を参照してください。

## 次のステップ

### 本章の内容

TargetLink 2023-B の新機能.....	182
TargetLink 2023-B および TargetLink Data Dictionary 2023-B への移行.....	200
今後の TargetLink バージョンでの変更予定.....	250

### 他章の参照情報

#### TargetLink 新機能と移行手順

TargetLink の各種リリースの新機能、移行手順、廃止、およびコードの変更に関する情報を提供します。

## TargetLink 2023-B の新機能

### 次のステップ

### 本章の内容

Simulink または Stateflow でのモデリング.....	182
Adaptive AUTOSAR.....	187
Classic AUTOSAR.....	189
コード生成のコア機能.....	191
MIL (Model-in-the-Loop) シミュレーション.....	191
ターゲットシミュレーション (PIL).....	192
ユーザビリティ.....	192
コードジェネレータオプション.....	195
API 関数とフックスクリプト.....	196
その他.....	197

## Simulink または Stateflow でのモデリング

### 次のステップ

### 本章の内容

構造体配列のサポートの改善.....	183
動的エレメントの指定.....	183
連立一次方程式の解法.....	184
Ensure Output is Virtual のサポート.....	185
Prelookup ブロックの可変ベクトル幅のサポート.....	185
Stateflow オブジェクトの使用.....	186
機能インターフェースに関する改善.....	186

## 構造体配列のサポートの改善

### バス配列信号のサポート

Release 2023-B では、TargetLink は Switch ブロックのデータ入出力でのバス配列信号をサポートしています。

#### 関連ドキュメント

- 「バス配列信号と構造体配列変数の基礎」(『TargetLink 準備およびシミュレーションガイド』[📖](#))

### アクセス関数を使った構造体配列へのアクセス

TargetLink 2023-B では、アドレスアクセス関数を使用して構造体配列変数にアクセスすることが可能になりました。

#### 関連ドキュメント

- 「使用事例に応じたアクセス関数の種類の特定」(『TargetLink カスタマイズおよび最適化ガイド』[📖](#))
- 「アクセス関数の変数の種類への関連付け」(『TargetLink カスタマイズおよび最適化ガイド』[📖](#))

### 関連トピック

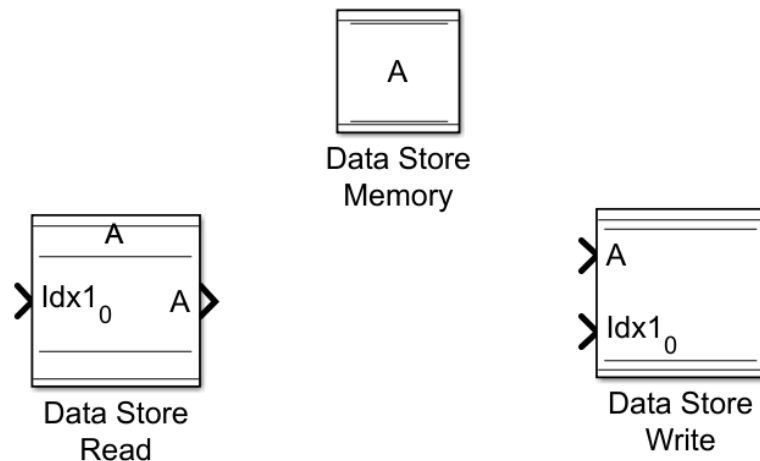
#### 参考文献

Switch ブロック (『TargetLink モデルエレメントリファレンス』[📖](#))

## 動的エレメントの指定

### Data Store Read ブロックおよび Data Store Write ブロックでの動的エレメント指定

TargetLink 2023-B は、Data Store Read および Data Store Write ブロックでのエレメントの選択と割り当てのための Simulink の Enable indexing オプションをサポートしています。このオプションにより、構造体配列変数を含む変数に動的にアクセスすることが可能になりました。



これは、Data Store Read と Selector ブロック、または Data Store Read、Assignment、および Data Store Write ブロックの組み合わせによって実現する機能と同等のものです。

TargetLink 2023-B では、Data Store Read および Data Store Write ブロックの TargetLink ブロックダイアログによって、既に入力したデータを誤って破棄することなく Element Selection/Element Assignment ボタンを使用してエレメント指定用の適切な Simulink ブロックダイアログに簡単に切り替えることができます。

#### 関連ドキュメント

- 「データストアを使用して構造体配列変数へのアクセスをモデリングする方法」(『TargetLink 準備およびシミュレーションガイド』[📖](#))
- 「データストア変数への新しい値の動的割り当ての基礎」(『TargetLink 準備およびシミュレーションガイド』[📖](#))

#### 関連トピック

#### 参考文献

Assignment ブロック (『TargetLink モデルエレメントリファレンス』[📖](#))  
 Data Store Read ブロック (『TargetLink モデルエレメントリファレンス』[📖](#))  
 Data Store Write ブロック (『TargetLink モデルエレメントリファレンス』[📖](#))

## 連立一次方程式の解法

### 分解による連立一次方程式の解法

TargetLink 2023-B では、SOLE Solver ブロックと関連する Linear Algebra Library を使用して、分解によって連立一次方程式 (SOLE) を解くことができます。

原理的には、連立一次方程式 (SOLE) は逆行列によって解くことができます。数学的にはエレガントな方法ですが、これは数値的な観点からは実行可能ではありません。

- 浮動小数点演算を使用する場合、行列の逆変換が不可能であったり、丸め誤差やオーバーフロー／アンダーフローを伴う可能性があります。
- 逆行列を計算するためには、右辺が単位行列である SOLE を解く必要があります (最初の点で示した理由により、解がない場合もあります)。
- SOLE を解く (逆行列を計算してから解を求める) ために、分解を使用する代わりに逆行列を使用すると、一般的に計算回数が多くなります。  
逆行列をまず計算し、それを元の SOLE の右辺と掛け合わせることは、元の SOLE を直接解くよりも常に行列の掛け算が 1 回多くなります。
- SOLE を直接解くために、SOLE の左行列は特別な性質を持つ行列に分解され、それによって SOLE が簡単に解けるようになります。

精度を高めるために、TargetLink は分解によって SOLE を解きます。

有効な分解を達成するには同様の問題が発生するため、TargetLink では、数値的な問題に対応するために使用できるエラーステータスを提供しています。



#### 関連ドキュメント

- 「連立一次方程式の解法」(『TargetLink 準備およびシミュレーションガイド』[📖](#))
- 「SOLE」(『TargetLink デモモデル』[📖](#))
- 「SOLE Solver ブロック」(『TargetLink モデルエレメントリファレンス』[📖](#))

#### 関連トピック

#### 参考文献

[SOLE Solver ブロックの機能](#) (『TargetLink モデルエレメントリファレンス』[📖](#))

## Ensure Output is Virtual のサポート

### Ensure Output is Virtual パラメータのサポート

Release 2023-B では、TargetLink は、条件付きで実行されたサブシステムの拡張されていない Output ブロックで、Simulink®のパラメータ Ensure output is virtual をサポートしています。

Simulink の診断パラメータ Underspecified initialization detection を使用して、従来の初期化モードから簡易初期化モードに切り替えると、MIL シミュレーションと生成されたコードに差異が生じることがあります。たとえば、IF 変数が追加されます。

簡易初期化モードでこれらの差異を回避するには、各 Output ブロックの Simulink ダイアログで Ensure output is virtual チェックボックスを選択します。

#### 関連ドキュメント

- 「[Ensure Output is Virtual](制限事項)」(『TargetLink 制限事項リファレンス』[📖](#))
- MathWorks®による Simulink のドキュメント

## Prelookup ブロックの可変ベクトル幅のサポート

### 幅マクロの使用

TargetLink 2023-B では、Prelookup ブロックは、Axis/Breakpoint 変数の幅マクロとして実装された可変ベクトル幅をサポートしています。

#### 関連ドキュメント

- 「変数の幅の変更可能化」(『TargetLink カスタマイズおよび最適化ガイド』[📖](#))
- 「Prelookup ブロックでの幅可変変数の使用例」(『TargetLink カスタマイズおよび最適化ガイド』[📖](#))

#### 関連トピック

#### 参考文献

[Prelookup ブロック](#) (『TargetLink モデルエレメントリファレンス』[📖](#))

## Stateflow オブジェクトの使用

### [Open Stateflow]ダイアログ

TargetLink 2023-B では、Simulink® Toolstrip の TargetLink タブにある Open Stateflow Dialogs ダイアログを使用して、モデルから直接 Stateflow®オブジェクトダイアログを開くことができます。

#### 関連ドキュメント

- 「Stateflow オブジェクトの使用の基礎」(『TargetLink 準備およびシミュレーションガイド』[📖](#))

## 機能インターフェースに関する改善

### 機能インターフェースにおける MISRA 準拠のモデリング

TargetLink 2023-B を使用すると、const として、関数インターフェースの値または参照ファンクションパラメータを修飾することができます。あらかじめ設定されたシステムテンプレートは、この目的のために以下の構成済み DD VariableClass オブジェクトを提供します。

- FCN\_INPUT
- FCN\_REF\_INPUT

これにより、MISRA に準拠してモデリングを行うことができます。

以下のモデルエレメントは、このモデリングアプローチをサポートしています。

- InPort ブロック(非仮想サブシステムの場合)
- Bus Inport ブロック(非仮想サブシステムの場合)
- ArgIn ブロック
- Stateflow MATLAB 関数の入力を除く、スコープ Input を持つ Stateflow チャートのデータオブジェクト。

#### 関連ドキュメント

- 「関数インターフェースの基礎」(『TargetLink カスタマイズおよび最適化ガイド』[📖](#))
- 「Data Dictionary 内のあらかじめ設定された VariableClass オブジェクトの詳細」(『TargetLink カスタマイズおよび最適化ガイド』[📖](#))

### 関連トピック

#### 参考文献

ArgIn ブロック (『TargetLink モデルエレメントリファレンス』[📖](#))  
Bus Inport ブロック (『TargetLink モデルエレメントリファレンス』[📖](#))  
InPort ブロック (『TargetLink モデルエレメントリファレンス』[📖](#))

# Adaptive AUTOSAR

## 次のステップ

## 本章の内容

サポートされている Adaptive AUTOSAR リリース.....	187
Adaptive AUTOSAR のフレームモデル生成.....	187
C++ 列挙クラスのデータタイプ.....	188
新しい Adaptive AUTOSAR モデリング機能.....	188
AUTOSAR バージョン R22-11 に準拠した変更.....	188
C++ の名前空間.....	189

## サポートされている Adaptive AUTOSAR リリース

TargetLink 2023-B では、以下の Adaptive AUTOSAR リリースをサポートするようになりました。

- R22-11
- R21-11
- R20-11

## Adaptive AUTOSAR のフレームモデル生成

### Adaptive AUTOSAR のフレームモデル生成

TargetLink 2023-B を使用すると、Data Dictionary で AUTOSAR の仕様から Adaptive AUTOSAR フレームモデル を生成できるようになりました。出発点は、Adaptive AUTOSAR 仕様を含む ARXML ファイルです。仕様を Data Dictionary にインポートした後、Adaptive AUTOSAR フレームモデルを生成することができます。TargetLink では、Adaptive AUTOSAR フレームモデル生成をカスタマイズするさまざまな機能を提供しています。

#### 関連ドキュメント

- 「Adaptive AUTOSAR フレームモデルの基礎」(『TargetLink Adaptive AUTOSAR モデリングガイド』[📖](#))
- 「Adaptive AUTOSAR データからフレームモデルの生成を開始する方法」(『TargetLink Adaptive AUTOSAR モデリングガイド』[📖](#))

## C++ 列挙クラスのデータタイプ

### C++ 列挙クラスのデータタイプ

TargetLink 2023-B では、Adaptive AUTOSAR インターフェースで C++ 列挙クラスデータタイプを使用できるようになりました。

#### 関連ドキュメント

- 「TargetLink での列挙データ型の使用の基礎」(『TargetLink 準備およびシミュレーションガイド』[📖](#))
- 「Adaptive AUTOSAR のデータタイプの基礎」(『TargetLink Adaptive AUTOSAR モデリングガイド』[📖](#))

## 新しい Adaptive AUTOSAR モデリング機能

### 永続メモリへのアクセスをモデル化する場合のユーザ指定のデータタイプのサポート

TargetLink 2023-B では、AdaptivePlatformTypes に加えて、永続メモリへのアクセスをモデル化する場合にユーザが指定したデータタイプ(スカラー、構造体、配列、列挙など)もサポートしています。

#### 関連ドキュメント

- 「永続メモリへのアクセス」(『TargetLink Adaptive AUTOSAR モデリングガイド』[📖](#))

### イベントとフィールドのデータストアブロックのサポート

TargetLink 2023-B では、データストアブロックを使用してフィールドにアクセスしたり、イベントを送受信したりできるようになりました。

#### 関連ドキュメント

- 「イベントの受信方法 (Adaptive AUTOSAR)」(『TargetLink Adaptive AUTOSAR モデリングガイド』[📖](#))
- 「イベントの送信方法 (Adaptive AUTOSAR)」(『TargetLink Adaptive AUTOSAR モデリングガイド』[📖](#))
- 「サービスコンシューマとしてフィールドの値を取得または設定する方法」(『TargetLink Adaptive AUTOSAR モデリングガイド』[📖](#))
- 「サービスプロバイダとしてフィールドの値を更新する方法」(『TargetLink Adaptive AUTOSAR モデリングガイド』[📖](#))

## AUTOSAR バージョン R22-11 に準拠した変更

### 新しい API: GetServiceState()

TargetLink 2023-B および Adaptive AUTOSAR バージョン R22-11 では、新しい API 関数 GetServiceState() が ARA アダプタコードで使用されます。この API 関数は、戻り値に基づいて、サービスの状態に関する情報を提供し、サービスが利用可能かどうかを判断することができます。この API 関数は、Adaptive AUTOSAR バージョン R22-11 より前のバージョンの ARA アダプタコードでは使用されません。

## C++の名前空間

### モジュールとソフトウェアコンポーネントのシンボルへの名前空間のアドレス指定

TargetLink 2023-B では、モジュールまたはソフトウェアコンポーネント内のシンボルに名前空間を指定することができます。つまり、その名前空間は、ソフトウェアコンポーネントやモジュール内のそれ自体では名前空間に対処していないすべての関数、変数、データタイプに継承されます。

#### 関連ドキュメント

- 「名前空間の使用の基礎」(『TargetLink カスタマイズおよび最適化ガイド』📖)

### 名前空間修飾への影響

TargetLink 2023-B では、すべてのシンボルを完全な名前空間または縮小された名前空間で修飾するようにコードジェネレータを設定することができます。NamespaceQualification コードジェネレータオプションを Reduced に設定すると、シンボルは周囲の名前空間と異なる名前空間のみで修飾されます。

#### 関連ドキュメント

- 「NamespaceQualification」(『TargetLink モデルエレメントリファレンス』📖)
- 「名前空間の使用の基礎」(『TargetLink カスタマイズおよび最適化ガイド』📖)

### 関連トピック

#### 参考文献

NamespaceQualification (『TargetLink モデルエレメントリファレンス』📖)

## Classic AUTOSAR

### 次のステップ

#### 本章の内容

サポートされる Classic AUTOSAR リリース.....	189
構造体配列のサポートの改善 (Classic AUTOSAR) .....	190
AUTOSAR のインポートおよびエクスポートの改良.....	190

## サポートされる Classic AUTOSAR リリース

TargetLink 2023-B では、Classic AUTOSAR Release R22-11 をサポートしています。

サポートされるすべての Classic AUTOSAR リリースのリストについては、「[サポートされている AUTOSAR リリース](#)」(『TargetLink 相互運用およびデータ交換ガイド』[📖](#))を参照してください。

## 構造体配列のサポートの改善 (Classic AUTOSAR)

### ARXML のインポート/エクスポートとコード生成の改善

TargetLink 2023-B で、構造体配列でタイプ付けされたデータプロトタイプに関して ARXML インポート/エクスポートとコード生成が改善されました。以下のデータプロトタイプについて、ARXML インポート/エクスポートとコード生成が可能になりました。

- PerInstanceMemory
- SharedCalPrm
- PerInstanceCalPrm
- CalPrmElement

## AUTOSAR のインポートおよびエクスポートの改良

### インポート時のパラメータ SWC のサポート

TargetLink 2023-B では、パラメータ SWC を含む ARXML ファイルを Data Dictionary にインポートできるようになりました。AUTOSAR フレームモデル生成時には、これらのパラメータ SWC は考慮されません。

#### 関連ドキュメント

- 「[AUTOSAR ファイルのインポートの基礎](#)」(『TargetLink 相互運用およびデータ交換ガイド』[📖](#))

### DD SwRecordLayoutContent オブジェクトの新しいプロパティ値

TargetLink 2023-B では、DD SwRecordLayoutContent オブジェクトの Kind プロパティに以下の新しいプロパティ値が追加されました。

- RESERVED
- IDENTIFIER

AUTOSAR Release R22-11 では、AUTOSAR インポート/エクスポート時に両プロパティ値が考慮されます。

#### 関連ドキュメント

- SwRecordLayoutContentKindEnum

## コード生成のコア機能

### コード効率性の向上

#### 構造体コンポーネントの並べ替え

TargetLink 2023-B では、構造化 DD Typedef オブジェクトで指定された構造体タイプのコンポーネントを並べ替えるように TargetLink に指示することができます。このソート順は、暗黙的に生成された構造体タイプの構成要素のソートのデフォルトと同じであり、このタイプの変数またはこのタイプを構成要素として組み込んだ変数のメモリフットプリントを小さくすることを目的としています。

#### 関連ドキュメント

- Optimization (Typedef Object)
- 「暗黙的構造体の構造体コンポーネントのソート順の変更」(234 ページ)
- 「暗黙的に生成されたバス構造体の構造体コンポーネントの並べ替え」(235 ページ)

#### スタック/コードサイズのトレードオフのための関数呼び出しの重複

相互に排他的な制御フロー分岐にコードを効果的に移動させることで、コードの複数のインスタンスを代償にして、コストの高い中間変数のコピーを避けることができます。

#### 関連ドキュメント

- 「UtilizeCodeDuplicationInControlFlowBranches」(『TargetLink モデルエレメントリファレンス』[📖](#))
- 「条件付きで実行される制御フローにおけるコードの重複」(227 ページ)

## MIL (Model-in-the-Loop) シミュレーション

### シミュレーション結果の分析

#### プロットズームの改善

TargetLink 2023-B では、More Infos Plot Window のプロットズームが改善されました。信号プロットの X 軸のスケーリングと、Deviation to other signals in Detailed Plot は常に同期しています。

#### 関連ドキュメント

- 「量子化誤差を調査する処理の例」(『TargetLink 準備およびシミュレーションガイド』[📖](#))

## ターゲットシミュレーション (PIL)

### TargetLink 2023-B でのターゲットシミュレーションモジュールの変更

新規および廃止されたコンパイラバージョン 次の表は、TargetLink 2023-B でサポートされるコンパイラバージョンの変更を示しています。

マイクロコントローラファミリ	コンパイラ	新しいコンパイラバージョン	廃止されたコンパイラバージョン
MPC57xxVLE	GreenHill	2023	2022
RH850	GreenHill	2023	2022
TriCore4xx	SMARTCODE	10.1	—

TargetLink でサポートされる評価ボードとコンパイラの完全なリストについては、「[評価ボード、マイクロコントローラ、コンパイラの組み合わせ](#)」(『Evaluation Board Reference』[📖](#))を参照してください。

#### 注記

有効なソフトウェア保守サービス (SMS) 契約に含まれる PIL サポート対象の組み合わせについては、[TargetLink 製品サポートセンター](#)にある dSPACE の [TargetLink PIL Support Web サイト](#)を参照してください。

## ユーザビリティ

### ユーザビリティに関する改良点

#### DD 三者間マージの改善

TargetLink 2023-B では、競合する DD オブジェクトのみを表示する新しいフィルタなど、DD3 ウェイマージに関して多くの点で使い勝手が向上しています。

#### 関連ドキュメント

- 「[DD 三者間マージを実行する方法](#)」(『TargetLink Data Dictionary 基本コンセプトガイド』[📖](#))
- 「[DD 三者間マージ](#)」(『TargetLink Data Dictionary Manager リファレンス』[📖](#))



## テキストベース DD ファイルのサポート

TargetLink 2023-B では、DD ファイルを .dd または .ddjson ファイルとして保存できます。テキストベースの .ddjson フォーマットでは、Git のようなバージョン管理システムで DD ファイルを使用して、次のようなことができます。

- DD プロジェクトファイルの変更を表示します。
- DD プロジェクトのファイル内容のレビューを実行します。
- blame 関数を使用します。

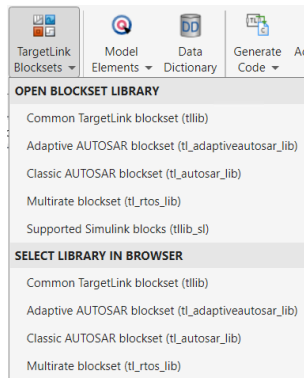
### 関連ドキュメント

- 「DD ファイル形式の基礎」(『TargetLink Data Dictionary 基本コンセプトガイド』[📖](#))
- 「Data Dictionary Command Line Tool の API コマンド」(『TargetLink Data Dictionary Manager リファレンス』[📖](#))

## Simulink ツールストリップの [TargetLink] タブの改善

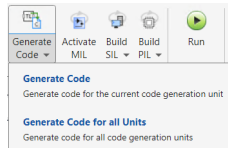
TargetLink 2023-B では、Simulink<sup>®</sup> ツールストリップで、次のような [TargetLink] タブの使い勝手が向上しています。

### Library セクション



拡張 Library セクションには、TargetLink ライブラリへのショートカットを含む新しい TargetLink Blocksets コマンドがあり、ライブラリを開くか Simulink<sup>®</sup> Library Browser に表示するかを選択することができます。

### Generate Code

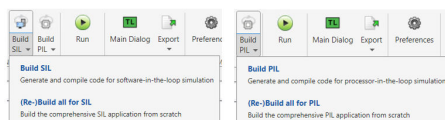


PRODUCTION CODE セクションの [Generate Code] コマンドに以下のオプションが追加されました。

- アイコンをクリックするか、ポップアップリストの [Generate Code] をクリックすると、現在のコード生成ユニット (CGU) のみに対してコード生成が開始されます。
- ポップアップリストで [Generate Code for all Units] をクリックすると、現在のサブシステム、またはサブアイテムを含む対応するコード生成ユニット (CGU) のコード生成が開始されます。

「コード生成の開始の概要」(『TargetLink 準備およびシミュレーションガイド』[📖](#))を参照してください。

## Build SIL/Build PIL



PRODUCTION CODE セクションでは、Build SIL と Build PIL コマンドも拡張されました。

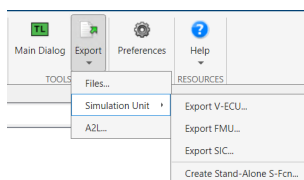
- アイコンをクリックするか、ポップアップリストの Build SIL/BuildPIL をクリックすると、現在の TargetLink サブシステムのそれぞれのビルドプロセスが開始されます。最初のステップとして、現在のサブシステムまたはサブアイテムを含む対応するコード生成ユニット (CGU) のコードが生成されます。

「選択したサブシステムに対してビルドプロセスを開始する方法」(『TargetLink 準備およびシミュレーションガイド』[📖](#))を参照してください。

- ポップアップリストで(Re-)Build all for SIL/(Re-)Build all for PIL をクリックすると、モデル内のすべての TargetLink サブシステムのビルドプロセスがワンステップで開始されます。

「すべてのサブシステムに対してビルドプロセスを開始する方法」(『TargetLink 準備およびシミュレーションガイド』[📖](#))を参照してください。

## Tools セクション



新しい Tools セクションには、以下の 2 つのコマンドが含まれています。

- [Main Dialog]をクリックすると、TargetLink Main Dialog が開きます。
- [Export]をクリックしてポップアップリストのアイテムを選択すると、対応する TargetLink ユーティリティが開くか、または A2L の場合は、ファイル生成が開始されます。次のコマンドを使用することができます。

コマンド	説明	ドキュメント
Files...	TargetLink File Export Utility を開きます。	「TargetLink File Export Utility」(『TargetLink ツールおよびユーティリティリファレンス』 <a href="#">📖</a> )
Simulation Unit - Export V-ECU...	TargetLink V-ECU Manager を開きます。	「TargetLink V-ECU Manager」(『TargetLink ツールおよびユーティリティリファレンス』 <a href="#">📖</a> )
Simulation Unit - Export FMU...	TargetLink FMU Manager を開きます。	「TargetLink FMU Manager」(『TargetLink ツールおよびユーティリティリファレンス』 <a href="#">📖</a> )
Simulation Unit - Export SIC...	TargetLink SIC Manager を開きます。	「TargetLink SIC Manager」(『TargetLink ツールおよびユーティリティリファレンス』 <a href="#">📖</a> )
Simulation Stand-Alone S-Fcn...	TargetLink Stand-Alone Model Manager を開きます。	「Stand-Alone Model Manager」(『TargetLink ツールおよびユーティリティリファレンス』 <a href="#">📖</a> )

コマンド	説明	ドキュメント
A2L	ASAM MCD-2 MC ファイル生成を開始します。	「dsdd_export_a2l_file」(『TargetLink API リファレンス』 <a href="#">📖</a> )

### Autoscaling View Set の改良

TargetLink 2023-B では、Property Manager の Autoscaling View Set に、Min と Max の列が追加表示されます。

#### 関連ドキュメント

- 「Property View を使用事例に合わせて最適化する方法」(『TargetLink 準備およびシミュレーションガイド』[📖](#))
- 「View Set Chooser」(『TargetLink ツールおよびユーティリティリファレンス』[📖](#))

### 関連トピック

#### 参考文献

[TargetLink Main Dialog ブロック](#) (『TargetLink モデルエレメントリファレンス』[📖](#))

## コードジェネレータオプション

### 新しいコードジェネレータオプション

TargetLink 2023-B では、次の新しいコードジェネレータオプションを使用することができます。

- `DecomposeStructAssignmentsForInitialValuePropagation`  
右側のコンポーネントへのアクセスをそれぞれの初期値に置き換えることができれば、構造体の代入をコンポーネント単位の代入に置き換えることができます。
- `NamespaceQualification`  
生成される C++コード内のシンボルを、完全修飾名前空間で修飾するか、周囲の名前空間と異なる部分のみで修飾するかを指定します。
- `UtilizeCodeDuplicationInControlFlowBranches`  
相互に排他的な制御フロー分岐にコードを効果的に移動させることで、コードの複数のインスタンスを代償にして、コストの高い中間変数のコピーを避けることができます。
- `EmitWarningIfMathHeaderIsIncluded`  
C 標準ライブラリヘッダー `math.h` が含まれている場合に警告を発するかどうかを指定します。
- `UseTLImplementationForFabsFunction`  
C 標準ライブラリヘッダー `math.h` の `fabs` 関数を、`?`演算子を使用した条件式として量産コードに直接実装するかどうかを指定します。

### 関連ドキュメント

- 「名前空間の使用の基礎」(『TargetLink カスタマイズおよび最適化ガイド』[📖](#))

### コードジェネレータオプションの移行上の注意点

詳細については、「コードジェネレータオプションに関する移行上の注意点」(209 ページ)を参照してください。

### 関連トピック

#### 参考文献

DecomposeStructAssignmentsForInitialValuePropagation (『TargetLink モデルエレメントリファレンス』[📖](#))  
 EmitWarningIfMathHeadersIncluded (『TargetLink モデルエレメントリファレンス』[📖](#))  
 NamespaceQualification (『TargetLink モデルエレメントリファレンス』[📖](#))  
 UseTLImplementationForFabsFunction (『TargetLink モデルエレメントリファレンス』[📖](#))  
 UtilizeCodeDuplicationInControlFlowBranches (『TargetLink モデルエレメントリファレンス』[📖](#))

## API 関数とフックスクリプト

### 次のステップ

#### 本章の内容

新しい API 関数.....	196
新しいフックスクリプト.....	197

## 新しい API 関数

API 関数	目的
tlExportCLibrary	TargetLink C ライブラリをパッケージコンテナまたは ZIP ファイルとしてエクスポートします。
tlRebuildLinearAlgebraLibrary	TargetLink Linear Algebra Library を再ビルドします。

## 関連トピック

## 参考文献

[tlExportCLibrary](#) (『TargetLink API リファレンス』[📖](#))  
[tlRebuildLinearAlgebraLibrary](#) (『TargetLink API リファレンス』[📖](#))

## 新しいフックスクリプト

フックスクリプト	説明
tl_post_add_adaptivearfunctions subsystem_hook	Adaptive AUTOSAR Function サブシステムがフレームモデルに追加された後に実行する独自のコマンドを入力することができます。
tl_post_add_methodbehaviorsubsystem_hook	Method Behavior サブシステムがフレームモデルに追加された後に実行する独自のコマンドを入力することができます。
tl_post_add_methodcalls subsystem_hook	Method Call サブシステムがフレームモデルに追加された後に実行する独自のコマンドを入力することができます。

## 関連トピック

## 参考文献

[tl\\_post\\_add\\_adaptivearfunctions subsystem\\_hook](#) (『TargetLink ファイルリファレンス』[📖](#))  
[tl\\_post\\_add\\_methodbehaviorsubsystem\\_hook](#) (『TargetLink ファイルリファレンス』[📖](#))  
[tl\\_post\\_add\\_methodcalls subsystem\\_hook](#) (『TargetLink ファイルリファレンス』[📖](#))

## その他

## 次のステップ

## 本章の内容

[FMI 3.0 のサポート](#)..... 198  
[TargetLink のデモと例](#)..... 198

## FMI 3.0 のサポート

### TargetLink が FMI 3.0 をサポート

TargetLink 2023-B は、FMI 3.0 規格の以下の機能をサポートしています。

- Co-Simulation インターフェース
- 整数、ブーリアン、浮動小数点 (32 ビットおよび 64 ビット)、および列挙データタイプ
- 多次元変数 (ベクトルや行列など)

#### 関連ドキュメント

- 「TargetLink からの FMU のエクスポートの基礎」(『TargetLink 相互運用およびデータ交換ガイド』[📖](#))

## TargetLink のデモと例


### 新しいデモモデル

新しいデモモデル	説明
TABLEND	tableNd デモモデルでは、最大 5 次元までのルックアップテーブルのテーブル値を指定するための利用可能な方法を示しています。 「ルックアップテーブルの概要」(『TargetLink 準備およびシミュレーションガイド』 <a href="#">📖</a> )を参照してください
SOLE	このデモでは、連立一次方程式を Linear Algebra Library を使用して SOLE Solver ブロックで解く方法を示します。 「連立一次方程式の解法」(『TargetLink 準備およびシミュレーションガイド』 <a href="#">📖</a> )を参照してください。

### 変更されたデモモデル




変更されたデモモデル	説明
AAR_COMMUNICATION	RP_ServiceB FieldB InPort ブロックでは、C++列挙クラスのデータタイプが使用されます。さらに、ソフトウェアコンポーネントは、GeneralNamespaceRef プロパティを通じて、名前空間 dspace/targetlink に割り当てられます。つまり、その名前空間は、ソフトウェアコンポーネント内のそれ自体では別の名前空間に指定していないすべての関数、変数、データタイプに継承されます。

## 削除されたデモモデル

削除されたデモモデル	説明
MATRIX_INVERSE	<p>原理的には、連立一次方程式(SOLE)は逆行列によって解くことができます。数学的にはエレガントな方法ですが、これは数値的な観点からは実行可能ではありません。</p> <ul style="list-style-type: none"> <li>浮動小数点演算を使用する場合、行列の逆変換が不可能であったり、丸め誤差やオーバーフロー／アンダーフローを伴う可能性があります。</li> <li>逆行列を計算するためには、右辺が単位行列である SOLE を解く必要があります(最初の点で示した理由により、解がない場合もあります)。</li> <li>SOLE を解く(逆行列を計算してから解を求める)ために、分解を使用する代わりに逆行列を使用すると、一般的に計算回数が多くなります。逆行列をまず計算し、それを元の SOLE の右辺と掛け合わせることは、元の SOLE を直接解くよりも常に行列の掛け算が 1 回多くなります。</li> <li>SOLE を直接解くために、SOLE の左行列は特別な性質を持つ行列に分解され、それによって SOLE が簡単に解けるようになります。</li> </ul> <p>精度を高めるために、TargetLink は分解によって SOLE を解きます。</p> <p>その結果、MATRIX_INVERSE デモは、TargetLink から削除されました。このデモモデルは、SOLE デモによって置き換えられています。</p> <p><a href="#">「連立一次方程式の解法」</a>(『TargetLink 準備およびシミュレーションガイド』)を参照してください。</p>

## 関連トピック

## 基本概要

[AAR\\_COMMUNICATION](#) (『TargetLink デモモデル』)  
[SOLE](#) (『TargetLink デモモデル』)  
[TABLEND](#) (『TargetLink デモモデル』)

## 参考文献

[SOLE Solver ブロックの機能](#) (『TargetLink モデルエレメントリファレンス』)

# TargetLink 2023-B および TargetLink Data Dictionary 2023-B への移行

## 次のステップ

## 本章の内容

移行に関する一般的な情報.....	200
TargetLink 2022-B から 2023-B への移行.....	209
TargetLink 2022-B と TargetLink 2023-B 間でのコードの変更.....	218
TargetLink 2023-B での廃止項目.....	247

## 移行に関する一般的な情報

## 次のステップ

## 本章の内容

モデル、ライブラリ、Data Dictionary のアップグレード.....	200
正確に表現可能な浮動小数点数の丸め動作の変更 (Microsoft).....	208

## モデル、ライブラリ、Data Dictionary のアップグレード

## 次のステップ

## 本章の内容

TargetLink バージョン間の移行に関する基礎.....	201
インクルード DD ファイルのある Data Dictionary をアップグレードおよび保存する方法.....	204
API を使用してライブラリとモデルを手動でアップグレードする方法.....	204
Data Dictionary の CodeDecorationSets への移行.....	205



## TargetLink バージョン間の移行に関する基礎

### TargetLink 3.1 以降からの自動アップグレード

TargetLink 2023-B では、TargetLink 3.1 以降で作成された場合、モデルと TargetLink 準拠のライブラリが自動的にアップグレードされます。

自動アップグレードは、間にある各 TargetLink バージョンに必要なすべてのステップを含んでいます。内部的には、モデルと Data Dictionary は、バージョンごとに段階的に自動的にアップグレードされます。

#### 注記

インタラクティブな操作が必要な場合は、TargetLink 各バージョンの TargetLink 移行ドキュメントを確認してください。

**必要なインタラクティブな操作** 以下のような場合は、自動アップグレードに追加のインタラクティブな操作が必要です。

- ライブラリが TargetLink 準拠である必要があります。それ以外の場合、アップグレードを実行することはできません。
- コード生成用のスタイルシートはバージョン固有のため、TargetLink のバージョンによって異なる場合があります。そのため、以前のバージョンの TargetLink の変更されたスタイルシートは、現在のバージョンに合わせてアップデートする必要があります(目的どおりに変更を再適用させます)。「[コード書式の基礎](#)」(『[TargetLink カスタマイズおよび最適化ガイド](#)』[📖](#))を参照してください。
- 32 ビットバージョンの TargetLink でビルドしたカスタムコード S-function は、64 ビットバージョンの TargetLink では使用することができません。  
`tlUpgrade('Model', <MyModel>, 'CheckModel', 'FixIssues')` API 関数を使用して、カスタムコード S-function すべての再ビルドを行います。
- 古いモデルでは、Simulink 列挙データタイプ用のあらかじめ設定された DD EnumImplementation オブジェクトを使用して、量産コードで Simulink 列挙定数の整数値を生成します。このデータタイプをあらかじめ設定された IMPLICIT\_ENUM データタイプと置き換えて、Simulink 列挙データタイプに従って暗黙的な TargetLink 列挙データタイプを生成します。「[TargetLink での列挙データ型の使用の基礎](#)」(『[TargetLink 準備およびシミュレーションガイド](#)』[📖](#))を参照してください。



### Data Dictionary のアップグレード

以前のバージョンの TargetLink で最後に保存された DD ファイルを開く場合は、アップグレードする必要があります。アップグレードは自動的に実行されます。Subsystems エリアと <Application> エリアは、互換性がなくなったため削除されました。アップグレードの動作を変更するには、「[tl\\_pref](#)」(『[TargetLink API リファレンス](#)』[📖](#))を参照してください。

アップグレードした DD ファイルを保存する場合、元の DD ファイルはアップグレードしたファイルで上書きされる前に、その DD ファイルと同じディレクトリにあるバックアップ DD ファイルに自動的に保存されます。自動バックアップを無効にするには、「[Topic Navigator](#)」(『[TargetLink ツールおよびユーティリティリファレンス](#)』[📖](#))の「[Data Dictionary Upgrade](#)」を参照してください。

インクルード DD ファイルを含む DD ファイルをアップグレードするには、「[インクルード DD ファイルのある Data Dictionary をアップグレードおよび保存する方法](#)」(204 ページ)を参照してください。

### あらかじめ設定された DD システムテンプレートの変更


新しい TargetLink バージョンでは、あらかじめ設定された DD システムテンプレートの内容が変更されている可能性があります。これらの変更をビジュアル表示するには、DD プロジェクトファイルを現在の TargetLink バージョンの対応する DD テンプレートと比較します。必要に応じて、新規または変更された DD オブジェクトを DD プロジェクトファイルにマージすることができます。「[New - Create New DD Workspace](#)」(『[TargetLink Data Dictionary Manager リファレンス](#)』)および「[DD 二者間比較を使用した DD オブジェクトとワークスペースの比較の基礎](#)」(『[TargetLink Data Dictionary 基本コンセプトガイド](#)』)を参照してください。

### 新規ライブラリを TargetLink 準拠にする

新規に作成され TargetLink ブロックで構成されるライブラリは、将来新しい TargetLink バージョンにアップグレードできるように上位互換性を持たせる必要があります。それ以外の場合、アップグレードを実行することはできません。

#### 注記

ライブラリは、TargetLink ブロックが含まれる場合、TargetLink ライブラリには自動的になりません。ライブラリ自体が TargetLink 準拠である必要があります。

「[ユーザーライブラリをアップグレード可能にする方法](#)」(『[TargetLink オリエンテーションおよび概要ガイド](#)』)を参照してください。

### 既存のライブラリを TargetLink 準拠にする

次の 2 つのアプローチで、以前のバージョンの TargetLink で作成したライブラリを TargetLink 2023-B に準拠させることができます。

**以前の TargetLink バージョンを使用可能** ライブラリを作成した TargetLink バージョンを使用して、ライブラリを TargetLink 準拠にします。以前の TargetLink バージョンの TargetLink マニュアルを参照してください。TargetLink は自動的にアップグレードを実行するため、このライブラリはそれ以降のすべての TargetLink バージョンで使用することができます。自動アップグレードによってライブラリが新しい TargetLink バージョンで保存されるわけではないので、ライブラリは TargetLink 2023-B より前の TargetLink バージョンでそのまま使用することができます。

**最新の TargetLink バージョン 2023-B のみ使用可能** ライブラリを TargetLink 準拠にするには、TargetLink 2023-B と t1Upgrade API コマンドを使用してください。「[API を使用してライブラリとモデルを手動でアップグレードする方法](#)」(204 ページ)を参照してください。この手順に従うと、ライブラリは TargetLink 2023-B で保存されます。そのため、TargetLink 2023-B より前の TargetLink バージョンでは使用できなくなります。

### TargetLink 2.x または 3.0.x からの手作業によるアップグレード

TargetLink バージョン 2.x または 3.0.x で作成したモデルおよびライブラリは、お持ちの最新 TargetLink バージョン 3.x(3.1~3.5)に手作業でアップグレードする必要があります。その後は自動アップグレードが可能です。

### 下位互換性なし

以前の TargetLink バージョンでは、新しい TargetLink バージョンのフォーマットを使用したモデル、ライブラリ、Data Dictionary を使用することはできません。

## データモデルフィルタールールファイル

TargetLink Data Dictionary のデータモデルが変更されているため、既存のデータモデルフィルタールールファイルには無効な元素が含まれている場合があります。以前の TargetLink バージョンに付属していた以下のファイルは影響を受ける可能性があります。

- DD\_Filter\_Admin.xml
- DD\_Filter\_AR\_User.xml
- DD\_Filter\_NonAR\_NonRTOS\_User.xml

MATLAB コマンドウインドウの API を使用してフィルタ規則ファイルを確認することができます。

単一ファイルの確認	フィルタ規則セット <sup>1)</sup> のチェック
<code>dsdd_free;</code>	<code>dsdd_free;</code>
<code>dsdd('ReadFilterRuleSet', 'file', '&lt;myFile&gt;.xml');</code>	<code>dsdd('ReloadFilterRuleSets');</code>
<code>ds_error_register(dsdd('GetMessageList'));</code>	<code>ds_error_register(dsdd('GetMessageList'));</code>
<code>ds_msgdlg('update');</code>	<code>ds_msgdlg('update');</code>

<sup>1)</sup> Preferences Editor の Data Dictionary - Filter Rules で定義したディレクトリにすべてのファイルが含まれています。

TargetLink Message Browser にエラーに関する情報を TargetLink に表示します。各エラーには以下の情報が含まれているため、任意の XML 対応エディタを使用してエラーを修正することができます。

- ファイル名
- 行番号
- 列番号

## 関連トピック

### 基本概要

[コード書式の基礎](#) (『TargetLink カスタマイズおよび最適化ガイド』[📖](#))  
[データモデルのフィルタ規則セットの基礎](#) (『TargetLink Data Dictionary 基本コンセプトガイド』[📖](#))

### 操作手順

[API を使用してライブラリとモデルを手動でアップグレードする方法](#)..... 204  
[ユーザライブラリをアップグレード可能にする方法](#) (『TargetLink オリエンテーションおよび概要ガイド』[📖](#))

### 参考文献

[tlUpgrade](#) (『TargetLink API リファレンス』[📖](#))

## インクルード DD ファイルのある Data Dictionary をアップグレードおよび保存する方法

**前提条件** ロードするメイン DD ファイルで、DD DDIncludeFiles オブジェクトの AutoLoad プロパティに on が設定されている必要があります。

**方法** **インクルード DD ファイルのある Data Dictionary をアップグレードおよび保存するには**

- 1 [File] - [Open]を使用して、Data Dictionary Manager とメイン DD ファイルを開きます。  
メイン DD ファイルおよびすべてのインクルード DD ファイルが、最新のリリースのデータモデルに自動的にアップグレードされます。
- 2 インクルードされている DD ファイルをメイン DD ファイルと一緒に保存する場合は、必要なすべての /Config/DDIncludeFiles/<DDIncludeFile> オブジェクトの AutoSave プロパティを On に設定します。
- 3 [File] - [Save]を使用してメイン DD ファイルを保存します。

**結果** メイン DD ファイルおよびインクルードされているすべての DD ファイルがアップグレードされ保存されます。TargetLink は、アップグレードされたすべての DD ファイルのリリース番号を最新のデータモデルリリースに調整します。オリジナルの DD ファイルのバックアップも、DD ファイルと同じディレクトリに保存されます。

### 関連トピック

#### 基本概要

[DD ファイルの表示と処理の基礎 \(『TargetLink Data Dictionary 基本コンセプトガイド』\)](#)

#### 操作手順

[部分的な Data Dictionary ファイルをインクルードする方法 \(『TargetLink Data Dictionary 基本コンセプトガイド』\)](#)

#### 参考文献

[Point of Inclusion \(『TargetLink Data Dictionary Manager リファレンス』\)](#)

## API を使用してライブラリとモデルを手動でアップグレードする方法

**目的** たとえば、複数のユーザとのツールチェーンでのライブラリとモデルの一元アップグレードの準備を行います。

**前提条件** モデルまたはライブラリファイルが MATLAB 検索パスで使用可能であるものの、開いてはいません。

必要な DD プロジェクトファイルは、  
`dsdd_manage_project('Open', '<name>.dd')` や `dsdd('Open', '<name>.dd')` などを使用して開いて自動的にアップグレードされています。

**方法** **API を使用してライブラリとモデルを手動でアップグレードするには**

- 1 MATLAB コマンドウィンドウに次の API コマンドを入力します。  

```
tlUpgrade('Model', '<Model|Library>.slx',  
'CheckModel', 'FixIssues')
```

モデルまたはライブラリがアップグレードされます。

**注記**

モデルおよびライブラリをアップグレードする場合には、他のライブラリを参照しない(ブロックおよびサブシステムに他のライブラリへのリンクがない)モデルまたはライブラリを最初にアップグレードします。一番下のライブラリから開始して、順次上のライブラリをアップグレードします。

- 2 アップグレードしたモデルまたはライブラリファイル(Library.slx など)を保存します。
- 3 他のすべてのモデルまたはライブラリに対して、手順 1 および 2 を繰り返します。

**結果** モデルとライブラリがアップグレードされます。

**関連トピック** 参考文献

[tlUpgrade \(『TargetLink API リファレンス』\)](#)

## Data Dictionary の CodeDecorationSets への移行

**CodeDecorationSet と CodeDecoration objects の導入**

TargetLink 4.3 で DD CodeDecorationSet と CodeDecoration オブジェクトが導入されました。

また、Data Dictionary のデータモデルから複数のプロパティが削除されました。

DD オブジェクト	変更	代替
FunctionClass	DeclarationStatements および SectionName プロパティの削除。	DD CodeDecoration.Settings オブジェクトの DeclarationStatements および SectionName プロ パティ。
VariableClass		
VariableClassTemplate.Filter	WidthSpec プロパティの削除。	DD CodeDecoration.Filter オブジェクトの WidthSpec プロパティ。

### TargetLink による自動アップグレード

**制限事項** TargetLink は変数クラスの幅固有タイプのプレフィックスをサポートしなくなりました。オリジナルの Data Dictionary に、幅固有タイプのプレフィックスを持つ変数クラスを派生するために使用する変数クラステンプレートが含まれる場合、Data Dictionary の自動アップグレードは失敗する場合があります。代わりに宣言ステートメントを使用します。

データモデルが最新リビジョンより古い Data Dictionary を開くと、自動アップグレードを実行するように指示されます。

オブジェクトの種類	トリガ	アップグレードのアクション
VariableClass FunctionClass	DeclarationStatements または SectionName プロパティが設定される。	<ol style="list-style-type: none"> <li>DD CodeDecorationSet オブジェクトを作成します。</li> <li>各 DD CodeDecorationSet オブジェクトに対して、単一の DD CodeDecoration オブジェクトを作成します。CodeDecoration オブジェクトとその子オブジェクトの設定は、オリジナルのオブジェクトの設定と一致します。</li> <li>オリジナルのオブジェクトで CodeDecorationSet オブジェクトを参照します。</li> </ol>
SubStructTemplate	Filter.VariableClass が設定される。	<p>以下のプロパティの値を変数クラスから SubStructTemplate オブジェクトのフィルタに転送します。</p> <ul style="list-style-type: none"> <li>DeclarationStatements</li> <li>SectionName</li> <li>TypePrefix</li> </ul>
VariableClassTemplate	<ul style="list-style-type: none"> <li>Filter.FilterCondition が ALL_TRUE に設定される。</li> <li>Settings.VariableClass が、DeclarationStatements または SectionName プロパティが設定されている DD VariableClass オブジェクトを参照します。</li> <li>Filter.WidthSpec プロパティが、この DD VariableClassTemplate オブジェクトまたは Filter.VariableClassSpec プロパティの値が同じである別の VariableClassTemplate オブジェクトに設定される。</li> </ul>	<ol style="list-style-type: none"> <li>新しい DD VariableClass オブジェクトを /Pool/VariableClasses/Templates に作成します。</li> <li>新しい DD CodeDecorationSet オブジェクトを /Pool/CodeDecorations/Templates に作成します。</li> <li>Filter.VariableClassSpec プロパティの値が同じ各 DD VariableClassTemplate オブジェクトに対して、DD CodeDecoration オブジェクトを CodeDecorationSet オブジェクトに追加します。</li> <li>必要に応じて、DD CodeDecoration オブジェクトを指定します。</li> <li>ステップ 1 で作成した VariableClass オブジェクトで、DD CodeDecorationSet オブジェクトを参照します。</li> <li>ステップ 1 で VariableClassTemplate.Settings.VariableClass プロパティにより作成した DD VariableClass オブジェクトを参照します。</li> </ol>

### 変数クラステンプレートの特別な考慮

Filter.FilterCondition プロパティが ALWAYS または NEVER に設定されている DD VariableClassTemplate オブジェクトを指定した場合、オブジェクトの Filter.WidthSpec プロパティは、アップグレード中に代替なしで削除されます。

プロパティの値を維持する場合は、DD VariableClassTemplate オブジェクトの Filter.FilterCondition プロパティを ALL\_TRUE に設定してから Data Dictionary をアップグレードします。

**制限事項** Filter.FilterCondition プロパティが ONE\_OR\_MORE または ALL\_FALSE に設定されている DD VariableClassTemplate オブジェクトはアップグレードされません。

## クリーンアップ

自動アップグレードでは、以前の Data Dictionary で指定された機能が保持されません。それを手動でクリーンアップすると、新しい Data Dictionary でオブジェクト数を削減することができます。

**幅固有変数クラスのマージ** 以前の Data Dictionary に幅固有の VariableClassTemplate/VariableClass オブジェクトが含まれていた場合、新しい Data Dictionary にもこれらすべての変数クラスがそのまま含まれます。

幅固有情報は DD CodeDecoration オブジェクトに格納されるようになったため、Data Dictionary 内の VariableClass オブジェクトの数は手動で削減することができます。たとえば、変数クラスを<Name>\_<Width>の形式で使用していた場合、適切なコード装飾セットを参照する単一の変数クラス<Name>に置き換えることができます。

次の 2 とおりの方法があります。

- コード装飾セットをマッピングする。
  1. アップグレード中に<Name>\_<Width>という名前の各変数クラスに対して生成されたすべての DD CodeDecoration オブジェクトを、単一の CodeDecorationSet オブジェクトにコピーします。
  2. そのフィルタを使用して各 CodeDecoration オブジェクトを幅固定にします。
  3. 結果の CodeDecorationSet オブジェクトを<Name>変数クラスで参照します。
- 変数クラステンプレートに作成されたコード装飾セットを使用する。
  1. オリジナルの<Name>\_<Width>変数クラスが変数クラステンプレートで参照された場合、DD アップグレードで、/Pool/CodeDecorationSets/Templates に幅固有のコード装飾セットが自動的に作成されます。
  2. この装飾セットは、<Name>という結果の変数クラスで参照することができます。

### 注記

モデルエレメントから<Name>\_<Width>という変数クラスへのリファレンスを<Name>へのリファレンスに置き換えてください。

**変数クラステンプレートの再ターゲット化** 以前の幅固有変数クラスをマージすると、もう一度変数クラステンプレートのターゲットとして使用することができます。さらに、アップグレード中に作成された/Pool/VariableClasses/Templates に含まれるすべての変数クラスを削除することができます。



**ユーザ指定のスキープの縮小チェーン(SRC)の単純化** ユーザ指定の SRC を使用して、宣言ステートメントまたは特定のスキープを持つ変数のセクション名を指定した場合、以下を行うことができます。

1. SRC の最初の変数クラス(最上位スキープ)に参照されるセットに属するコード装飾の Filter.ScopeSpec プロパティを、必要に応じて調整することができます。
2. SRC のその他の変数クラスを削除することができます。
3. SRC を使用して静的ローカル変数の回避も行っていた場合は、代わりに AvoidStaticLocalScope コードジェネレータオプションを使用できるようになりました。

**廃止された変数クラステンプレートの削除** Filter.VariableClassSpec プロパティの値が同じ DD VariableClassTemplate オブジェクトを検索し、1 つを除いてすべて削除します。

#### 生成された量産コード内の変更

DD CodeDecoration オブジェクトでの変更は、生成された量産コードに以下の点で影響を及ぼす可能性があります。

- 「コードコメントの変更」(『TargetLink 新機能と移行手順』📖)
- 「変数定義のソート」(『TargetLink 新機能と移行手順』📖)

「TargetLink 4.3 と TargetLink 4.4 間でのコードの変更」(『TargetLink 新機能と移行手順』📖)を参照してください。

## 正確に表現可能な浮動小数点数の丸め動作の変更 (Microsoft)

### 正確に表現可能な浮動小数点数の丸め動作の変更 (Microsoft)

#### 概要

Microsoft は、`printf` 関数ファミリの丸め動作を変更しました。Windows 10 バージョン 2004 (build 19041) から、IEEE 754-2019 に準拠した正確に表現可能な浮動小数点数を出力するようになりました。

#### TargetLink でのコード生成への影響

Microsoft によるこの変更は、TargetLink でのコード生成に影響を与えません。Windows のバージョンによっては、丸めの動作がマシンによって異なる場合があります。なお、これは TargetLink とは無関係です。

Windows 10 < 2004 (build 19041)	Windows 10 ≥ 2004 (build 19041)
5 で終わる正確に表現可能な浮動小数点数は切り上げられます。	IEEE 754-2019 に準拠した最近接丸め。



# TargetLink 2022-B から 2023-B への移行

## 次のステップ

## 本章の内容

コードジェネレータオプション.....	209
API 関数とフックスクリプト.....	211
移行に関するその他の注意点.....	217
MATLAB のサポート.....	218

## コードジェネレータオプション

### コードジェネレータオプションに関する移行上の注意点

#### ConformingFreestandingImplementation

ConformingFreestandingImplementation Code Generator オプションは、以下のオプションに分けられます。

- `EmitWarningIfMathHeaderIsIncluded`  
C 標準ライブラリヘッダー `math.h` が含まれている場合に警告を発するかどうかを指定します。デフォルトは `off` です。
- `UseTLImplementationForFabsFunction`  
C 標準ライブラリヘッダー `math.h` の `fabs` 関数を、`?`演算子を使用した条件式として量産コードに直接実装するかどうかを指定します。デフォルトは `off` です。

さらに、`t1_fabs dsfxp` ファイルは量産コードには含まれません。これは、コードジェネレータがそこで定義されたマクロを呼び出さなくなり、「`?`」演算子を使用した条件式を生成するからです。

#### 関連ドキュメント

- [「新しいコードジェネレータオプション」](#)(195 ページ)
- [「廃止されたコードジェネレータオプション」](#)(249 ページ)
- [「その他のコード変更」](#)(238 ページ)

#### 変更されたデフォルトの基礎

Code Generator オプションの設定は、モデルとともに保存されます (モデルベースのオプションの保存)。さらに、コードジェネレータオプションのユーザ定義セットを DD `CodegenOptionSet` オブジェクトに保存することができます (DD ベースのオプション保存)。DD `CodegenOptionSet` オブジェクトは、上書きの中心的ソースとして使用でき、TargetLink 4.1 以降使用されているモデルベースのオプション設定を置き換えるためにも使用することができます。

モデルベースのオプションの値が以前のデフォルト値と等しい場合は、アップグレードの際に新しいデフォルト値に自動的に変更されます。DD ベースのオプションの値が以前のデフォルト値と等しい場合は、アップグレードの際に新しいデフォルト値に変更されず、以前のデフォルト値が維持されます。

**オプションの値 = 以前のデフォルト値** コードジェネレータオプションが TargetLink の以前のバージョンのデフォルト値に設定され、TargetLink の新しいバージョンが変更されたデフォルト値を使用する場合は、以下の点に注意してください。

■ モデルベースのオプション:

以前のデフォルト値を維持する場合は、それらを手作業でリセットする必要があります。

■ DD ベースのオプション:

新しいデフォルト値を使用する場合は、それらを手作業で調整する必要があります。

3つのオプション値 9、11、13 における TargetLink アップグレード (TargetLink<sub>Old</sub> から TargetLink<sub>New</sub> へ) の影響について、次の表に例を示します。9、11、and 13 表では、2つの基本的な移行シナリオについて説明します。

■ シナリオ#1: 新しいデフォルト値 = 以前のデフォルト値

Code Generator オプションのデフォルト値は新しい TargetLink バージョンで変更されず、デフォルト値は 9 のままとなります。

オプション値はいずれも変更されません。

■ シナリオ#2: 新しいデフォルト値 ≠ 以前のデフォルト値

Code Generator オプションのデフォルト値は新しい TargetLink バージョンで変更され、デフォルト値は 11 に変更されます。

オプションの保存	オプション値 (TargetLink <sub>Old</sub> ) デフォルト = 9	オプション値 (≤ TargetLink <sub>New</sub> )	
		デフォルト = 9 (シナリオ#1)	デフォルト = 11 (シナリオ#2)
モデルベース	9 <sup>1)</sup>	9 <sup>1)</sup>	11 <sup>2)</sup>
	11	11	11 <sup>1)</sup>
	13	13	13
DD ベース	9	9	9 <sup>3)</sup>
	11	11	11
	13	13	13

1) オプション値は、デフォルト値と等しいためこのモデルで保存されません。

2) 手作業によるリセットが必要な場合があります。

3) 手作業による調整が必要な場合があります。

**オプションの値 = 新しいデフォルト値** コードジェネレータオプションが TargetLink の以前のバージョンのデフォルト値に設定されていない (A) が、TargetLink の新しいバージョンのデフォルト値に設定されている場合 (B)、TargetLink では、新しい TargetLink バージョンで意図的にデフォルト値が指定されたと思なされます。TargetLink の次のバージョンでデフォルト値が再び変更された場合も同様です (C)。

**注記**

TargetLink<sub>A</sub> ⇒ TargetLink<sub>B</sub> ⇒ TargetLink<sub>C</sub> のアップグレードと TargetLink<sub>A</sub> ⇒ TargetLink<sub>C</sub> のアップグレードにより、異なるオプション値になる可能性があります。下の表を参照してください。

TargetLink のバージョン A、B、C のデフォルト値が 9、11、13 であり、オプション値がバージョン A で 11 に設定された場合は、バージョン C へのアップグレードにより次のようにオプション値が変更されます。

アップグレード方法	オプション値 TargetLink <sub>A</sub> デフォルト = 9	オプション値 TargetLink <sub>B</sub> デフォルト = 11	オプション値 TargetLink <sub>C</sub> デフォルト = 13
A ⇒ B ⇒ C	11 (≠ デフォルト)	11 (= デフォルト) <sup>1)</sup>	13 (= デフォルト) <sup>1)</sup>
A ⇒ C	11 (≠ デフォルト)	—	11 (≠ デフォルト)

<sup>1)</sup> オプション値は、デフォルト値と等しいためこのモデルで保存されません。

**新しいコードジェネレータオプション**

新しいコードジェネレータオプションの詳細については、「[新しいコードジェネレータオプション](#)」(『TargetLink 新機能と移行手順』[📖](#))を参照してください。

**関連トピック****参考文献**

EmitWarningIfMathHeaderIsIncluded (『TargetLink モデルエレメントリファレンス』[📖](#))  
 UseTLImplementationForFabsFunction (『TargetLink モデルエレメントリファレンス』[📖](#))

## API 関数とフックスクリプト

**次のステップ****本章の内容**

API 関数の変更.....	212
フックスクリプトの変更.....	215

## API 関数の変更

### 量産コードのビルドとコンパイル

Data Dictionary から直接、量産コードのビルドとコンパイルができるようになりました。

**新しいプロパティ** t1\_build\_host および t1\_build\_target の場合:

プロパティ	説明
CompilerOptions	コンパイラフラグを追加できます。これは Data Dictionary からビルドする場合にのみ機能します。
SimConfigName	t1_build_target の場合のみ。シミュレーション設定名を指定します。デフォルトは default EVM です。

t1\_compile\_host および t1\_compile\_target の場合:

プロパティ	説明
DDModules	量産コードをコンパイルする DD Module または DD ModuleGroup オブジェクトのリストを指定します。このプロパティを Model と TLSubsystem プロパティなしで使用した場合、コードは Data Dictionary からコンパイルされます。
CompilerOptions	コンパイラフラグを追加できます。これは Data Dictionary からビルドする場合にのみ機能します。
SimConfigName	t1_compile_target の場合のみ。シミュレーション設定名を指定します。デフォルトは default EVM です。

**プロパティの変更** t1\_build\_host および t1\_build\_target の場合:

プロパティ	説明
DDCodeGeneratorUnits	このプロパティを Model と TLSubsystem プロパティなしで使用した場合、コードは Data Dictionary から生成されコンパイルされます。
DDModules	このプロパティを Model と TLSubsystem プロパティなしで使用した場合、コードは Data Dictionary から生成されコンパイルされます。

t1\_compile\_host および t1\_compile\_target の場合:

プロパティ	説明
DDCodeGeneratorUnits	このプロパティを Model と TLSubsystem プロパティなしで使用した場合、量産コードは Data Dictionary からコンパイルされます。
DDModules	このプロパティを Model と TLSubsystem プロパティなしで使用した場合、量産コードは Data Dictionary からコンパイルされます。

**削除されたプロパティ** t1\_compile\_host および t1\_compile\_target の場合:

- RebuildAll

## 関連ドキュメント

- 「tl\_build\_host」(『TargetLink API リファレンス』[📖](#))
- 「tl\_build\_target」(『TargetLink API リファレンス』[📖](#))
- 「tl\_compile\_host」(『TargetLink API リファレンス』[📖](#))
- 「tl\_compile\_target」(『TargetLink API リファレンス』[📖](#))

## dsdd\_export\_a2l\_file

## 新しいプロパティ

プロパティ	説明
ASAP2Version	ASAM MCD-2 MC 規格のバージョンを指定します。選択可能な値は、1.61 および 1.71 です。デフォルトは 1.61 です。

## 関連ドキュメント

- 「dsdd\_export\_a2l\_file」(『TargetLink API リファレンス』[📖](#))
- 「A2L ファイル形式の基礎」(『TargetLink 相互運用およびデータ交換ガイド』[📖](#))

## tl\_clean

## 新しいプロパティ

プロパティ	説明
bForceDelete	指定したコード生成ユニットのファイルを確認を求めることなく削除するかどうかを指定します。デフォルトは false です。

## 関連ドキュメント

- 「tl\_clean」(『TargetLink API リファレンス』[📖](#))

## tl\_generate\_fmu

## 新しいプロパティ

プロパティ	説明
FmiVersion	FMI バージョンを指定します。選択可能な値は、2.0 または 3.0 です。デフォルトは 3.0 です。

## 関連ドキュメント

- 「tl\_generate\_fmu」(『TargetLink API リファレンス』[📖](#))

## tl\_get

AUTOSAR DD エLEMENTのフル DD パスを AUTOSAR ブロックから取得できるようになりました。

## 関連ドキュメント

- 「tl\_get」(『TargetLink API リファレンス』[📖](#))

## tlEnumDataType

## 新しいプロパティ

プロパティ	目的
Overwrite	既存の Simulink 列挙データタイプ指定ファイルを上書きするかマージ(デフォルト)するかを指定します。
SLEnumType	Simulink 列挙データタイプ型の名前を指定します。Simulink 列挙データタイプ名が既に DD EnumTemplate オブジェクトによって指定されている場合、このプロパティは無視されます。
CreateTemplate	Simulink 列挙データタイプと DD Typedef オブジェクトとの間のマッピングを維持する DD EnumTemplate オブジェクトを作成します。DD EnumTemplate オブジェクトが既に存在する場合、このプロパティは無視されます。

## 関連ドキュメント

- 「tlEnumDataType」(『TargetLink API リファレンス』[📖](#))
- 「TargetLink での列挙データ型の使用の基礎」(『TargetLink 準備およびシミュレーションガイド』[📖](#))

## tl\_pref

## 新しいプロパティ

プロパティ	目的
importtolinux	TLPreferences.xml を linux プラットフォームにインポートします。

## 関連ドキュメント

- 「tl\_pref」(『TargetLink API リファレンス』[📖](#))

## tl\_generate\_swc\_model

## 名前が変更されたプロパティ

以前の名前	新しい名前
AddRunnableTriggerPort	AddTriggerPortToRunnableSubsystem
AddOperationSubsystemTriggerPort	AddTriggerPortToOperationSubsystem
BusObjectNamePrefix	SimulinkDataTypePrefix
UpdateBusObjectWithMappedDDTypedef	UpdateSimulinkDataTypeWithDDTypedef

下位互換性を保つため、旧名称もサポートされています。今後の TargetLink のバージョンでは削除される予定です。

## 削除されたプロパティ

- AutosarVersion
- ModelClientServerPorts
- AddOperationCallTriggerPort

## 関連ドキュメント

- 「tl\_generate\_swc\_model」(『TargetLink API リファレンス』[📖](#))
- 「Adaptive AUTOSAR フレームモデルの基礎」(『TargetLink Adaptive AUTOSAR モデリングガイド』[📖](#))

- 「TargetLink での AUTOSAR 開発アプローチとフレームモデル生成の基礎」  
(『TargetLink Classic AUTOSAR モデリングガイド』📖)
- 「TargetLink で使用する Simulink® Bus オブジェクトの基礎」(『TargetLink 準備およびシミュレーションガイド』📖)

**dsdd ('Set', ...)**

dsdd('Set', ...) API の動作は、カスタムプロパティに関して変更されています。既存のプロパティのタイプを明示的に指定しなかった場合、そのタイプは保持され、デフォルトのタイプに変更されることはありません。

**関連ドキュメント**

- Set
- Custom Properties

**CreateModulesForServiceInterfaces**

CreateModulesForServiceInterfaceTypes API 関数の名前が CreateModulesForAdaptiveAutosarInterfaceTypes に変更されました。/Pool/Autosar/Config/ImportExport の適切な DD プロパティで dsdd\_master\_adaptive\_autosar.ddjson[System] テンプレートも名前が変更されました。

**関連ドキュメント**

- CreateModulesForAdaptiveAutosarInterfaceTypes

## フックスクリプトの変更

**tl\_post\_add\_arportblock\_フック**

フックスクリプトが Adaptive AUTOSAR のフレーム生成でも動作するようになりました。そのため、以下のあらかじめ設定された新しい変数が追加されました。

あらかじめ設定された変数	説明
aarSubsystemPortInfo	現在のポートを記述する構造体 Adaptive AUTOSAR にのみ適用されます。

構造体コンポーネントの詳細については、「tl\_post\_add\_arportblock\_hook」(『TargetLink ファイルリファレンス』📖)を参照してください。

**関連ドキュメント**

- 「tl\_post\_add\_arportblock\_hook」(『TargetLink ファイルリファレンス』📖)
- 「Adaptive AUTOSAR フレームモデルの基礎」(『TargetLink Adaptive AUTOSAR モデリングガイド』📖)

**tl\_post\_add\_comspecblock\_フック**

フックスクリプトが Adaptive AUTOSAR のフレーム生成でも動作するようになりました。

定義済み変数 hRunnablePort は hConnectedBlock に名前が変更されました。あらかじめ設定された変数 hRunnablePort は下位互換性のためにまだサポートされていますが、将来の TargetLink リリースでは削除される予定です。

関連ドキュメント [「tl\\_post\\_add\\_comspecblock\\_hook」](#) (『TargetLink ファイルリファレンス』[📖](#))

#### tl\_post\_add\_swcsubsystem\_フック

フックスクリプトが Adaptive AUTOSAR のフレーム生成でも動作するようになりました。

次のあらかじめ設定された新しい変数が追加されました。

あらかじめ設定された変数	説明
aarFunctionSubsystemInfoList	ソフトウェアコンポーネントのすべての Adaptive AUTOSAR Function および Method Behavior サブシステムを記述する aarSubsystemInfo 構造のベクトル。

関連ドキュメント [「tl\\_post\\_add\\_swcsubsystem\\_hook」](#) (『TargetLink ファイルリファレンス』[📖](#))

#### tl\_pre\_add\_arportblock\_hook

フックスクリプトが Adaptive AUTOSAR のフレーム生成でも動作するようになりました。そのため、以下のあらかじめ設定された新しい変数が追加されました。

あらかじめ設定された変数	説明
aarSubsystemPortInfo	現在のポートを記述する構造体 Adaptive AUTOSAR にのみ適用されます。

構造体コンポーネントの詳細については、[「tl\\_pre\\_add\\_arportblock\\_hook」](#) (『TargetLink ファイルリファレンス』[📖](#))を参照してください。

#### 関連ドキュメント

- [「tl\\_pre\\_add\\_arportblock\\_hook」](#) (『TargetLink ファイルリファレンス』[📖](#))
- [「Adaptive AUTOSAR フレームモデルの基礎」](#) (『TargetLink Adaptive AUTOSAR モデリングガイド』[📖](#))

#### tl\_pre\_add\_comspecblock\_フック

フックスクリプトが Adaptive AUTOSAR のフレーム生成でも動作するようになりました。

あらかじめ設定された変数 hRunnablePort は、hConnectedBlock に名前が変更されました。あらかじめ設定された変数 hRunnablePort は、下位互換性のためにまだサポートされていますが、将来の TargetLink リリースでは削除される予定です。

関連ドキュメント [「tl\\_pre\\_add\\_comspecblock\\_hook」](#) (『TargetLink ファイルリファレンス』[📖](#))

#### tl\_pre\_procode\_sim\_hook

フック関数 ( tl\_pre\_procode\_sim\_hook ) の使用時にエラーが発生した場合、TargetLink はエラーメッセージを表示し、シミュレーションをキャンセルします。

#### 関連ドキュメント

- [「tl\\_pre\\_procode\\_sim\\_hook」](#) (『TargetLink ファイルリファレンス』[📖](#))



# 移行に関するその他の注意点

## 移行に関するその他の注意点

### Data Dictionary の Subsystems 領域における DD Function オブジェクトの位置

Subsystems 領域では、すべての DD Function オブジェクトの位置が変わります。TargetLink 2023-B では、これらは Functions という名前の DD オブジェクトの下にカプセル化されています。スクリプトを使用してこれらの DD オブジェクトにアクセスする場合は、Subsystems 領域の新しい位置に従ってスクリプトを修正する必要があります。

#### 関連ドキュメント

- 「Data Dictionary の構造 (DD オブジェクトツリー) の基礎」(『TargetLink Data Dictionary 基本コンセプトガイド』[📖](#))

### Data Dictionary のあらかじめ設定されたシステムテンプレートの新しいファイル形式

Data Dictionary のあらかじめ設定されたシステムテンプレートのファイル形式が .dd から .ddjson に変更されました。あらかじめ設定されたシステムテンプレートに元の名前でアクセスするスクリプトは、修正する必要があります。

#### 関連ドキュメント

- 「DD ファイル形式の基礎」(『TargetLink Data Dictionary 基本コンセプトガイド』[📖](#))
- 「DD ワークスペースを作成する方法」(『TargetLink Data Dictionary 基本コンセプトガイド』[📖](#))

### IF\_\*変数を回避

TargetLink 2023-B では、最適化されていないコードでより多くの IF\_\*変数が生成されます。以下の方法で、この動作を防ぐことができます。

方法	手順	備考
最適化を有効にする	TargetLink Main Dialog の [Advanced] ページで Generate optimized code チェックボックスを選択し、InliningThreshold コードジェネレータオプションを $\geq 1$ に設定します。	この方法により、コードが変更される可能性があります。「アトミックサブシステムおよび機能インターフェースの主な(または最新の変更点)」(218 ページ)を参照してください。
Ensure output is virtual をアクティブにする	Output ブロックの Simulink ダイアログで [Ensure output is virtual] チェックボックスを選択します。	MIL の動作を変更できます。MathWorks®社の Simulink マニュアルを参照してください。

関連するコード変更 「簡易初期化モードで追加される IF\_\* 変数」(219 ページ)

# MATLAB のサポート

## MATLAB R2023a および R2023b はサポートされない

当初、TargetLink 2023-B(23.1)は MATLAB® R2023a および R2023b をサポートしていませんでした。MATLAB R2023a および R2023b のサポートは、TargetLink 23.1 Patch 1 で提供される予定です。ダウンロードについては、<https://www.dspace.com/go/PatchesTL> を参照してください。

## TargetLink 2022-B と TargetLink 2023-B 間でのコードの変更

### 次のステップ

### 本章の内容

アトミックサブシステムおよび機能インターフェースの主な(または最新の)変更点.....	218
コードの効率性.....	223
Adaptive-AUTOSAR 関連と CPP 関連.....	231
構造体関連と構造体配列関連.....	233
Classic AUTOSAR 関連.....	237
その他のコード変更.....	238

## アトミックサブシステムおよび機能インターフェースの主な(または最新の)変更点

### 概要

以前のバージョンの TargetLink には、以下のコンテキストで誤ったコード(MIL/SIL の違い)が表示されるバグがありました。

- 条件付きで実行または反復されるサブシステムの出力ポート、および簡易初期化モードにおけるサブシステムの出力ポートへの先行ブロックからの初期値の継承。

TargetLink 2023-B では、このバグは修正されました。

このバグを修正するために、TargetLink が条件付き実行サブシステムまたは反復サブシステムの出力ポートに IF\_\*変数が必要かどうかを判断する方法を変更する必要がありました。その結果、バグの影響を直接受けないコンテキストであっても、量産コードに IF\_\*変数が追加される可能性があります。

この変更を緩和するため、IF\_\*変数の作成に関する追加変更が導入され、IF\_\*変数や、IF\_\*変数が他の変数と置き換わる構造的に同一のコードをより適切に削除できるようになりました。


#### 注記

以下の条件が満たされていれば、量産コードはほとんど影響を受けることはありません。

- 仕様によって関数のインライン化が防止されないこと。
- DD VariableClass オブジェクトは、モデリングに必要な場合にのみ使用すること。
- モデルが従来の初期化モードを使用するように設定されていること。  
従来の初期化モードのサポートは、今後の TargetLink バージョンで廃止され削除される予定です。「[廃止予定の機能](#)」(250 ページ)を参照してください。

#### 簡易初期化モードで追加される IF\_\* 変数

簡易初期化モードで、条件付き実行サブシステムまたは反復サブシステムの拡張されていない Outport ブロックが先行ブロックから初期値を継承しない場合、TargetLink は、IF\_<Name of outport>という命名スキームを持つ出力ポート用の変数を作成します。以下の表に、以前の TargetLink バージョンと現在の TargetLink バージョンの違いを示します。

TargetLink < 2023-B	TargetLink 2023-B
初期値を継承することが出力ポートで指定されている場合、IF_*変数は作成されません。	初期値が実際に継承される場合、IF_*変数は作成されません。 Simulink では、実際の継承は信号線上の IC アイコンによってビジュアル表示されます。  Simulink マニュアルの <a href="#">Conditional Subsystem Initial Output Values</a> も参照してください。

最初に生成されたコードのアトミックサブシステムから生じる関数がインライン化されていれば、追加された IF\_\*変数は最適化によって排除することができます。サブシステム出力ポートの先行ブロックの出力変数は、通常、異なる初期スコープと初期値なしで作成され、追加された IF\_\*変数は、最適化の順序と最適化アルゴリズムのマッチングに関して、TargetLink 最適化の自由度を大きくします。したがって、TargetLink < 2023-B (23.1)と比較すると、以下に示すような追加コードの変更が発生する可能性があります。

- 追加コメント
- 変数名の変更
- 変数が初期化されない
  - 変数の宣言/定義の順序が異なる
  - 制御フローに移動した変数

- 異なるステートメントの順序
- アクセス関数を持つ変数の場合、最初に生成された関数横断的なアクセスパターンが、関数局所的なアクセスパターンになることもあれば、その逆の場合もあります。従って、生成されるコードはより効率的になる場合もあれば、効率的でなくなる場合もあります。あるいは、関数インターフェースでの通信パラダイムが、アクセス関数経由の通信から IF\_\*変数経由の通信に変わる場合もあります。
- 反復サブシステムの場合、先行ブロックが Assignment ブロックであれば、コードジェネレータは Simulink 出力ポート用の出力変数を生成します。その結果、最適化のできない効率性の悪いコードが生成されます。

#### 理由

- バグ修正
- MIL/SIL/PIL シミュレーションモードの違いを解決

**条件付きで実行されるサブシステムの移行の問題** Simulink がそれを許可し、MIL/モデルのセマンティクスの変更を受け入れることができる場合、条件付きで実行されるサブシステムの出力ポートの Ensure output is virtual チェックボックスを有効にすることができます。

**反復サブシステムの移行の問題** 多くの場合、以下の条件がすべて満たされていれば、MIL/モデルのセマンティクスを変更することなく、反復サブシステムの出力ポートの Ensure output is virtual チェックボックスを有効にすることができます。

1. その周囲では、反復サブシステムは少なくとも 1 回は反復される
2. 反復サブシステムの Simulink 出力ポートの先行ブロックは、常にその出力を完全に書き込む

先行ブロックが Assignment ブロックである場合、さらなるアクションが必要になる場合もあります。

Assignment ブロックのデフォルト設定のコードパターンは、入力 Y0 によって完全なブロック出力 Y を初期化することで、2 番目の条件を満たしています。

もし Assignment ブロックのデフォルトの動作を Omit dispensable initializations チェックボックスを有効にするか、出力変数の変数クラスを MERGEABLE に指定した場合、ブロック出力が完全に書き込まれることは保証されなくなり、以下の条件が適用される必要があります。

- Iterator ブロックは Assignment ブロックのインデックスポートを駆動する必要があります。
- その環境では、反復システムは、Assignment ブロックの出力変数のすべてのエレメントを反復処理する必要があります。

**追加情報** 「[影響と相互作用](#)」(222 ページ)を参照してください。

#### より構造化された IF\_\*変数

TargetLink 2023-B は、より多くの状況で、複数の非構造化 IF\_\*変数の代わりに、1 つの構造化 IF\_\*変数を生成します。

具体的には、IF\_\*変数は、以下の条件が満たされる場合、先行ブロックの出力変数の構造体タイプを取得することが多くなります。

- バス信号は、量産コードで構造化された出力変数を持つ先行ブロックを持つ、拡張されていない Inport または Outport ブロックに接続されます。
- TargetLink は、この Inport または Outport ブロックに対して IF\_\*変数を作成する必要があります。

TargetLink ≤ 2022-B	TargetLink 2023-B
<pre> UInt32 IF_Sa2_Outputport = 0; UInt16 IF_Sa2_Outputport_a = 0; ... Float64 IF_Sa2_Outputport_g[6] = { /* [0..5] */ 0., 0., 0., 0., 0., 0. /* 0., 0., 0., 0., 0., 0. */ }; UInt32 IF_Sa2_Outputport_h = 0;  struct e_new_Component_e_a_t_tag Sa5_Selector1;  IF_Sa2_Outputport = Sa5_Selector1.e_a_a; IF_Sa2_Outputport_a = Sa5_Selector1.e_a_b.e_a_b_a.e_a_b_a_a; ... for (Aux_ = 0; Aux_ &lt; 6; Aux_++) { IF_Sa2_Outputport_g[Aux_] = Sa5_Selector1.e_a_c.e_a_c_a[Aux_]; } IF_Sa2_Outputport_h = Sa5_Selector1.e_a_d; </pre>	<pre> struct e_new_Component_e_a_t_tag IF_Sa2_Outputport = { 0, /* e_a_a */ { 0, /* e_a_b_a_a */ ... } }, { { /* [0..5] */ 0., 0., 0., 0., 0., 0. /* 0., 0., 0., 0., 0., 0. */ } /* e_a_c_a */ }, 0 /* e_a_d */ };  struct e_new_Component_e_a_t_tag Sa5_Selector1;  IF_Sa2_Outputport = Sa5_Selector1; </pre>

**理由**

- コードの効率性
- コードサイズ
- モデルとコードのマッチングが容易
- 整合性の向上
- 可読性

**追加情報** 「[影響と相互作用](#)」(222 ページ)を参照してください。

### 変数クラステンプレート SLFunctionInputStruct および SLFunctionOutputStruct の一 貫性の高い使用

バス信号に接続される拡張されていない Simulink ポートについては、変数クラステンプレート SLFunctionInputStruct と SLFunctionOutputStruct が、これらのポートを量産コードでどのように実装するかを決定します。構造体変数がこのようなポートに接続されている場合、デフォルトでは、TargetLink 2023-B の参照渡しファンクションパラメータとして実装されます。

TargetLink ≤ 2022-B では、先行ブロックが構造体タイプを使用して参照渡しとして実装されている場合にのみ、このようなポートをファンクションパラメータとして実装します。そうでない場合、TargetLink は該当するポートに対して、構造化されていないグローバル変数 IF\_\* を生成していました。

実際のパラメータの名前テンプレートは、IF\_ では始まりません。TargetLink ≤ 2022-B と TargetLink 2023-B の両方が構造化された IF\_\* 変数を持ち、アトミックサブシステムから生じる関数がインライン化されている場合には、IF\_\* 変数は外見적으로는 IF\_ 接頭辞を失います。

**理由**

- バグ修正
- コードの効率性
- 整合性の向上
- MISRA C 準拠
- 精度
- 可読性
- コンパイル依存の低減

**追加情報** 「影響と相互作用」(222 ページ)を参照してください。

**影響と相互作用**

言及されたコード変更は、より複雑なコード変更につながる形で相互作用する可能性があります。

**関数の再利用と複数のインスタンス化可能な AUTOSAR ソフトウェアコンポーネント**

- グローバルインターフェース変数から参照呼び出し通信への変更によって、実際のパラメータに初期値が必要でない限り、それぞれの変数を再利用または自動作成された PIM 構造から効果的に削除することができます。

[「変数クラステンプレート SLFunctionInputStruct および SLFunctionOutputStruct の一貫性の高い使用」](#)(221 ページ)を参照してください。

- 個々の数値変数から構造化インターフェース変数への変更は、構造体の割り当てに拘束されている構造体タイプの不要なコンポーネントが削除されなかったり、他の変数で置き換えられなかったりするため、再利用構造体にコンポーネントが追加される可能性があります。

[「より構造化された IF\\_\\* 変数」](#)(220 ページ)を参照してください。

- 先行変数のスコープと初期値が失われることで、再利用や自動作成される PIM 構造が変化します。

[「簡易初期化モードで追加される IF\\_\\* 変数」](#)(219 ページ)を参照してください。

- 構造体コンポーネントの並べ替えの変更に伴い、これらの暗黙的に生成される構造体タイプのレイアウトが以前の TargetLink バージョンと大きく異なる場合があります。

[「暗黙的構造体の構造体コンポーネントのソート順の変更」](#)(234 ページ)および暗黙的に生成されたバス構造体の構造体コンポーネントの並べ替えを参照してください。

**関数境界でのコードの効率性**

- [簡易初期化モードで追加される IF\\_\\* 変数](#)は、特に関数境界がインライン化によって解消されていない場合、しばしばコードの効率性に悪影響を及ぼします。

[「簡易初期化モードで追加される IF\\_\\* 変数」](#)(219 ページ)を参照してください。

- [より構造化された IF\\_\\* 変数](#)は通常、インライン化によって解消されない関数境界のコードの効率性を向上させ、先行変数が除去できる限り、不都合な変更を補うことができます。

[「より構造化された IF\\_\\* 変数」](#)(220 ページ)を参照してください。

- [変数クラステンプレート SLFunctionInputStruct および SLFunctionOutputStruct の一貫性の高い使用](#)は通常、スタック消費量の増加と引き換えに、インライン化によって解消されない関数境界の RAM 消費量を減少させ、常に書き込まれるわけではない先行状態(条件付きで実行されるサブシステムの出力など)の除去や包括を妨げる可能性があります。

「変数クラステンプレート SLFunctionInputStruct および SLFunctionOutputStruct の一貫性の高い使用」(221 ページ)。

- Stateflow 入力の構造体変数も参照してください。

**Includes** より構造化された IF\_\*変数とは、t1\_basetypes および udt ヘッダーがヘッダーとソースファイルの間で切り替わる原因になる場合があります。

「より構造化された IF\_\*変数」(220 ページ)および「変数クラステンプレート SLFunctionInputStruct および SLFunctionOutputStruct の一貫性の高い使用」(221 ページ)を参照してください。

## 関連トピック

### 参考文献

Assignment ブロック (『TargetLink モデルエレメントリファレンス』)

## コードの効率性

カスタムコードにおける構造体配列の中間変数の除去を改善

カスタムコードのコンテキストで、カスタムコードの入力が定義された範囲のサブセットにアクセスする場合、TargetLink は構造体配列の中間変数を削除できるようになりました。

TargetLink ≤ 2022-B	TargetLink 2023-B
<pre>for (Aux_S32 = 0; Aux_S32 &lt; 5; Aux_S32++) {     A2[Aux_S32] = A1[Aux_S32]; }  /* Custom code */ {     Sa1_OutPort = A2[Sa1_InPort - 1].m12; }</pre>	<pre>/* Custom code */ {     Sa1_OutPort = A1[Sa1_InPort - 1].m12; }</pre>

### 理由

- コードの効率性

信号分割時の中間変数の除去を改善

TargetLink で、信号の分岐によって生じるテンポラリ変数を置き換える機能が、値の割り当てが条件付きで行われる場合でも改善されました。構造体変数と構造体配列変数では、削除する変数と置換する変数の初期値が同じであれば、削除が可能になりました。

TargetLink ≤ 2022-B	TargetLink 2023-B
<pre>if (Sa1_InPort1 != 0) {     LastValid_State = Sa1_InPort; } Sa1_OutPort1 = LastValid_State.c0; Sa1_OutPort = LastValid_State ;</pre>	<pre>if (Sa1_InPort1 != 0) {     Sa1_OutPort = Sa1_InPort; } Sa1_OutPort1 = Sa1_OutPort.c0;</pre>

さらに、置換されるシンボルにエイリアスがある場合、つまりシンボルがポインタであるか、シンボルへのポインタが存在する場合、数値変数を置換することができません。

#### 理由

- コードの効率性
- コードサイズ
- コードの安定性
- ユーザの期待に応えるため
- デバッグのしやすさ
- モデルとコードのマッチングが容易
- コンパイルの容易化

#### コードスケジューリングによる構造体／構造体配列タイプの間変数の除去の改善

TargetLink は、構造体中間変数の削除が可能な状況を作り出すために、ブロッキングステートメントを移動します。

#### 例 1

##### TargetLink ≤ 2022-B

```
Sa4_In.RunA_NonAROut = NonAROut_RunA;

Sa4_In.RootIn3 = InPort3_Root;

NonARBusInport_RunB = X_Unit_Delay_Root; // blocks the replacement of Sa4_In by X_Unit_Delay_Root

X_Unit_Delay_Root = Sa4_In;
```

##### TargetLink 2023-B

```
NonARBusInport_RunB = X_Unit_Delay_Root;

X_Unit_Delay_Root.RunA_NonAROut = NonAROut_RunA;

X_Unit_Delay_Root.RootIn3 = InPort3_Root;
```



## 例 2

## TargetLink ≤ 2022-B

```

Sa2_BusCreation(&BusCreationOut1, BusCreationOut2_signal1, &BusCreationOut2_signal3);
for (Aux_S32 = 0; Aux_S32 < EW1; Aux_S32++)
{
    Sa1_signal1_a[Aux_S32] = X_Sa1_Unit_Delay[Aux_S32];
    X_Sa1_Unit_Delay[Aux_S32] = BusCreationOut2_signal1[Aux_S32];
    Sa1_signal2_a[Aux_S32] = X_Sa1_Unit_Delay_a[Aux_S32];
    X_Sa1_Unit_Delay_a[Aux_S32] = BusCreationOut2_signal2[Aux_S32];
}

Sa1_signal3_a = X_Sa1_Unit_Delay_b;
X_Sa1_Unit_Delay_b = BusCreationOut2_signal3;

for (Aux_S32 = 0; Aux_S32 < 8; Aux_S32++)
{
    Sa1_signal1[Aux_S32] = X_Sa1_Unit_Delay1.signal1[Aux_S32];
    Sa1_signal2[Aux_S32] = X_Sa1_Unit_Delay1.signal2[Aux_S32];
}
Sa1_signal3 = X_Sa1_Unit_Delay1.signal3;
X_Sa1_Unit_Delay1 = BusCreationOut1;

```

## TargetLink 2023-B

```

for (Aux_S32 = 0; Aux_S32 < EW1; Aux_S32++)
{
    Sa1_signal1_a[Aux_S32] = X_Sa1_Unit_Delay[Aux_S32];
    Sa1_signal2_a[Aux_S32] = X_Sa1_Unit_Delay_a[Aux_S32];
}

Sa1_signal3_a = X_Sa1_Unit_Delay_b;

for (Aux_S32 = 0; Aux_S32 < 8; Aux_S32++)
{
    Sa1_signal1[Aux_S32] = X_Sa1_Unit_Delay1.signal1[Aux_S32];
    Sa1_signal2[Aux_S32] = X_Sa1_Unit_Delay1.signal2[Aux_S32];
}

Sa1_signal3 = X_Sa1_Unit_Delay1.signal3;
Sa2_BusCreation(&X_Sa1_Unit_Delay1, X_Sa1_Unit_Delay, &X_Sa1_Unit_Delay_b);

for (Aux_S32 = 0; Aux_S32 < EW1; Aux_S32++)
{
    X_Sa1_Unit_Delay_a[Aux_S32] = BusCreationOut2_signal2[Aux_S32];
}

```

## 理由

- コードの効率性

**a = a** の状況における構造体配列／構造体の中間変数の除去

ループのコンテキストで、除去結果が **a = a** となる場合は、TargetLink は構造体／構造体配列タイプのより多くの中間変数を削除できるようになりました。

## コード例:

```
for (Aux_S32 = 0; Aux_S32 < 4; Aux_S32++)
{
    Sa1_DSR_AosM2x3_T3[Aux_S32] = DSM_AosT3[1][2].a1[Aux_S32];

    DSM_AosT3[1][2].a1[Aux_S32] = Sa1_DSR_AosM2x3_T3[Aux_S32];
}

DSM_AosT3[0][2].a1[Index] = Sa1_DSR_AosM2x3_T3[Index];
```

これは次のように最適化されます。

```
DSM_AosT3[0][2].a1[Index] = DSM_AosT3[1][2].a1[Index];
```

## 理由

- コードの効率性

構造体配列の RTE API リターン  
ポインタの削除

以前のバージョンの TargetLink では、RTE API のリターンポインタを削除できるのは数値変数と構造体変数の場合のみでした。現在のバージョンでは、TargetLink は構造体ベクトル変数の RTE API リターンポインタを削除することができます。数値行列変数と同様に、行列の配列変数はキャストによって削除されることはありません。

## TargetLink ≤ 2022-B

```
const myStruct2 * p_MyImplicitIRV_T1_3;

p_MyImplicitIRV_T1_3 = Rte_IrvIRead_Run_MyImplicitIRV_T1_3();

Sa3_Selector1 = p_MyImplicitIRV_T1_3[1];
```

## TargetLink 2023-B

```
Sa3_Selector1 = Rte_IrvIRead_Run_MyImplicitIRV_T1_3()[1];
```

## 理由

- コードの効率性
- 整合性の向上

構造体配列コンポーネントの中間  
変数の削除

TargetLink で、後から 1 つの要素のみが選択される場合に、中間変数として非スカラー構造体タイプを持つ構造体コンポーネントを削除できるようになりました:

## TargetLink ≤ 2022-B

```
for i
{
    S.v[i] = InAos[3].v[i]
}

Out = S.v[MyIdx]
```

## TargetLink 2023-B

```
Out = InAos[3].v[MyIdx]
```

## 理由

- コードの効率性
- コードサイズ

## 条件付きで実行される制御フローにおけるコードの重複

TargetLink 2023-B では、条件付きで実行される制御フローの最適化が改善されました。

TargetLink は、Switch ブロック出力のような制御フローステートメントの出力変数を消費する関数呼び出しを、それぞれの制御フロー分岐に個々の関数呼び出しを配置することによって、相互に排他的な条件付き実行制御フロー分岐に移動できるようになりました。この変更の目的は、少なくとも一部の分岐の内部で以前の出力変数を除去し、除去できない場合は平均スタック消費量を減らすことです。

## TargetLink ≤ 2022-B

```
Int16 o;
if (...) {
    o = q;
} else {
    if (...) {
        funcDef1(o);
    } else {
        funcDef2(o);
    }
}
funcUse(o);
```

## TargetLink 2023-B

```
if (...) {
    funcUse(q);
} else {
    if (...) {
        Int16 o_a;
        funcDef1(o_a);
        funcUse(o_a);
    } else {
        Int16 o_b;
        funcDef2(o_b);
        funcUse(o_b);
    }
}
```

出力変数を除去することだけがメモリを節約できます。この最適化によってその後の最適化も容易になるため、コードサイズは必ずしも増加しません。

この最適化は、UtilizeCodeDuplicationInControlFlowBranches Code Generator オプションを使用して制御することができます。

[「条件制御フロー分岐でのバッファ変数の書き込みアクセス」](#)(237 ページ)も参照してください。

## 理由

- コードの効率性
- コードサイズ
- ユーザの期待に応えるため

**移行の問題** この最適化は、アクセス関数のローカル値コピーの作成と競合する可能性があります。DD CreateLocalValueCopy プロパティを参照してください。

### 構造体配列変数のインデックスアクセスの簡素化

TargetLink では、ルートまたはサブ構造体レベルの構造体配列変数のインデックスアクセスが簡素化されました。特に、キャストが削除され、代数的な単純化と定数の畳み込みが行われるようになりました。

TargetLink ≤ 2022-B	TargetLink 2023-B
<code>AoS[(T) &lt;expr&gt;].c</code>	<code>AoS[&lt;expr&gt;].c</code>
<code>AoS[idx].sub[(T) (2 - 1)].c</code>	<code>AoS[idx].sub[1].c</code>

#### 理由

- コードの効率性
- ユーザの期待に応えるため
- 可読性
- 余分なコードの抑制

### インデックス式の初期値伝播

TargetLink は、インデックス式にのみ配置されている読み取り専用変数の初期値を伝播できるようになりました。以前のバージョンでは、この最適化を発動させるには、インデックスアクセス以外のアクセスが少なくとも 1 回必要でした。

出発点	TargetLink ≤ 2022-B	TargetLink 2023-B
<code>T a = 5;</code> <code>T c = 3;</code>  <code>b[a] = a + d[c];</code>	<code>T c = 3;</code>  <code>b[5] = 5 + d[c];</code>	<code>b[5] = 5 + d[3];</code>

#### 理由

- バグ修正
- コードの効率性
- ユーザの期待に応えるため
- 整合性の向上
- 副作用
- 余分なコードの抑制

### Stateflow 入力の構造体変数

バス信号に接続され、Create input variable チェックボックスがオフになっている Stateflow 入力には、プレーン変数を指定する代わりに、以下の構造体変数を使用または作成します。

- 構造体先行変数
- 構造体 IF 変数

#### 例 1

##### 注記

構造体代入には構造体全体のモデル化されたコピーが含まれ、以前に未使用であったコンポーネントは省略することができます。

**TargetLink ≤ 2022-B**

```

for (Sc6_For_Iterator_it = 1; Sc6_For_Iterator_it <= Sc1_Dim1_1; Sc6_For_Iterator_it++)
{
    ...
    for (Aux_S32 = 0; Aux_S32 < 5; Aux_S32++)
    {
        MyAoSType3_1_e_Assignment[Sc1_Dim1_1 - 1].y_2[Aux_S32] =
        MyAoSType3_1_c[1].m_1.z_3[Sc6_For_Iterator_it - 1].y_2[Aux_S32];
    }

    MyAoSType3_1_e_Assignment[Sc1_Dim1_1 - 1].y_3 =
    MyAoSType3_1_c[1].m_1.z_3[Sc6_For_Iterator_it - 1].y_3;
    ...
}

```

**TargetLink 2023-B**

```

for (Sc6_For_Iterator_it = 1; Sc6_For_Iterator_it <= Sc1_Dim1_1; Sc6_For_Iterator_it++)
{
    ...
    MyAoSType3_1_e_Assignment[Sc1_Dim1_1 - 1] =
    MyAoSType3_1_c[1].m_1.z_3[Sc6_For_Iterator_it - 1];
    ...
}

```

**例 2****TargetLink ≤ 2022-B**

```

static Float64 Ca1_inbus;
static Float64 Ca1_inbus_a;
static Int32 Ca1_inbus_b[5];
static Int32 Ca1_inbus_b[5];
static Int8 Ca1_inbus_c;
static UInt8 Ca1_inbus_e[5];

static void Ca1_Chart1(void)
{
    Int32 Aux_S32;

    Ca1_outbus.y_3.v_1 = Ca1_inbus_c;
    Ca1_outbus.y_3.v_2 = Ca1_inbus_d;
    for (Aux_S32 = 0; Aux_S32 < 5; Aux_S32++)
    {
        Ca1_outbus.y_2[Aux_S32] = Ca1_inbus_b[Aux_S32];
        Ca1_outbus.y_3.v_3[Aux_S32][0] = Ca1_inbus_e[Aux_S32];
    }
    Ca1_outbus.y_3.v_4[0] = Ca1_inbus_f[0];
    Ca1_outbus.y_3.v_4[1] = Ca1_inbus_f[1];
    Ca1_outbus.y_1.x_1.w = Ca1_inbus + 0.01;
    Ca1_outbus.y_1.x_2 = Ca1_inbus_a + 0.02;
}

static void Sa6_ForIterator_AoST3_1_a(...)
{
    ...
    for (Sa6_For_Iterator_it = 1;
        Sa6_For_Iterator_it <= ((Int16) Sa6_Dimension1);
        Sa6_For_Iterator_it++)
    {
        ...
        Ca1_inbus = MyAoSType3_1_m_1_6.z_3[Sa6_For_Iterator_it - 1].y_1.x_1.w;
        Ca1_inbus_a = MyAoSType3_1_m_1_6.z_3[Sa6_For_Iterator_it - 1].y_1.x_2;
    }
}

```

## TargetLink ≤ 2022-B

```

Ca1_inbus_c = MyAoSType3_1_m_1__6_.z_3[Sa6_For_Iterator_it - 1].y_3.v_1;
Ca1_inbus_d = MyAoSType3_1_m_1__6_.z_3[Sa6_For_Iterator_it - 1].y_3.v_2;

for (Aux_S32 = 0; Aux_S32 < 5; Aux_S32++)
{
    Ca1_inbus_b[Aux_S32] = MyAoSType3_1_m_1__6_.z_3[Sa6_For_Iterator_it - 1].y_2[Aux_S32];
    Ca1_inbus_e[Aux_S32] = MyAoSType3_1_m_1__6_.z_3[Sa6_For_Iterator_it - 1].y_3.v_3[Aux_S32][0];
}
Ca1_inbus_f[0] = MyAoSType3_1_m_1__6_.z_3[Sa6_For_Iterator_it - 1].y_3.v_4[0];
Ca1_inbus_f[1] = MyAoSType3_1_m_1__6_.z_3[Sa6_For_Iterator_it - 1].y_3.v_4[1];

Ca1_Chart1();
}
...
}

```

## TargetLink 2023-B

```

static struct m_1_z_3_t_tag Sa6_Selector1;

static void Ca1_Chart1(void)
{
    Ca1_outbus = Sa6_Selector1;

    Ca1_outbus.y_1.x_1.w = Sa6_Selector1.y_1.x_1.w + 0.01;
    Ca1_outbus.y_1.x_2 = Sa6_Selector1.y_1.x_2 + 0.02;
}

static void Sa6_ForIterator_AoST3_1_a(...)
{
    ...
    for (Sa6_For_Iterator_it = 1;
        Sa6_For_Iterator_it <= ((Int16) Sa6_Dimension1);
        Sa6_For_Iterator_it++)
    {
        ...
        Sa6_Selector1 =
            MyAoSType3_1_m_1__6_.z_3[Sa6_For_Iterator_it - 1];

        Ca1_Chart1();
        ...
    }
    ...
}

```

## 理由

- バグ修正
- コードの効率性
- コードサイズ
- 整合性の向上
- 可読性

**移行の問題** チャート内部でアクセスされる Stateflow チャート入力の構成要素が1つだけ、またはわずかしかない例1のような状況では、構成要素が多く、または大きい構造体タイプでは、構造体コピーの効率が低下する可能性があります。

以下の対応策を使用することができます。

- Bus Selector ブロックを使用して、関連するバスエレメントを選択し、構造体からの Stateflow チャートの入力をそれぞれのコンポーネントの番号とデータタイプに応じて調整します。
- StructAssignmentDecompositionAuxVarMemoryUsageThreshold コードジェネレータオプションまたは StructAssignmentDecompositionStatementMultiplicationThreshold コードジェネレータオプションを調整して、TargetLink が構造体代入を分割し、不要なコンポーネントまたは構造体変数の未使用部分への不要な代入を削除できるようにします。

## 関連トピック

### 参考文献

[CreateLocalValueCopy](#) (『TargetLink Data Dictionary Reference』)
 [Switch ブロック](#) (『TargetLink モデルエレメントリファレンス』)
 [UtilizeCodeDuplicationInControlFlowBranches](#) (『TargetLink モデルエレメントリファレンス』)

## Adaptive-AUTOSAR 関連と CPP 関連

### cpp ファイル内の Cpp キャスト

cpp ファイルは一般的に C キャストではなく C++キャストを使うようになっています。

C++キャスト	説明
<code>static_cast&lt;T&gt;(expr)</code>	ほとんどの数値 C キャストを置き換えます。
<code>reinterpret_cast&lt;T&gt;(expr)</code>	ビットパターンの再解釈につながります。この C++キャストは、AddressMapped マクロと行ポインタキャストで使用されます。
<code>const_cast&lt;T&gt;(expr)</code>	この C++キャストは、ポインタターゲットの <code>const</code> または <code>volatile</code> プロパティを変更するために使用されます。

### 理由

- C++規格準拠

### 名前空間修飾の変更

新しいコードジェネレータオプション `NamespaceQualification` は、生成される C++コード内のシンボルの名前空間修飾を変更しました。デフォルトでは、コードジェネレータは、`::`をプレフィックスとする完全修飾名前空間を使用します。これにより、たとえば、`::A::B` の内部であれば、`A::x` が `::A::x` と `::A::B::A::x` にマッチングするというように、同じ名前空間の識別子が誤ってマッチングしてしまうことを防ぐことができます。名前空間 `std` と `ara` はこの対象から除外されています。

TargetLink ≤ 2022-B	TargetLink 2023-B
<pre>namespace A {     namespace B     {         using MyType = std::int32_t;         A::B::MyType MyVar = 5;         struct MyStruct {             MyType a;         };     }      void func() {         A::B::MyType MyLocalVar = A::B::MyVar;     } }</pre>	<pre>namespace A {     namespace B     {         using MyType = std::int32_t;         ::A::B::MyType MyVar = 5;         struct MyStruct {             ::A::B::MyType a;         };     }      void func() {         ::A::B::MyType MyLocalVar = ::A::B::MyVar;     } }</pre>

## 理由

- バグ修正
- 精度

## 名前空間宣言での開く側(左側)の波括弧

TargetLink ≤ 2022-B	TargetLink 2023-B
<pre>namespace A {     namespace B {         using MyType = std::int32_t;     } }</pre>	<pre>namespace A {     namespace B     {         using MyType = std::int32_t;     } }</pre>

## 理由

- 整合性の向上
- 可読性

## マクロとして実装される列挙データタイプの変更

マクロとして実装され、CreateTypedef プロパティが off に設定されている列挙データタイプでは、名前空間が宣言されず、名前空間の閉じる側(右側)の波括弧が最後の列挙要素と同じ行に出力されなくなりました。

TargetLink ≤ 2022-B	TargetLink 2023-B
<pre>namespace MyNamespace {     using EnumWithNamespace_CreateTypeDefOn =     std::int32_t;     #define Red3     (static_cast&lt;EnumWithNamespace_CreateTypeDefOn&gt;(1))     #define Yellow3     (static_cast&lt;EnumWithNamespace_CreateTypeDefOn&gt;(0))     #define Red (static_cast&lt;std::int32_t&gt;(1))     #define Yellow (static_cast&lt;std::int32_t&gt;(0)) } }</pre>	<pre>#define Red (static_cast&lt;std::int32_t&gt;(1)) #define Yellow (static_cast&lt;std::int32_t&gt;(0)) namespace MyNamespace {     using EnumWithNamespace_CreateTypeDefOn =     std::int32_t;     #define Red3     (static_cast&lt;EnumWithNamespace_CreateTypeDefOn&gt;(1))     #define Yellow3     (static_cast&lt;EnumWithNamespace_CreateTypeDefOn&gt;(0)) }</pre>



**理由**

- バグ修正
- 可読性

タイプガード名に名前空間が含まれる

制約のあるデータタイプについては、コード内にタイプガードが作成されます。これらのデータタイプにも C++ 名前空間がある場合、これらのデータタイプには、<NS1>\_<NS2>\_...<NSn>\_<Typename>\_TYPE などの、個々の名前空間名を含む拡張タイプガード名が与えられます。

TargetLink ≤ 2022-B	TargetLink 2023-B
<pre>namespace my {   namespace types   {     #ifndef MyInt16_WithNS_TYPE     #define MyInt16_WithNS_TYPE     using MyInt16_WithNS = std::int16_t;     #endif /* MyInt16_WithNS_TYPE */   } }</pre>	<pre>namespace my {   namespace types   {     #ifndef my_types_MyInt16_WithNS_TYPE     #define my_types_MyInt16_WithNS_TYPE     using MyInt16_WithNS = std::int16_t;     #endif /* my_types_MyInt16_WithNS_TYPE */   } }</pre>

**理由**

- コードの安定性
- 可読性
- C++ 規格準拠

**ファイルヘッダーの言語コメント**

TargetLink ≤ 2022-B	TargetLink 2023-B
<pre>*** Simulink model      : aar_communication *** TargetLink subsystem : aar_communication/S1 *** Codefile           : S1.h *** *** Generated by TargetLink, the dSPACE production quality code generator</pre>	<pre>*** Simulink model      : aar_communication *** TargetLink subsystem : aar_communication/S1 *** Codefile           : S1.h *** Language           : Cpp *** *** Generated by TargetLink, the dSPACE production quality code generator</pre>

**理由**

- ユーザの期待に応えるため

## 構造体関連と構造体配列関連

より多くのモデリング状況において、バス対応ブロックが構造体変数を継承します。

Inherit Properties チェックボックスが選択されているバス対応ブロックは、バス信号またはバス配列信号に接続されます。このブロックは、構造化されたデータタイプまたは構造体配列データタイプを持つ出力変数やブロックパラメータを生成するために、先行ブロックから構造化されたデータタイプを継承しようとします。これは常

に可能というわけではありません。そのような場合には、次のようなことが発生します。

- ブロックが構造体配列タイプの変数を生成しようとする、コード生成はエラーになりキャンセルされます。
- ブロックが構造体変数を 1 つ生成しようとする、代わりに n 個の非構造化変数が生成されます。

TargetLink 2023-B では、このようなブロックが先行ブロックから構造化されたデータタイプを継承できる状況が増えています。

- 構造化された DD Typedef オブジェクトは、DD Typedef オブジェクトを参照するコンポーネントを持っています。この DD Typedef オブジェクトには制約があり、その CreateTypedef プロパティは off に設定されています。
- このデータタイプのリーフ構造体コンポーネントの Min/Max 値が設定されます。

#### 理由

- バグ修正
- コードの効率性
- コードサイズ
- 整合性の向上

### 構造体変数の一貫した取り扱い

暗黙構造体変数の未使用コンポーネントは、固定データタイプのインスタンスコンポーネントと同じ方法で生成され、同期化されます。

さらに、以下のコード変更も参照してください。

- Original size... のコメントは、再利用構造体の n-D テーブルにのみ使用されます。
- 未使用の暗黙構造体変数コンポーネントの Description、Min、Max の値は表示されません。

#### 理由

- バグ修正
- 整合性の向上

### 暗黙的構造体の構造体コンポーネントのソート順の変更

構造体コンポーネントのソート基準が次のように変更されました。

ソート基準	変更
BaseType	TargetLink では、まずベースタイプの名前でソートし、次に次元数でソートするようになりました。
Type	TargetLink では、まずタイプ名でソートし、次に次元数でソートするようになりました。
Size	TargetLink では、構造体コンポーネントの次元数がサイズ計算に含まれるようになりました。たとえば、Int16[2]タイプのコンポーネントは、16-bit ではなく 32-bit としてカウントされます。

これは、以下のコンテキストで暗黙的に生成される構造体に影響を及ぼします。

- 関数の再利用
- バリエーションコーディング
- パスポートブロックから生成される暗黙構造体

**理由**

- バグ修正
- コードの効率性
- ユーザの期待に応えるため

**移行の問題** 固定された構造体タイプのレイアウトが必要な場合は、Optimization プロパティに COMPONENTS\_SORTABLE を指定せずに、構造体 DD Typedef オブジェクトを使用します。

**追加情報** 「影響と相互作用」(222 ページ)を参照してください。

### 暗黙的に生成されたバス構造体の構造体コンポーネントの並べ替え

Create implicit struct チェックボックスが選択されているバスポートブロックから生成された構造体の構造体コンポーネントを並べ替える場合、TargetLink はサブ構造体を考慮するようになりました。その結果、構造体コンポーネントの順序が異なったり、サブ構造体の名前が異なったりすることがあります。

TargetLink ≤ 2022-B	TargetLink 2023-B
<pre>struct BSS_IP_Sa2_InPort {     Int32 InPort_b_i32;     Int16 Sa2_a_i16; };  struct BSS_IP_Sa2_InPort_a {     Int32 InPort_b_i32;     UInt32 Sa2_InPort;     Int16 InPort_a_i16; };  struct BSS_IP_Sa2_InPort_b {     Int32 Sa2_InPort_a[3];     Int16 Sa2_InPort; }; /* Description: bus inport struct */  struct BS_IP_Sa2_InPort {     Int32 InPort_f_i32_vec5[5];     Int32 InPort_b_i32;     Int16 InPort_d_i16_vec5[5];     Int16 Sa2_a_i16;     struct BSS_IP_Sa2_InPort InPort_e_sub;     struct BSS_IP_Sa2_InPort_a InPort_l_sub2;     struct BSS_IP_Sa2_InPort_b InPort_l_sub3; };</pre>	<pre>struct BSS_IP_Sa2_InPort {     Int32 Sa2_InPort_a[3];     Int16 Sa2_InPort; };  struct BSS_IP_Sa2_InPort_a {     Int32 InPort_b_i32;     UInt32 Sa2_InPort;     Int16 InPort_a_i16; };  struct BSS_IP_Sa2_InPort_b {     Int32 InPort_b_i32;     Int16 Sa2_a_i16; };  struct BS_IP_Sa2_InPort {     struct BSS_IP_Sa2_InPort InPort_l_sub3;     struct BSS_IP_Sa2_InPort_a InPort_l_sub2;     struct BSS_IP_Sa2_InPort_b InPort_e_sub;     Int32 InPort_f_i32_vec5[5];     Int32 InPort_b_i32;     Int16 InPort_d_i16_vec5[5];     Int16 Sa2_a_i16; };</pre>

**理由**

- バグ修正
- コードの効率性
- ユーザの期待に応えるため
- 整合性の向上

**追加情報** 「影響と相互作用」(222 ページ)を参照してください。

### 未使用の構造体コンポーネントのユーザ定義変数クラス

構造体タイプが明示的か暗黙的に関係なく、未使用の構造体コンポーネントのユーザ定義変数クラスが考慮されるようになりました。

このため、次のようにコードが変更される場合があります。

- 未使用のコンポーネントについては、アクセス関数が生成されます。
- ユーザ定義変数クラスの Optimization プロパティが ERASABLE に設定されている場合、未使用のコンポーネントはデータタイプから省略されます。

#### 理由

- バグ修正
- ユーザ経験と一致
- 整合性の向上

#### シングルトンの次元を小さくする

以下の状況では、次元は常に小さくなります。

- 構造体の配列
- 構造体のルート次元のみ。
- DD Variable オブジェクトがない
- [1x1]行列および[1]ベクトル
- 次元は明示されません。

#### 理由

- バグ修正

#### Restart 関数でのテンポラリ変数の並べ替え

構造体配列変数を初期化するために生成されるテンポラリ変数は、SLLocal の代わりに変数クラス SLLocalInit を取得します。その結果、Restart 関数の変数宣言の並び順が変更されます。

#### TargetLink ≤ 2022-B

```
/* SLLocal: Default storage class for local variables | Width: N.A. */
ShV5hV5h_tp M2x2_ShV5hV5h_Aux = {
    0, /* Comp1Scalar */
    ...
};

/* SLLocal: Default storage class for local variables | Width: 32 */
Int32 Aux_S32;
```

#### TargetLink 2023-B

```
/* SLLocal: Default storage class for local variables | Width: 32 */
Int32 Aux_S32;

/* SLLocalInit: Default storage class for local variables with init value | Width: N.A. */
ShV5hV5h_tp M2x2_ShV5hV5h_Aux = {
    0, /* Comp1Scalar */
    ...
};
```

#### 理由

- バグ修正
- 整合性の向上

## Classic AUTOSAR 関連

### Classic AUTOSAR 配列のデータタイプ

Classic AUTOSAR 配列のデータタイプが変更され、基礎となる TargetLink 基本タイプによって命名されなくなりました。代わりに、スカラー要素のタイプ名が使用されます。

次のように設定された MyArrayType と呼ばれる DD Typedef オブジェクトについて考察します。

- CreateTypedef は off に設定
- BaseType は sint8 に設定
- AutosarArrayElementRef は MyScalarType に設定

コードは次のように変更されます。

TargetLink ≤ 2022-B	TargetLink 2023-B
<code>sint8 DE_Array[3]; Rte_Read_Swc_R_Port_SRI_DE_Array(DE_Array);</code>	<code>MyScalarType DE_Array[3]; Rte_Read_Swc_R_Port_SRI_DE_Array(DE_Array);</code>

これはまた、サーバルナブルの仮パラメータについても影響を及ぼします。

```
void Runnable(MyScalarType * Arg_Matrix)
{
    MyScalarType (*p_Arg_Matrix)[2];

    p_Arg_Matrix = (sint8 (*)[2]) Arg_Matrix;
```

キャストは常にベースタイプに起因するため、影響を受けません。

#### 理由

- バグ修正
- ユーザ経験と一致
- 整合性の向上

### 条件制御フロー分岐でのバッファ変数の書き込みアクセス

以下の RTE API 関数で生成されるコードは、変更されている可能性があります。

- Rte\_Write
- Rte\_IWrite
- Rte\_IrvWrite
- Rte\_IrvIWrite

バッファ変数全体を渡す書き込みアクセスは、条件付き制御フロー(if ... then ... else コンストラクト)で並列に実行されるようになりました。

[「条件付きで実行される制御フローにおけるコードの重複」](#)(227 ページ)を参照してください。

### AUTOSAR 配列の演算結果タイプへの余分なキャストは省略されます。

2 つの 32 ビットオペランドが加算または減算される場合、TargetLink は AUTOSAR 配列の演算結果タイプへのキャストを生成しなくなりました。

TargetLink ≤ 2022-B	TargetLink 2023-B
<code>I32Out = (Int32)(I32VecIn1[0] + I32VecIn2[0]);</code>	<code>I32Out = I32VecIn1[0] + I32VecIn2[0];</code>

**理由**

- バグ修正
- コードの効率性
- 整合性の向上
- 可読性
- 余分なコードの抑制

## その他のコード変更

### Stateflow コード内のマクロ

Stateflow チャートからコードを生成するとき、ユーザ定義変数クラスの Optimization プロパティが ERASABLE に設定され、OmitCastOfMacroValue プロパティが on に設定されている場合、ユーザ定義変数クラスを持つマクロはその初期値に置き換えられます。

**理由**

- バグ修正
- コードの効率性
- 整合性の向上

### 改良された SQRT パターン

SQRT ブロックへの入力が整数で結果が浮動小数点の場合の SQRT パターンが改良されました。

TargetLink ≤ 2022-B	TargetLink 2023-B
<pre>if(in &lt; 0) {   out = 0; } else {   out = sqrt(in); }</pre>	<pre>if(in &lt;= 0) {   out = 0; } else {   out = sqrt(in); }</pre>

**理由**

- コードの効率性

### LOG マクロ呼び出しでの列挙の整数キャスト

列挙は、内部ロギング構造では整数として表現されます。そのため、コードジェネレータは、LOG マクロの呼び出しに、列挙の基礎となる整数タイプへのキャストを含めるようになりました。

TargetLink ≤ 2022-B	TargetLink 2023-B
<pre>LOG_MATRIX(MyEnumMatrix, _MyEnumMatrix, 3, 2, MyEnumMatrix);</pre>	<pre>LOG_MATRIX(MyEnumMatrix, _MyEnumMatrix, 3, 2, (Int32) MyEnumMatrix);</pre>

**理由**

- バグ修正

## 列挙キャストの符号化

列挙キャストの符号化が修正されました。以前の TargetLink バージョンでは、基礎となる列挙データタイプが符号なしであっても、列挙データタイプが符号付きであると主張する場合があります。

TargetLink ≤ 2022-B	TargetLink 2023-B
<code>Aux_b = (Int32) (((Int32) Ca1_LightColorsIn) &lt;&lt; 1);</code>	<code>Aux_b = (Int32) (((UInt32) Ca1_LightColorsIn) &lt;&lt; 1);</code>

## 理由

- バグ修正

余分な `tl_basetypes.h` のインクルード

モジュールが列挙データタイプしか含まない場合、`tl_basetypes.h` のインクルードは削除されます。

TargetLink ≤ 2022-B	TargetLink 2023-B
<pre>#include "tl_basetypes.h"  typedef enum ...  enum T..._tag ...</pre>	<pre>typedef enum ...  enum T..._tag ...</pre>

## 理由

- バグ修正
- ユーザ経験と一致

## 関数呼び出し時のステートメントコメント内の順序とコメント部分の重複

関数呼び出しの実際のパラメータに対するコメント部分の順序が次のように変更されました。

TargetLink ≤ 2022-B	TargetLink 2023-B
<pre>/* call of function: Subsystem/F64FcnU64Outside # combined # update(s) for inport Subsystem/F64FcnU64Outside/In3 # combined # update(s) for inport Subsystem/F64FcnU64Outside/In2 # combined # update(s) for inport Subsystem/F64FcnU64Outside/In1 */ Sa2_F64FcnU64Outside((Float64) Sa1_U64Source, 1, 0., &amp;Sa2_Out1);</pre>	<pre>/* call of function: Subsystem/F64FcnU64Outside # combined # update(s) for inport Subsystem/F64FcnU64Outside/In1 # combined # update(s) for inport Subsystem/F64FcnU64Outside/In2 # combined # update(s) for inport Subsystem/F64FcnU64Outside/In3 */ Sa2_F64FcnU64Outside((Float64) Sa1_U64Source, 1, 0., &amp;Sa2_Out1);</pre>

# combined #コメントの変更点:

TargetLink ≤ 2022-B	TargetLink 2023-B
<pre>/* Sum: Subsystem/Sum # combined # Unit delay: Subsystem/Unit Delay # combined # TargetLink output: Subsystem/Out1 # combined # Unit delay: Subsystem/Unit Delay */ Sa1_Out1 = (Int16) (Sa1_InPort + Unit_Delay_2);</pre>	<pre>/* Sum: Subsystem/Sum # combined # Unit delay: Subsystem/Unit Delay # combined # TargetLink output: Subsystem/Out1 */ Sa1_Out1 = (Int16) (Sa1_InPort + Unit_Delay_2);</pre>

TargetLink ≤ 2022-B	TargetLink 2023-B
<pre>/* # combined # Custom code: Path/to/Custom Code Block: &lt;&lt; output code &gt;&gt;   Update of input(s) for custom code block : Path/to/Custom Code Block */ {   Sa2_Custom_Code_Block_y = (((uint16) 1)) &lt;&lt; (Var1); } </pre>	<pre>/* Custom code: Path/to/Custom Code Block: &lt;&lt; output code &gt;&gt;   Update of input(s) for custom code block : Path/to/Custom Code Block */ {   Sa2_Custom_Code_Block_y = (((uint16) 1)) &lt;&lt; (Var1); } </pre>
<pre>/* Path1/to/Block1: Omitted comparison with constant. update(s) for inport Path1/to/Block2 Path1/to/Block3: Omitted comparison with constant. # combined # update(s) for inport Path1/to/Block4 # combined # update(s) for inport Path1/to/Block5 */ if (Ca4_Var1 &amp;&amp; Var2 &amp;&amp; Ca5_Var3) {? </pre>	<pre>/* Path1/to/Block1: Omitted comparison with constant. # combined # update(s) for inport Path1/to/Block2 Path1/to/Block3: Omitted comparison with constant. # combined # update(s) for inport Path1/to/Block4 # combined # update(s) for inport Path1/to/Block5 */ if (Ca4_Var1 &amp;&amp; Var2 &amp;&amp; Ca5_Var3) {?? </pre>
<pre>/* Switch: Path1/to/SwitchBlock1 # combined # Product: Path2/to/ProductBlock1 # combined # Unit delay: Path3/to/UnitDelayBlock1 */ Aux_S16 += (X_Sa2_Unit_Delay * Sa3_Abs1); </pre>	<pre>/* # combined # Switch: Path1/to/SwitchBlock1 # combined # Product: Path2/to/ProductBlock1 # combined # Unit delay: Path3/to/UnitDelayBlock1 */ Aux_S16 += (X_Sa2_Unit_Delay * Sa3_Abs1); </pre>

## 理由

- バグ修正
- 整合性の向上
- 可読性

プリプロセッサの制御フローにおける関数宣言と定義のインデント(字下げ)

プリプロセッサの制御フローにおける変数の宣言と定義と同様に、関数の宣言と定義もインデント(字下げ)されます。

TargetLink ≤ 2022-B	TargetLink 2023-B
<pre>#if U extern void Sa2_ss_A(void); #endif </pre>	<pre>#if U   extern void Sa2_ss_A(void); #endif </pre>

## 理由

- 整合性の向上

Volatile は自動保存への変更時に省略される

volatile キーワードは、変数が自動保存期間に変更されるときに除外されます。これは、Advice 31419 の出力をトリガします。しかし、変数が ScopeReducedClass エントリによって定義されたスコープ縮小チェーンを持つ変数クラスを持っている場合、Scope=local、Storage=default の変数クラスの Volatile プロパティは、望ましい動作を指定し、アドバイスは与えられません。

## 理由

- バグ修正
- コードの効率性
- 余分なコードの抑制



**コメント中の BlockType**

ほとんどのブロックでは、BlockType は量産コードのコメントで次のように設定されま  
す。

- TL\_プレフィックスを除いたマスクタイプ。
- Simulink ブロックタイプ

**理由**

- バグ修正
- 整合性の向上
- 可読性

**MinMax ブロックパターンでマク  
ロを制限する**

MinMax ブロックに 1 つの非スカラー入力がある場合、TargetLink は max 関数  
に以下のコードを生成します。

```
MinMaxBlockOut = MinValueOfBlockOutputType;
if(In[0] > MinMaxBlockOut) MinMaxBlockOut = In[0];
...
if(In[n] > MinMaxBlockOut) MinMaxBlockOut = In[n];
```

浮動小数点では、生成されるコードが次のようになります。

TargetLink ≤ 2022-B	TargetLink 2023-B
<pre>MinMaxBlockOut = -1.7976931348623157e+308; if ...</pre> <p>この変更は、Float32 と min 関数にも適用されます。</p>	<pre>MinMaxBlockOut = t1_FLOAT64NMIN; if ...</pre>

**理由**

- バグ修正
- コードの安定性
- 整合性の向上
- 可読性

**ConformingFreestanding fabs**

UseTLImplementationForFabsFunction コードジェネレータオプション(旧バー  
ジョンでは ConformingFreestandingImplementation に該当 )を有効にする  
と、TargetLink は条件式を使用します。

TargetLink ≤ 2022-B	TargetLink 2023-B
<pre>#include "t1_fabs.h" Sa1_AbsF64Out = fabs(Sa1_F64In);</pre>	<pre>Sa1_AbsF64Out = (Sa1_F64In &lt; 0.) ? (-Sa1_F64In) : Sa1_F64In;</pre>

**理由**

- MISRA C 準拠

**移行の問題** 「[ConformingFreestandingImplementation](#)」(209 ページ)を参照し  
てください。

最適化の順序を変更すると、リレーションに関するコードに違いが生じることがあります。

コード生成時間を短縮するために、TargetLink 内の最適化順序が変更されました。ゼロベースの For Iterator サブシステムを含む以下の特殊な状況では、生成されるコードがわずかに変更されます。

出発点	TargetLink ≤ 2022-B	TargetLink 2023-B
<pre>DimSize = 3; Aux_ = (Int16) (DimSize - 1); for (it = 0; it &lt;= Aux_; it++) {     ... }</pre>	<pre>for (it = 0; it &lt;= (Int16) 2; it++) {     ... }</pre>	<pre>for (it = 0; it &lt; (Int16) 3; it++) {     ... }</pre>

#### 理由

- バグ修正
- 副作用

Int タイプは、すべてのターゲットに符号付きのロングフォームを使用

タイプヘッダーは、符号付きロングフォームを使用して TargetConfig.xml から生成されます。

```
typedef signed short int t1_dsfxp_Int16; /* Description: signed 16 bit integer */
typedef signed long int t1_dsfxp_Int32; /* Description: signed 32 bit integer */
typedef signed long long int t1_dsfxp_Int64; /* Description: signed 64 bit integer */
typedef signed char t1_dsfxp_Int8; /* Description: signed 8 bit integer */
```

#### 理由

- バグ修正
- MISRA C 準拠

ベクトルまたは行列変数に対する追加の最適化

少なくとも 1 つの次元での反復を伴う変数の固定部分範囲を含むベクトル変数や行列変数へのアクセスがある場合、追加の最適化が実行されることがあります。次の例は、拡張された状況において、その後の最適化を容易にする、状態更新の列単位の再スケジューリングを示しています。

TargetLink ≤ 2022-B
<pre>for (Aux_S32 = 0; Aux_S32 &lt; 5; Aux_S32++) {     /* TargetLink Outport: TL_Root/IRead_Inport_Matrix/Out1 */     Formal[Aux_S32][0] = pISV-&gt;X_SReuse1_Unit_Delay[Aux_S32][0];     Formal[Aux_S32][1] = pISV-&gt;X_SReuse1_Unit_Delay[Aux_S32][1];      /* UnitDelay: TL_Root/IRead_Inport_Matrix/Unit Delay */     pISV-&gt;X_SReuse1_Unit_Delay[Aux_S32][0] = p_Matrix5x3[Aux_S32][0];     pISV-&gt;X_SReuse1_Unit_Delay[Aux_S32][1] = p_Matrix5x3[Aux_S32][1]; }</pre>

## TargetLink 2023-B

```

for (Aux_S32 = 0; Aux_S32 < 5; Aux_S32++)
{
    /* TargetLink Outport: TL_Root/IRead_Inport_Matrix/Out1 */
    Formal[Aux_S32][0] = pISV->X_SReuse1_Unit_Delay[Aux_S32][0];

    /* UnitDelay: TL_Root/IRead_Inport_Matrix/Unit Delay */
    pISV->X_SReuse1_Unit_Delay[Aux_S32][0] = p_Matrix5x3[Aux_S32][0];

    /* TargetLink Outport: TL_Root/IRead_Inport_Matrix/Out1 */
    Formal[Aux_S32][1] = pISV->X_SReuse1_Unit_Delay[Aux_S32][1];

    /* UnitDelay: TL_Root/IRead_Inport_Matrix/Unit Delay */
    pISV->X_SReuse1_Unit_Delay[Aux_S32][1] = p_Matrix5x3[Aux_S32][1];
}

```

## 理由

- コードの効率性
- コードサイズ

## 不要なポインタから定数へのキャストの省略

C++キャストのサポートにより、特に Classic AUTOSAR プロジェクトでは定数へのキャストが省略されるようになりました。特定のターゲットコンパイラをサポートするために、TargetLink の以前のバージョンでは標準 C では不要であるにもかかわらず、間接指示の第 1 レベル (通常はポインタから定数へ) にタイプ修飾子を追加するためにキャストを挿入してきました。

TargetLink の現在のバージョンでは、C++のキャストタイプが使用されるようになり、これらの不要なキャストは挿入されなくなりました。C コードと C++コードの構造的な等価性を保つために、C コードでのこれらのキャストも省略されます。この変更は、主として Classic AUTOSAR プロジェクトとルックアップテーブル (関数呼び出しまたはマップ構造体の初期化) に影響します。

## 例:

- ```
void myFcn(const Int16 * PointerToArray);
```

```
Int16 MyArray[5];
```
- 関数の再利用

| TargetLink ≤ 2022-B                               | TargetLink 2023-B                  |
|---------------------------------------------------|------------------------------------|
| <pre>myFcn((const Int16 *)&amp;MyArray[0]);</pre> | <pre>myFcn(&amp;MyArray[0]);</pre> |

**ポインタ整数キャストの省略** 古いバージョンの TargetLink では、変数のアドレスでポインタを初期化するときに、2 つのタイプのタイププレフィックスが異なる場合にはキャストが生成されていました。TargetLink 2023-B では、このようなキャストは生成されません。

以下の例では、マップ構造体はタイププレフィックス MyQual を持っています。z\_table もこのタイププレフィックスを持っていますが、z\_table のアドレスが割り当てられているコンポーネントは、このタイププレフィックスを持っていません。

| TargetLink ≤ 2022-B                                                                                                                                                                                                              | TargetLink 2023-B                                                                                                                                                                                                |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>MyQual MAP_Tab2DS0I2T16676_a Look_Up_Table__2D__map = {     2, /* Nx: Description: number of axis#1 points */     ...     (const Int16 *) &amp;(MyTable[0][0]) /* z_table:     Description: matrix table values */ };</pre> | <pre>MyQual MAP_Tab2DS0I2T16676_a Look_Up_Table__2D__map = {     2, /* Nx: Description: number of axis#1 points */     ...     &amp;(MyTable[0][0]) /* z_table: Description: matrix     table values */ };</pre> |

#### 理由

- 余分なコードの抑制
- 可読性
- MISRA C++ガイドライン
- 整合性の向上

#### キャストでの TargetLink ベース タイプの使用

Int16 などの TargetLink ベースタイプへのキャストがより一貫して適用されるようになりました。

| TargetLink ≤ 2022-B                                                                | TargetLink 2023-B                                                              |
|------------------------------------------------------------------------------------|--------------------------------------------------------------------------------|
| <pre>SetAbsAlarm(MyAlarm, (TickType) TL_FX_GROUND, (TickType) TL_FX_GROUND);</pre> | <pre>SetAbsAlarm(MyAlarm, (UInt32) TL_FX_GROUND, (UInt32) TL_FX_GROUND);</pre> |

入力が 2 つより多いスカラー積を計算するブロックは、次のようにキャストを変更できません。

| TargetLink ≤ 2022-B                                                                                                                                     | TargetLink 2023-B                                                                                                                                       |
|---------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>X_Sa11_DiscrFilter[0] += ((THETA) (((THETA) (((Int32) Sa11_Input) + ((Int32) X_Sa11_DiscrFilter[1])) &gt;&gt; 1)) - X_Sa11_DiscrFilter[0]));</pre> | <pre>X_Sa11_DiscrFilter[0] += ((Int16) (((Int16) (((Int32) Sa11_Input) + ((Int32) X_Sa11_DiscrFilter[1])) &gt;&gt; 1)) - X_Sa11_DiscrFilter[0]));</pre> |

#### 理由

- バグ修正
- 整合性の向上

#### 移行の問題

#### 外部サブシステムおよびインクリメンタルコード生成用に設定されたサブシステムに対する最適化の変更

TargetLink は、外部サブシステムおよびインクリメンタルコード生成用に設定されたサブシステムの内容を把握することはできず、未知または可変であると想定しています。つまり、TargetLink は、呼び出されるたびに出力変数が完全に記述されているかどうかを判断することができません。

TargetLink 2023-B 以降、TargetLink では、外部サブシステムおよびインクリメンタルコード生成用に設定されたサブシステムの出力パラメータは、ステップ関数クラスが Optimization プロパティに SIDE\_EFFECT\_FREE を含む Function ブロックから参照されない限り、最適化中に条件付きまたは不完全に記述されたものと見なされます。その結果、これらのサブシステムの出力に影響する最適化は実行されない可能性があります。

**理由**

- バグ修正
- 整合性の向上

**tlnomath.h からのインクルードを削除**

TargetLink は、標準 C ライブラリには含まれていない数学関数 `hypot()` を計算するために、`Dsfxp` ライブラリで独自の実装を提供しています。この実装では、`sqrt()` 関数を呼び出すために、`math.h` をインクルードします。`tlnomath.h` のインクルードを削除するのはこのためです。

| TargetLink ≤ 2022-B                                          | TargetLink 2023-B                   |
|--------------------------------------------------------------|-------------------------------------|
| <pre>#include "tlnomath.h"... Out = C_HYPOT(In1, In2);</pre> | <pre>Out = C_HYPOT(In1, In2);</pre> |

**理由**

- バグ修正
- 副作用

**3-D、4-D、5-D ルックアップ関数呼び出しでの変数名と順序の改善**

3-D、4-D、5-D ルックアップ関数呼び出しにおける暗黙変数 `Offset`、`Fraction`、`LinIdx` の命名テンプレートが `$S_$B_X` に変更されました。ルックアップ関数呼び出しのこれらのパラメータの準備はグループ化され、ブロックへのモデルパスを示すコメントが付けられます。例:

```
/* Interpolation: Path/To/Interpolation Using PreLookup */
<Offset vector Assignments>
<Fraction vector Assignments>
<Computation of LinIdx>
```

さらに、生成された 3-D、4-D、5-D ルックアップ関数について、以下のように説明的なコメントが追加されます。

- テーブルアクセスについて: Gather all values
- 補間について: Interpolation along dimension X.

**理由**

- AUTOSAR 準拠
- バグ修正
- ユーザ経験と一致
- 整合性の向上
- 可読性

**ap\_Entry および ap\_Exit 呼び出しの識別子の変更**

TargetLink は、呼び出し関数、制御フロー内のネスト深度、およびインライン化の順序に応じて、`ap_Entry` および `ap_Exit` 呼び出しの識別子を自動的に割り当てます。TargetLink のバージョン間で同じコード生成の識別子が非決定的になる問題を解決したため、個々のケースで識別子の入れ替えが発生する可能性があります。

| TargetLink ≤ 2022-B                                                                                                                                                                                                                                                          | TargetLink 2023-B                                                                                                                                                                                                                                                            |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> if (Cond1) {   ap_Entry_SID1_Name_1_101();   ...   ap_GetVector_In(Aux_Vector_In_ID);   ...   ap_Exit_SID1_Name_1_101(); } ...  if (Cond2) {   ap_Entry_SID1_Name_1_100();   ...   ap_GetScalar_In(&amp;Aux_Scalar_In_ID);   ...   ap_Exit_SID1_Name_1_100(); } </pre> | <pre> if (Cond1) {   ap_Entry_SID1_Name_1_100();   ...   ap_GetVector_In(Aux_Vector_In_ID);   ...   ap_Exit_SID1_Name_1_100(); } ...  if (Cond2) {   ap_Entry_SID1_Name_1_101();   ...   ap_GetScalar_In(&amp;Aux_Scalar_In_ID);   ...   ap_Exit_SID1_Name_1_101(); } </pre> |

**理由**

- バグ修正
- コードの安定性

**異なるループ変数タイプによるループマージ**

異なるループ変数タイプでのループマージが以下のように変更されました。

- ループ変数がタイプが異なるループ変数に置き換えられる場合、ループ変数へのアクセスの前にキャストが挿入される場合があります。
- アドレスアクセスを持つループ変数は、他のタイプのループ変数とスワップされなくなりました。その結果、ループのマージやその他のループ変数が少なくなる可能性があります。
- ループ変数は符号が異なるループ変数とは交換されません。

**理由**

- バグ修正

**enum マクロ定義と enum 定義の間に空行がある**

マクロ実装を持つ enum の最後のマクロ定義の後に、enum タイプ定義などの複数行のステートメントが続く場合、空行が出力されるようになりました。

| TargetLink ≤ 2022-B                                                                                                                                                                                                                                                                                        | TargetLink 2023-B                                                                                                                                                                                                                                                                                           |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> typedef Int32 MyBasicColors2; #define Black ((MyBasicColors2) 0) #define Pink ((MyBasicColors2) 1) #define Brown ((MyBasicColors2) 2) #define Magenta ((MyBasicColors2) 3) typedef enum MyBasicColors1_tag {   Red = 0,   Yellow = 1,   Green = 2,   Blue = 3,   Orange = 4 } MyBasicColors1; </pre> | <pre> typedef Int32 MyBasicColors2; #define Black ((MyBasicColors2) 0) #define Pink ((MyBasicColors2) 1) #define Brown ((MyBasicColors2) 2) #define Magenta ((MyBasicColors2) 3)  typedef enum MyBasicColors1_tag {   Red = 0,   Yellow = 1,   Green = 2,   Blue = 3,   Orange = 4 } MyBasicColors1; </pre> |

**理由**

- 可読性

プリプロセッサステートメントのコメント      プリプロセッサステートメント内のコメントは、通常の C コメントと同様にバックスラッシュを使わずに囲むようになりました。

#### TargetLink ≤ 2022-B

```
#define SetVarS16(_var, _value) (_var = _value) /* Use SetVarTp(<Name>) macro to write a scalar\
variable */
```

#### TargetLink 2023-B

```
#define SetVarS16(_var, _value) (_var = _value) /* Use SetVarTp(<Name>) macro to write a scalar vari\
able */
```

#### 理由

- バグ修正

#### 関連トピック

#### 参考文献

Function ブロック (『TargetLink モデルエレメントリファレンス』)

## TargetLink 2023-B での廃止項目

#### 次のステップ

#### 本章の内容

|                           |     |
|---------------------------|-----|
| 廃止された TargetLink の機能..... | 247 |
| 廃止された制限事項.....            | 248 |
| 廃止されたコードジェネレータオプション.....  | 249 |

## 廃止された TargetLink の機能

#### Microsoft Express Edition コンパイラの廃止

Microsoft Express Edition コンパイラはサポートされなくなりました。その代わりに、Microsoft Visual Studio C++ (MSVC) 製品ファミリの 2015 年現在のすべてのエディション (Community、Professional、Enterprise) を MEX コンパイラとして、また SIL シミュレーションモードでのデバッグ用に使用することができます。SIL シミュレーションモードでのデバッグは前述のコンパイラでのみ可能であるため、`tlProductionCodeSILCompiler` API コマンドは不要になりました。SIL デバッグの詳細については、「SIL シミュレーションモードでデバッグする方法」(『TargetLink 準備およびシミュレーションガイド』)を参照してください。

## 廃止された制限事項

### ファンクション入力の const 修飾子

TargetLink では、ファンクションパラメータとして実装された(つまり、次のいずれかの方法で指定された)ファンクション入力の const 修飾をサポートしていません。

- Scope プロパティが value\_param または ref\_param に設定された DD VariableClass オブジェクトを参照する変数を持つ TargetLink InPort ブロック。
- 変数が明示的な構造体として指定され、Scope プロパティが ref\_param に設定された DD VariableClass オブジェクトを参照している TargetLink Bus InPort ブロック。

ブロック出力変数の const 修飾は、次の場合のみ可能です。

- TargetLink Constant ブロックの出力変数の場合
- 間接的な再利用に指定された TargetLink InPort の場合(つまり、InPort ブロックが次のように指定された DD VariableClass オブジェクトを参照している場合)

| プロパティ                        | 値                |
|------------------------------|------------------|
| Scope                        | ref_param        |
| FunctionReuse                | InstanceSpecific |
| FunctionReuseAccessByPointer | On               |

### Adaptive AUTOSAR とコード生成に関する制限事項

次の両方の条件に該当する場合、TargetLink では、Function ブロックの[Role]が Adaptive AUTOSAR Function に設定されているサブシステムを含むモデルでのコード生成をサポートしていません。

- このサブシステムが TargetLink サブシステムの下で直接モデル化されている。
- このサブシステムに関連付けられた DD AdaptiveAutosarFunction ブロックが、DD ModuleTemplate オブジェクトの影響を受ける DD Module オブジェクトを参照している。

### ポインタの戻り値

ポインタタイプの戻り値を持つ関数呼び出しはサポートされていません。

### ConformingFreestandingImplementation オプションの矛盾する動作

このオプションをアクティブにすると、生成されたコードの標準的な math.h 関数が回避され、代わりに TargetLink 固有のインプリメンテーションが使用されます。ただし、TargetLink 固有のインプリメンテーションは fabs 関数のみに存在します。



## 廃止されたコードジェネレータオプション

| 削除されたコードジェネレータオプション                  | 説明                                                                                                                                                                                                                                                                                                                                                                                     |
|--------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ConformingFreestandingImplementation | <p>コードジェネレータオプションは、以下の新しいコードジェネレータオプションに分割されました。</p> <ul style="list-style-type: none"><li>▪ <code>EmitWarningIfMathHeaderIsIncluded</code><br/>C 標準ライブラリヘッダー <code>math.h</code> が含まれている場合に警告を発するかどうかを指定します。</li><li>▪ <code>UseTLImplementationForFabsFunction</code><br/>C 標準ライブラリヘッダー <code>math.h</code> の <code>fabs</code> 関数を、?演算子を使用した条件式として量産コードに直接実装するかどうかを指定します。</li></ul> |

### 関連トピック

### 参考文献

[EmitWarningIfMathHeaderIsIncluded](#) (『TargetLink モデルエレメントリファレンス』)  
[UseTLImplementationForFabsFunction](#) (『TargetLink モデルエレメントリファレンス』)

## 今後の TargetLink バージョンでの変更予定

### 次のステップ

### 本章の内容

|                          |     |
|--------------------------|-----|
| 廃止予定の機能.....             | 250 |
| 廃止予定の API 関数.....        | 251 |
| 廃止されたコードジェネレータオプション..... | 251 |

## 廃止予定の機能

|                                                                        |                                                                                                                                                                                                                                                                     |
|------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RTOS/OSEK コード生成モード                                                     | TargetLink の RTOS/OSEK コード生成モードのサポートは、今後の TargetLink バージョンで廃止される予定です。                                                                                                                                                                                               |
| ターゲット最適化モジュール (TOM)                                                    | TargetLink のターゲット最適化モジュール (TOM) のサポートは、今後の TargetLink バージョンで廃止される予定です。                                                                                                                                                                                              |
| Simulink の従来の初期化モード                                                    | Simulink の従来の初期化モードのサポートは、今後の TargetLink バージョンで廃止される予定です。                                                                                                                                                                                                           |
| 動的コンポーネント                                                              | DD Variable オブジェクトに対する動的コンポーネントの指定は、今後の TargetLink バージョンでサポートが廃止される予定です。                                                                                                                                                                                            |
| SWC ReceiverPort および SWC SenderPort ブロック (TargetLink AUTOSAR Blockset) | SWC ReceiverPort および SWC SenderPort ブロックのサポートは、今後の TargetLink バージョンで廃止される予定です。代わりに、InPort および OutPort ブロックを使用することができます。                                                                                                                                             |
| 関連トピック                                                                 | 参考文献                                                                                                                                                                                                                                                                |
|                                                                        | <a href="#">InPort ブロック (『TargetLink モデルエレメントリファレンス』)</a><br><a href="#">OutPort ブロック (『TargetLink モデルエレメントリファレンス』)</a><br><a href="#">SWC ReceiverPort ブロック (『TargetLink モデルエレメントリファレンス』)</a><br><a href="#">SWC SenderPort ブロック (『TargetLink モデルエレメントリファレンス』)</a> |

## 廃止予定の API 関数

### 廃止された API 関数

次の API 関数は廃止され、今後の TargetLink バージョンで削除される予定です。

| 関数                       | 廃止されるバージョン     | 代替りの関数 |
|--------------------------|----------------|--------|
| tl_compare_fcn_signature | TargetLink 5.0 | -      |

## 廃止されたコードジェネレータオプション

### 廃止されたコードジェネレータオプション

次のコードジェネレータオプションは廃止され、今後の TargetLink バージョンで削除される予定です。

- 「AllowStructAssignments」(『TargetLink モデルエレメントリファレンス』[📖](#))
- 「DeriveDefaultSymbolClasses」(『TargetLink モデルエレメントリファレンス』[📖](#))
- 「ExploitRangesIndependentOfErasable」(『TargetLink モデルエレメントリファレンス』[📖](#))
- 「HandleUnscaledStateflowExpressionsWithTIType」(『TargetLink モデルエレメントリファレンス』[📖](#))
- 「StrictTypedefPlacement」(『TargetLink モデルエレメントリファレンス』[📖](#))
- 「UseRootFileNameAsUserDefinedTypesHeaderBasename」(『TargetLink モデルエレメントリファレンス』[📖](#))
- 「UtilizeValueEqualityStructFunctionArguments」(『TargetLink モデルエレメントリファレンス』[📖](#))



# VEOS

## 次のステップ

## 本章の内容

|                                                          |     |
|----------------------------------------------------------|-----|
| VEOS 2023-B の新機能.....                                    | 253 |
| VEOS 2023-B の新機能について概要を説明します。                            |     |
| VEOS 2023-B の互換性.....                                    | 255 |
| VEOS 2023-B の互換性に関する情報を提供します。                            |     |
| VEOS 2023-B への移行.....                                    | 259 |
| VEOS 2023-A から VEOS 2023-B に移行するには、特定の移行手順が必要になる場合があります。 |     |
| VEOS 2024-A での廃止予定項目.....                                | 260 |
| VEOS 2024-A で予定される廃止項目について記載します。                         |     |

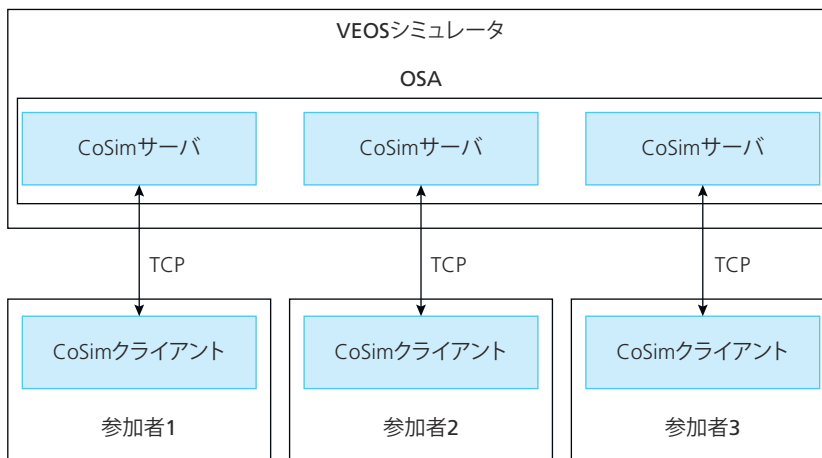
## VEOS 2023-B の新機能

### VEOS を使用した協調シミュレーション

dSPACE は、VEOS を使用した時間同期の協調シミュレーションを実装するための新しいインターフェースを提供しています。これにより、VEOS をデータ再生ツールやセンサ信号シミュレータなどのドメイン固有のシミュレータと組み合わせることができます。協調シミュレーションインターフェースは、シミュレートされた CAN、LIN、Ethernet バスメッセージや I/O 信号の交換を可能にします。

VEOS を使用した協調シミュレーションは、クライアント/サーバ アーキテクチャで実装されます。この場合、VEOS シミュレータが協調シミュレーションのサーバとなり、すべての協調シミュレーション参加者にシミュレーション時間全体を提供します。このインターフェースは、汎用的な C ライブラリとして実装されています。

次の図を参照してください。



この機能には、adaptive V-ECU license(「[VEOS モジュール](#)」(『[VEOS 概要](#)』[📄](#)))を参照)が必要です。

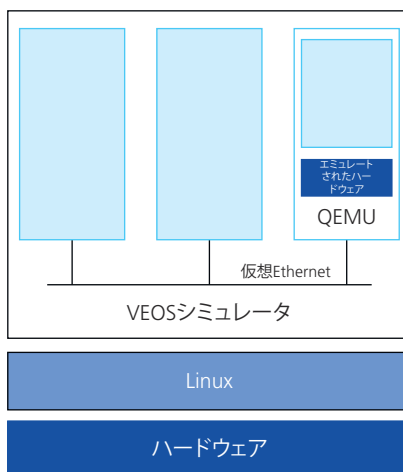
詳細については、「[VEOS を使用する協調シミュレーションの基礎](#)」(『[VEOS マニュアル](#)』[📄](#))を参照してください。

## VEOS シミュレーションへの QEMU の統合

VEOS シミュレーションに QEMU システムエミュレーションを統合できるようになりました。ARM および x86 ハードウェアアーキテクチャのエミュレーションをサポートしており、VEOS 用にビルドされていないバイナリを実行することができます。

QEMU システムエミュレーション中、VEOS は QEMU 仮想時間を制御します。QEMU の統合は、現在、VEOS on Linux でサポートされています。

次の図を参照してください。



この機能には、adaptive V-ECU license(「[VEOS モジュール](#)」(『[VEOS 概要](#)』[📄](#)))を参照)が必要です。

詳細については、「[VEOS での QEMU 統合の基礎](#)」(『[VEOS マニュアル](#)』[📄](#))を参照してください。

### VEOS Simulator Console の機能強化

VEOS Simulator Console には、以下のいくつかの機能強化があります。

- info コマンド:  
このコマンドは、実際の経過時間と加速係数を表示するようになりました。
- start および step コマンド:
  - シミュレーション時間を指定できるようになりました。例:  
`veos-sim start 30s`: シミュレーションは、start コマンドが呼び出された後 30 秒間実行されます。継続時間は、PC の時間ではなく、仮想シミュレーションの時間を指します。
  - さらに、新しい `--blocking` オプションにより、指定したシミュレーション時間が経過するまで `veos-sim` が戻らないように指定することができます。例:  
`veos-sim step 20s --blocking`: シミュレーションは、step コマンドが呼び出された後 20 秒間段階的に実行されます。その後、VEOS Simulator Console に戻ります。

「[VEOS Simulator Console コマンドリファレンス](#)」(『[VEOS マニュアル](#)』[📖](#))を参照してください。

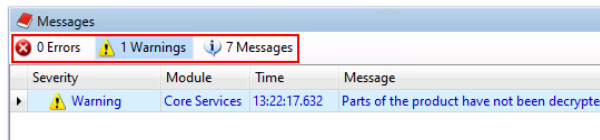
### VEOS Linux カーネル再ビルド用 Dev コンテナファイル

Visual Studio Code の Dev Containers 拡張機能を使用するだけで、GPU パススルーデモ用の VEOS Linux カーネルを再ビルドできる Dev Container 設定ファイルが追加されました。

「[流体シミュレーションを実行するホスト PC を準備する方法](#)」(『[VEOS マニュアル](#)』[📖](#))を参照してください。

### VEOS Player: [Messages]ペインのフィルタ設定が自動保存されます。

VEOS Player の Messages(「[\[Messages\]ペイン](#)」(『[VEOS マニュアル](#)』[📖](#))を参照)ペインのフィルタ設定が自動保存され、次回 VEOS Player を起動したときに再適用されるようになりました。



## VEOS 2023-B の互換性

### 全般的な互換性

dSPACE では、同一の dSPACE Release のソフトウェア製品のみを使用することをお勧めしています。これにより、最大限のランタイム互換性が保証されます。

### サポートされるコンパイラのバージョン

サポートされるコンパイラバージョンの詳細については「[シミュレーションシステムの統合の基礎](#)」(『[VEOS シミュレーションシステムの統合](#)』[☞](#))を参照してください。

#### 注記

モデルコンテナを個別のステップで作成する場合、たとえば、既存のモデルコンテナを dSPACE モデルコンテナユーティリティで事前にコンパイルする場合は、[互換性のあるバージョンのコンパイラとリンク](#)を使用することをお勧めします。これにより、ビルド結果のバイナリ互換性が確保されます。

詳細については、「[モデルコンテナ構築時のバイナリ互換性の確保](#)」(『[VEOS マニュアル](#)』[☞](#))を参照してください。

### サポートしているオペレーティングシステム

VEOS は、Windows と Linux で利用することができます。サポートされている配布については、「[オペレーティングシステム](#)」(263 ページ)を参照してください。

**VEOS on Linux** VEOS on Linux については、「[VEOS 上の Linux の概要](#)」(『[VEOS マニュアル](#)』[☞](#))を参照してください。

**VEOS の仮想化** クラウド/クラスタ環境での VEOS の仮想化については、「[VEOS の仮想化](#)」(『[VEOS マニュアル](#)』[☞](#))を参照してください。

### BSC ファイルの互換性

次の表は、VEOS 2023-B とバスシミュレーションコンテナ (BSC) ファイルの互換性を示しています。

| BSC ファイルを作成した製品 ...                                                                                                                                                                          | BSC のバージョン |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| <ul style="list-style-type: none"> <li>▪ Bus Manager 2023-B</li> <li>▪ Bus Manager 2023-A</li> </ul>                                                                                         | 1.12       |
| <ul style="list-style-type: none"> <li>▪ Bus Manager 2022-B</li> <li>▪ Ethernet Configuration Package 4.5<sup>1)</sup></li> <li>▪ Ethernet Configuration Package 4.4<sup>1)</sup></li> </ul> | 1.11       |

<sup>1)</sup> Ethernet Configuration Package に関する詳細情報および最新の互換性情報については、<http://www.dspace.com/go/ethcpinfo> を参照してください。

▪ BSC ファイルを *SIC* ファイルを使用して生成した場合、*SIC* ファイル生成用に選択したターゲットアーキテクチャで、VEOS ビルドプロセスで利用可能なシミュレーションターゲットを指定します。

「[Simulink インプリメンテーションコンテナの基礎](#)」(『[Model Interface Package for Simulink - モデリングガイド](#)』[☞](#))を参照してください。

▪ BSC ファイルを *SIC* ファイルを使用しないで生成した場合、Simulation Target Manager でサポートされる任意のシミュレーションターゲットを選択することができます。

### FMU ファイルの互換性

VEOS では、次の FMI 規格バージョンに準拠した Functional Mock-Up Units (FMU) をサポートしています。

- FMI 2.0
- FMI 3.0

FMI 3.0 リリースの詳細については、<https://fmi-standard.org/news/2022/05/10/fmi-3.0-release.html> を参照してください。



dSPACE FMI サポートに関する詳細情報および最新の互換性情報については、次の Web サイトを参照してください。

<http://www.dspace.jp/go/FMI-Compatibility>

### OSA ファイルの互換性

次の表は、VEOS 2023-B とオフラインシミュレーションアプリケーション(OSA)ファイルの互換性を示しています。

| OSA ファイルを作成した製品 ...              | OSA のバージョン         |
|----------------------------------|--------------------|
| VEOS 2023-B                      | 23.2               |
| VEOS 2023-A                      | 23.1 <sup>1)</sup> |
| VEOS 2022-B                      | 22.2 <sup>1)</sup> |
| VEOS 5.4 (dSPACE Release 2022-A) | 5.4 <sup>1)</sup>  |

<sup>1)</sup> OSA ファイルを最初に作成したバージョンよりも後の VEOS バージョンで OSA ファイルを開く場合、OSA ファイルに含まれる VPU のプロパティを変更することはできません。ただし、ポートやネットワーク接続の編集と試験サービスの有効化/無効化は行うことができます。VEOS バージョンを別のバージョンに移行する場合、既存のモデルインプリメンテーションコンテナファイル (BSC、FMU、SIC、VECU) のバイナリ OSA ファイルを再ビルドすることをお勧めします。

### SIC ファイルの互換性

次の表は、VEOS 2023-B と Simulink インプリメンテーションコンテナ(SIC)ファイルの互換性を示しています。

| SIC ファイルを作成した製品 ...                                                                                                                                                                 | SIC のバージョン <sup>1)</sup> |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------|
| <ul style="list-style-type: none"> <li>▪ Model Interface Package for Simulink 2023-B</li> <li>▪ Model Interface Package for Simulink 2023-A</li> <li>▪ TargetLink 2023-B</li> </ul> | 1.12                     |
| <ul style="list-style-type: none"> <li>▪ Model Interface Package for Simulink 2022-B</li> <li>▪ TargetLink 2022-B</li> </ul>                                                        | 1.11 <sup>2)</sup>       |
| <ul style="list-style-type: none"> <li>▪ Model Interface Package for Simulink 22.1</li> <li>▪ Model Interface Package for Simulink 4.6</li> <li>▪ TargetLink 5.2</li> </ul>         | 1.10 <sup>2)</sup>       |

<sup>1)</sup> SIC ファイル生成用に選択したターゲットアーキテクチャで、VEOS ビルドプロセスで利用可能なシミュレーションターゲットを指定します。「[Simulink インプリメンテーションコンテナの基礎](#)」(『[Model Interface Package for Simulink - モデリングガイド](#)』[☞](#))を参照してください。

<sup>2)</sup> SIC ファイルが以前の dSPACE Release で作成され、SIC ファイルに ASM モデルが含まれている場合、VEOS 2023-B (dSPACE Release 2023-B) でモデルをシミュレーションすることはできません。詳細については、「[ASM モデルの移行](#)」(『[VEOS 新しい機能と移行手順](#)』[☞](#))を参照してください。

### SMC ファイルの互換性

次の表は、VEOS 2023-B とシステムモデルコンテナ(SMC)ファイルの互換性を示しています。

| SMC ファイルを作成した製品 ...                                                                                                                                       | SMC のバージョン |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| <ul style="list-style-type: none"> <li>▪ VEOS 2023-B</li> <li>▪ VEOS 2023-A</li> <li>▪ VEOS 2022-B</li> <li>▪ VEOS 5.4 (dSPACE Release 2022-A)</li> </ul> | 1.2        |

また、インポートする SMC ファイルに含まれる個々のコンテナファイルに関する次の互換性の制約を考慮する必要があります。SMC ファイルに、サポートされていないバージョンのコンテナが含まれている場合、VEOS 2023-B はサポートされていないコンテナも、サポートされていないコンテナに基づいたアプリケーションプロセスへの接続もインポートしません。

### VECU ファイルの互換性

次の表は、VEOS 2023-B と V-ECU インプリメンテーションコンテナ (VECU) ファイルの互換性を示しています。

| V-ECU インプリメンテーションの作成に使用した製品                                                                                                                  | V-ECU インプリメンテーションのバージョン |
|----------------------------------------------------------------------------------------------------------------------------------------------|-------------------------|
| <ul style="list-style-type: none"> <li>▪ SystemDesk 2023-B</li> <li>▪ TargetLink 2023-B</li> </ul>                                           | 3.3                     |
| <ul style="list-style-type: none"> <li>▪ SystemDesk 2022-B</li> <li>▪ TargetLink 2022-B</li> </ul>                                           | 3.2                     |
| <ul style="list-style-type: none"> <li>▪ SystemDesk 5.6 (dSPACE Release 2021-B)</li> <li>▪ TargetLink 5.2 (dSPACE Release 2021-B)</li> </ul> | 3.1                     |

### Real-Time Testing との互換性

VEOS と ControlDesk で RTT を使用する場合は、シミュレーションシステムを実行する VEOS シミュレータで使用する Real-Time Testing (RTT) バージョンと、PC でアクティブな RTT バージョンが同じである必要があります。

VEOS シミュレータのバージョンと対応する RTT のバージョンを次の表に示します。

| VEOS シミュレータ | RTT のバージョン                       |
|-------------|----------------------------------|
| VEOS 2023-B | Real-Time Testing Version 2023-B |
| VEOS 2023-B | Real-Time Testing Version 2023-A |
| VEOS 2022-B | Real-Time Testing Version 2022-B |
| VEOS 5.4    | Real-Time Testing Version 5.2    |
| VEOS 5.3    | Real-Time Testing Version 5.1    |
| VEOS 5.2    | Real-Time Testing Version 5.0    |
| VEOS 5.1    | Real-Time Testing Version 4.4    |
| VEOS 5.0    | Real-Time Testing Version 4.3    |

ControlDesk 2023-B は、自動的に VEOS 2023-B の VEOS シミュレータを使用します。したがって、PC 上で Real-Time Testing Version 2023-A がアクティブであれば、VEOS と ControlDesk で RTT を使用することができます。

### Python との互換性

VEOS 2023-B の VEOS Player の自動化インターフェースは、Python 3.11 に対応しています。

## 関連トピック

## 基本概要

Windows でのアダプティブ V-ECU シミュレーションのためのハイパーバイザの設定  
 (『VEOS とアダプティブ V-ECUs の連携』[📄](#))  
 モデルコンテナ構築時のバイナリ互換性の確保 (『VEOS マニュアル』[📄](#))

## VEOS 2023-B への移行

## 概要

VEOS 2023-A から VEOS 2023-B に移行するには、特定の移行手順が必要になる場合があります。

## 注記

2023-A より前のバージョンから VEOS 2023-B に移行するには、その間の VEOS バージョンの移行手順も必要になる場合があります。

## Python 3.11 への移行

dSPACE Release 2023-B 以降、dSPACE 製品は Python 3.9 をサポートしなくなりました。その代わりに、Python 3.11 がサポートされます。

VEOS 固有の移行上の注意点はありません。

詳細については、「[Python 3.9 から Python 3.11 への Python スクリプトの移行](#)」(31 ページ)を参照してください。

## SIC コンテナのポート名の変更

Model Interface Package for Simulink 2023-B は、多数の Simulink®ポートタイプのサポートが強化されています。Simulink インプリメンテーションコンテナ(SIC)ファイルを生成する場合、SIC ファイル内のルートレベルの Simulink ポートの名前は、Model Interface Package for Simulink の以前のバージョンで生成された名前とは異なります。

その結果、オフラインシミュレーションアプリケーションに、Model Interface Package for Simulink 2023-A 以前のバージョンで作成された Simulink インプリメンテーションコンテナ(SIC)ファイルに基づく VPU が含まれている場合に、Model Interface Package for Simulink 2023-B 以降のバージョンで SIC ファイルを再作成して VEOS に再インポートすると、インポート時に警告メッセージが表示され、関連する VPU のポート接続が失われます。

VEOS でこれらのポートを再接続する必要があります。

詳細については、「[Model Interface Package for Simulink 2023-B への移行](#)」(144 ページ)を参照してください。

## ASM モデルの移行

VEOS インストレーションが属する dSPACE Release よりも以前の dSPACE Release で作成された OSA または SIC ファイルにモデルが含まれる場合、ASM モデルを VEOS でシミュレートすることはできません。

VEOS のバージョンが属する dSPACE Release より前の dSPACE Release で保存された ASM モデルをシミュレートするには、次の手順を実行します。

1. ASM モデルを VEOS のバージョンが属する dSPACE Release に移行します。  
ASM モデルの移行については、「[ASM モデルの移行](#)」(『[ASM ユーザガイド](#)』[📖](#))を参照してください。
2. *Model Interface Package for Simulink* を使用して、ASM モデルに基づく Simulink インプリメンテーションコンテナ (SIC) ファイルを生成します。  
詳細については、「[Simulink インプリメンテーションコンテナの生成](#)」(『[Model Interface Package for Simulink - モデリングガイド](#)』[📖](#))を参照してください。
3. SIC ファイルをご使用の VEOS バージョンの VEOS Player にインポートします。  
詳細については、「[Simulink インプリメンテーションコンテナ \(SIC\) の統合の基礎](#)」(『[VEOS シミュレーションシステムの統合](#)』[📖](#))を参照してください。

#### 以前の VEOS バージョンからの移行

以前の VEOS バージョンから移行して既存のオフラインシミュレーションアプリケーションを再利用するには、追加の移行手順が必要な場合があります。移行手順の詳細については、「[以前のバージョンの VEOS からの移行](#)」(『[VEOS マニュアル](#)』[📖](#))を参照してください。

## VEOS 2024-A での廃止予定項目

#### クロスコンパイルのサポートはありません

dSPACE Release 2024-A では、VEOS on Linux 用のオフラインシミュレーションアプリケーションを Windows 上でビルドすることができなくなりました。これは、VEOS Player と VEOS Build Console に適用されます。

VEOS on Linux 用のオフラインシミュレーションアプリケーションをビルドするには、代わりに VEOS Build Console on Linux を使用します。「[オフラインシミュレーションアプリケーションをビルドする方法](#)」(『[VEOS シミュレーションシステムの統合](#)』[📖](#))を参照してください。

#### Microsoft Hyper-V を介して Windows 上でアダプティブ V-ECU のシミュレーションはできません

dSPACE Release 2024-A では、Microsoft Hyper-V を介した Windows 上でのアダプティブ V-ECU シミュレーションはサポートされなくなりました。

別の方法として、QEMU エミュレーションでアダプティブ V-ECU を実行し、QEMU エミュレーションを VEOS シミュレーションに統合することもできます。

詳細については、「[VEOS での QEMU 統合の基礎](#)」(『[VEOSQEMU を VEOS に統合する](#)』[📖](#))を参照してください。

# 互換性情報

## 次のステップ

## 本章の内容

|                                     |     |
|-------------------------------------|-----|
| サポートしている MATLAB リリース.....           | 261 |
| オペレーティングシステム.....                   | 263 |
| 仮想マシン (VM) での dSPACE ソフトウェアの使用..... | 265 |
| dSPACE ソフトウェアのランタイム互換性.....         | 270 |
| Windows 機能の使用に関する制限事項.....          | 270 |
| Linux 機能の使用に関する制限事項.....            | 272 |

## サポートしている MATLAB リリース

### MATLAB®/Simulink®

各種 dSPACE 製品を使用するには、MATLAB をホスト PC にインストールしておく必要があります。

#### ヒント

MathWorks®社製ソフトウェアのシステム要件については、<http://www.mathworks.com/support/sysreq.html> を参照してください。

| MATLAB のリリース | dSPACE Release 2023-B の各コンポーネントによるサポート |                                        |                                     |                   |                      |                                               |                            |
|--------------|----------------------------------------|----------------------------------------|-------------------------------------|-------------------|----------------------|-----------------------------------------------|----------------------------|
|              | RCP and HIL 2023-B <sup>1), 2)</sup>   | ConfigurationDesk 2023-B <sup>3)</sup> | AutomationDesk 2023-B <sup>4)</sup> | TargetLink 2023-B | Model Compare 2023-B | dSPACE Python Extensions 2023-B <sup>5)</sup> | XIL API .NET MAPort 2023-B |
| R2023b       | ✓ <sup>6)</sup>                        | ✓                                      | ✓                                   | — <sup>7)</sup>   | ✓                    | ✓                                             | ✓                          |
| R2023a       | ✓ <sup>6)</sup>                        | ✓                                      | ✓                                   | — <sup>7)</sup>   | ✓                    | ✓                                             | ✓                          |
| R2022b       | ✓ <sup>6)</sup>                        | ✓                                      | ✓                                   | ✓                 | ✓                    | ✓                                             | ✓                          |
| R2022a       | ✓                                      | ✓                                      | ✓                                   | ✓                 | ✓                    | ✓                                             | ✓                          |

- 1) 'RCP and HIL Software'は、ASM、Model Interface Package for Simulink、RTI、ModelDesk などのさまざまな dSPACE ソフトウェア製品が含まれるソフトウェアパッケージを指す総称です。これらのソフトウェア製品は、共通のフォルダへインストールされます。dSPACE Release 2023-A では、ConfigurationDesk は RCP and HIL ソフトウェアパッケージに含まれなくなりました。専用のフォルダにインストールされます。
- 2) MATLAB/Simulink Student Suite は Automotive Simulation Models (ASM) によってサポートされていません。
- 3) ConfigurationDesk は、MATLAB への接続と Simulink モデルを扱うために、Model Interface Package for Simulink (RCP and HIL ソフトウェアパッケージの一部) が必要です。したがって、RCP and HIL ソフトウェアをインストールすることが、ConfigurationDesk での MATLAB サポートの前提条件となります。
- 4) AutomationDesk の MATLAB Access ライブラリには MATLAB が必要です。
- 5) dSPACE Python Extensions の matlplotlib2 には MATLAB が必要です。
- 6) R2023b、R2023a、R2022b は、FPGA Programming Blockset 23.2 – FPGA Interface ではサポートされません。対応策については、<http://www.dspace.com/faq?295> を参照してください。
- 7) サポートされていません。TargetLink 23.1 パッチ 1 のインストール後はフルサポートです。ダウンロードについては、<http://www.dspace.jp/go/PatchesTL> を参照してください。

dSPACE ソフトウェアと組み合わせて使用可能なその他の MATLAB Release の最新情報については、<http://www.dspace.jp/go/MATLABCompatibility> を参照してください。

## オペレーティングシステム

### ホスト PC の Windows オペレーティングシステム

dSPACE Release 2023-B の dSPACE 製品は、以下のオペレーティングシステムをサポートしています。

- Windows 10 の、次のエディション、チャンネルおよびサービスオプションがサポートされます。
  - Windows 10 Professional、Education、Enterprise (64 ビット版)  
Windows 10 Home、および Windows 10 S エディションはサポートされません。
  - Long-Term Servicing Channel: LTSC 2019 および LTSC 2021
  - General Availability Channel (以前の Semi-Annual Channel を置き換えたもの): Microsoft の互換性に関する記述が適用されます。このチャンネルでリリースされた新しいバージョンは、以前のすべてのバージョンに準拠していることとなります。dSPACE は、General Availability Channel の 22H2 バージョンをテストに使用しました。
- Windows 11 の、次のエディション、チャンネルおよびサービスオプションがサポートされます。
  - Windows 11 Professional、Education、Enterprise  
Windows 11 Home エディションはサポートされません。
  - General Availability Channel: Microsoft の互換性に関する記述が適用されます。このチャンネルでリリースされた新しいバージョンは、以前のすべてのバージョンに準拠していることとなります。dSPACE は、General Availability Channel の 22H2 バージョンをテストに使用しました。

#### 注記

Windows 11 22H2 で当社の製品をテストしたところ、いくつかの問題が発生しましたが、その原因や対処法については dSPACE の責任ではありません。詳細については、<http://www.dspace.jp/go/Win11GAC22H2> を参照してください。

- Windows Server LTSC 2019 Standard および Datacenter エディション、各エディションにデスクトップエクスペリエンスインストールオプション付き: 上記のエディションのみサポートされます。Windows Server 2019 Essentials エディションはサポートされません。
- Windows Server LTSC 2022 Standard および Datacenter エディション、各エディションにデスクトップエクスペリエンスインストールオプション付き: 上記のエディションのみサポートされます。Windows Server 2022 Essentials エディションはサポートされません。

dSPACE ソフトウェアを Windows の機能と組み合わせて使用する場合には、いくつかの制限事項が適用されます。「[Windows 機能の使用に関する制限事項](#)」(270 ページ)を参照してください。

**Windows Docker のサポート** 一部の dSPACE 製品は、Windows Docker コンテナでも実行することができます。詳細については、dSPACE サポート ([http://www.dspace.jp/go/jpn\\_supportrequest](http://www.dspace.jp/go/jpn_supportrequest)) にお問い合わせください。

## ホスト PC の Linux オペレーティングシステム

dSPACE Release 2023-B の Linux と互換性がある dSPACE 製品は、以下のオペレーティングシステムをサポートしています。

- デスクトップ、サーバ、およびクラウドバージョンの（それぞれ AMD 64 ビットプロセッサアーキテクチャをベースにした）一般利用可能 (GA) カーネルが使用されている Ubuntu 20.04 LTS
- Ubuntu 18.04 LTS のサポートは終了しました。

dSPACE ソフトウェアを Linux の機能と組み合わせて使用する場合には、いくつかの制限事項が適用されます。「Linux 機能の使用に関する制限事項」(272 ページ)を参照してください。

**Linux Docker のサポート** Linux 互換の dSPACE 製品は、Linux Docker コンテナでも実行することができます。詳細については、dSPACE サポート ([http://www.dspace.jp/go/jpn\\_supportrequest](http://www.dspace.jp/go/jpn_supportrequest)) にお問い合わせください。

## フローティングネットワークライセンス用サーバのオペレーティングシステム

フローティングネットワークライセンスを購入した場合は、ネットワーク PC の 1 台をライセンスサーバとして指定する必要があります。CodeMeter Runtime ソフトウェアを使用するどの PC でもライセンスサーバとして使用することができます。

**dSPACE ソフトウェアを使用しないサーバの場合** ライセンスサーバでは、保護された dSPACE ソフトウェアと組み合わせて Microsoft Windows のオペレーティングシステムを使用することができます。

### 注記

- ライセンスサーバのオペレーティングシステムに Ubuntu Linux を使用する場合は、dSPACE サポート ([www.dspace.jp/go/jpn\\_supportrequest](http://www.dspace.jp/go/jpn_supportrequest)) にお問い合わせください。
- その他のオペレーティングシステムはテストされていません。このようなオペレーティングシステムはユーザご自身の責任において使用することができません。その場合、dSPACE はサポートを提供しません。

**dSPACE Installation Manager を使用するサーバの場合** dSPACE Installation Manager は、上述のように、他の dSPACE ソフトウェア製品と同じオペレーティングシステムをサポートします。

## 通信許可

**追加のファイアウォールルールのインストール** 各種 dSPACE ソフトウェア製品のインストール時には、Windows のファイアウォールルールが追加してインストールされます。その 1 つは、AutoBox などの dSPACE 拡張ボックスとの通信を許可するためのルールです。もう 1 つは、MotionDesk によるネットワークチャンネルからのモーションデータの受信を許可するためのルールです。これらのルールは、次のコマンドで生成されます。

- `netsh advfirewall firewall add rule name="dSPACE Net Service" service=any dir=in action=allow profile=any protocol=icmpv4:0, any description="Allow the dSPACE Net Service to connect to a dSPACE expansion box via network."`
- `netsh advfirewall firewall add rule name="dSPACE MotionDesk" program=<main installation path>\dSPACE MotionDesk 2023-B\MotionDesk\Bin\MotionDesk.exe"`



```
dir=in action=allow profile=any description="Allow dSPACE
MotionDesk to receive motion data via network."
```

**開いている TCP/IP ネットワークポートの要件** ホスト PC でサードパーティ製ファイアウォールソフトウェアを使用している場合は、dSPACE ソフトウェアの TCP/IP 通信がブロックされないかどうか確認してください。

- VEOS では、次の TCP/IP ネットワークポートが開いている必要があります：
  - Windows の場合：111 (TCP および UDP)、3702 (UDP)、7214 (TCP および TCP6)、7215 (TCP および UDP)、7216 (TCP)、8090 (TCP)、9923 (UDP)、49152 ...65535 (TCP、TCP6 および UDP)
  - Linux の場合：111 (TCP および UDP)、7215 (TCP および TCP6)、7216 (TCP)、8090 (UDP)、9923 (UDP)、32768 ...60999 (TCP、TCP6 および UDP)
- MotionDesk では、次の TCP/IP ネットワークポートが開いている必要があります：15000 (UDP)
- ConfigurationDesk を 1 つまたは複数の接続された MATLAB インストレーションとともに使用するには、最大 10 個の未使用の TCP/IP ネットワークポートが必要です。ポート範囲は 49196~49205 (TCP) です。
  - 接続された MATLAB が開始されるたびに、指定された範囲から最初に利用可能なポートが使用されます。
  - また、ConfigurationDesk を起動すると、指定した範囲から最初に空いているポートを使用します。
  - ConfigurationDesk が実行されていない場合、または MATLAB インスタンスが実行されていない場合は、指定された範囲のポートは使用されません。
- dSPACE Installation Manager および CodeMeter ライセンスソフトウェアには、以下の TCP/IP ネットワークポートが必要です：
  - LAN ネットワーク通信用に 22350 (TCP および UDP) (デフォルト設定から変更していない場合)。
  - 22352 (TCP および UDP) : http を介して CodeMeter WebAdmin にアクセスする場合。
  - 22353 (TCP および UDP) : https を介して CodeMeter WebAdmin にアクセスする場合。
- dSPACE Help には、コンポーネント間のプロセス間通信のためのオープン TCP/IP ネットワークポートが必要です。デフォルトポート番号は 11000 です。このポート番号が既に使用されている場合、別の空きポートが自動的に使用されます。関連するプロセスは、次のプレフィックスを介して識別することができます。HelpAbsLayer<xxx>、HelpInstaller<xxx>。

## 仮想マシン (VM) での dSPACE ソフトウェアの使用

### 概要

仮想マシンにインストールされたいくつかの dSPACE 製品を操作することができます。ただし、一部の dSPACE 製品は制限付きでのみ VM をサポートし、その他の dSPACE 製品は VM 上ではまったく操作することができません。

使用上の制限事項

注記

dSPACE エンドユーザ使用許諾契約書 (EULA) では、以下のことは禁止されています。

- ライセンス保護メカニズムの迂回、取得したライセンスの複数使用、またはライセンスタイプによって定められた使用以外の使用を目的として仮想マシンを使用すること。

質問や発生した問題については、dSPACE サポート

([www.dspace.jp/go/jpn\\_supportrequest](http://www.dspace.jp/go/jpn_supportrequest)) にお問い合わせください。

推奨される仮想マシンソフトウェア

dSPACE では、最新の VMware 製品および VM ハードウェア互換性バージョン 10 および 13 を使用して dSPACE ソフトウェア製品の機能をテストしています。

ホストのオペレーティングシステムとして Windows、Linux、または macOS® を使用することができます。

仮想マシンでの dSPACE ソフトウェアのサポート

注記

次の表にすべての dSPACE 製品の互換性を示します。制限付きで VM をサポートする製品については、既知の制限事項がリストされています。これらの製品では、ユースケースによってはその他の制限事項が適用される場合があります。

| 製品                          | フルサポート | 一定の制限付きサポート                                                                                                                                                                                                                                                         | サポートなし |
|-----------------------------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| AutomationDesk              | —      | ✓ 既知の制限事項: <ul style="list-style-type: none"> <li>▪ dSPACE リンクボードを介した DS1006 モジュール型システムへのアクセスは行うことができません。</li> <li>▪ Ethernet 接続とスロット CPU を介した DS1006 モジュール型システムへのアクセス: 通信および性能が非常に低くなります。</li> <li>▪ DS1104 R&amp;D Controller Board へのアクセスは行うことができません。</li> </ul> | —      |
| Automotive Simulation Model | ✓      | —                                                                                                                                                                                                                                                                   | —      |
| Bus Manager (スタンドアロン)       | ✓      | —                                                                                                                                                                                                                                                                   | —      |
| Calibration API Package     | ✓      | —                                                                                                                                                                                                                                                                   | —      |
| ConfigurationDesk           | ✓      | —                                                                                                                                                                                                                                                                   | —      |
| ControlDesk                 | —      | ✓ 既知の制限事項: <ul style="list-style-type: none"> <li>▪ dSPACE リンクボードを介した DS1006 モジュール型システムへのアクセスは行うことができません。</li> <li>▪ Ethernet 接続とスロット CPU を介した DS1006 モジュール型システムへのアクセス: 通信および性能が非常に低くなります。</li> <li>▪ DS1104 R&amp;D Controller Board へのアクセスは行うことができません。</li> </ul> | —      |
| Data Dictionary Manager     | ✓      | —                                                                                                                                                                                                                                                                   | —      |
| DCI Configuration Tool      | ✓      | —                                                                                                                                                                                                                                                                   | —      |
| dSPACE AUTOSAR Compare      | ✓      | —                                                                                                                                                                                                                                                                   | —      |

| 製品                                   | フルサポート | 一定の制限付きサポート                                                                                                                                                                                          | サポートなし          |
|--------------------------------------|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| dSPACE Bus API Package               | —      | ✓ 既知の制限事項:<br>▪ USB インターフェースの使用はサポートされていません。                                                                                                                                                         | —               |
| dSPACE CAN API Package               | —      | ✓ 既知の制限事項:<br>▪ USB インターフェースの使用はサポートされていません。                                                                                                                                                         | —               |
| dSPACE ECU Flash Programming Tool    | ✓      | —                                                                                                                                                                                                    | —               |
| dSPACE FlexRay Configuration Package | ✓      | —                                                                                                                                                                                                    | —               |
| dSPACE Installation Manager          | ✓      | —                                                                                                                                                                                                    | —               |
| ECU Interface Manager                | ✓      | —                                                                                                                                                                                                    | —               |
| ECU bypassing target compiler        | ✓      | —                                                                                                                                                                                                    | —               |
| Failure Simulation                   | —      | ✓ 既知の制限事項:<br>▪ dSPACE リンクボードを介した DS1006 モジュール型システムへのアクセスは行うことができません。<br>▪ Ethernet 接続とスロット CPU を介した DS1006 モジュール型システムへのアクセス: 通信および性能が非常に低くなります。<br>▪ DS1104 R&D Controller Board へのアクセスは行うことができません。 | —               |
| Firmware Archives                    | —      | ✓ 既知の制限事項:<br>▪ dSPACE リンクボードを介した DS1006 モジュール型システムへのアクセスは行うことができません。<br>▪ Ethernet 接続とスロット CPU を介した DS1006 モジュール型システムへのアクセス: 通信および性能が非常に低くなります。<br>▪ DS1104 R&D Controller Board へのアクセスは行うことができません。 | —               |
| Firmware Manager                     | —      |                                                                                                                                                                                                      |                 |
| FPGA Programming Blockset            | —      | ✓ 既知の制限事項:<br>▪ dSPACE リンクボードを介した DS1006 モジュール型システムへのアクセスは行うことができません。<br>▪ Ethernet 接続とスロット CPU を介した DS1006 モジュール型システムへのアクセス: 通信および性能が非常に低くなります。<br>▪ DS1104 R&D Controller Board へのアクセスは行うことができません。 | —               |
| Model Compare                        | ✓      | —                                                                                                                                                                                                    | —               |
| Model Container Utility              | ✓      | —                                                                                                                                                                                                    | —               |
| ModelDesk                            | —      | ✓ 既知の制限事項:<br>▪ Traffic Object Manager は、プレビューでカスタムセンサポイントを示すことができません。<br>▪ 開始トリガが使用されている場合、時折、プロットが開始されない場合があります。                                                                                   | —               |
| Model Interface Package for Simulink | ✓      | —                                                                                                                                                                                                    | —               |
| MotionDesk                           | —      | —                                                                                                                                                                                                    | ✓ <sup>1)</sup> |
| Platform API Package                 | —      | ✓ 既知の制限事項:<br>▪ dSPACE リンクボードを介した DS1006 モジュール型システムへのアクセスは行うことができません。<br>▪ Ethernet 接続とスロット CPU を介した DS1006 モジュール型システムへのアクセス: 通信および性能が非常に低くなります。<br>▪ DS1104 R&D Controller Board へのアクセスは行うことができません。 | —               |

| 製品                                  | フルサポート          | 一定の制限付きサポート                                                                                                                                                                                                                                                         | サポートなし |
|-------------------------------------|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| Real-Time Testing                   | —               | ✓ 既知の制限事項: <ul style="list-style-type: none"> <li>▪ dSPACE リンクボードを介した DS1006 モジュール型システムへのアクセスは行うことができません。</li> <li>▪ Ethernet 接続とスロット CPU を介した DS1006 モジュール型システムへのアクセス: 通信および性能が非常に低くなります。</li> <li>▪ DS1104 R&amp;D Controller Board へのアクセスは行うことができません。</li> </ul> | —      |
| RTI Blocksets (Real-Time Interface) | —               | ✓ 既知の制限事項: <ul style="list-style-type: none"> <li>▪ dSPACE リンクボードを介した DS1006 モジュール型システムへのアクセスは行うことができません。</li> <li>▪ Ethernet 接続とスロット CPU を介した DS1006 モジュール型システムへのアクセス: 通信および性能が非常に低くなります。</li> <li>▪ DS1104 R&amp;D Controller Board へのアクセスは行うことができません。</li> </ul> | —      |
| Signal List Editor                  | ✓               | —                                                                                                                                                                                                                                                                   | —      |
| SYNECT                              | ✓               | —                                                                                                                                                                                                                                                                   | —      |
| SYNECT Server                       | ✓               | —                                                                                                                                                                                                                                                                   | —      |
| SYNECT License Server               | ✓               | —                                                                                                                                                                                                                                                                   | —      |
| SystemDesk                          | ✓               | —                                                                                                                                                                                                                                                                   | —      |
| TargetLink                          | ✓               | —                                                                                                                                                                                                                                                                   | —      |
| VEOS                                | ✓ <sup>2)</sup> | —                                                                                                                                                                                                                                                                   | —      |

1) VM はグラフィックカードの要件を満たしていません。

2) Adaptive AUTOSAR V-ECU をシミュレートする場合は、「Windows でのアダプティブ V-ECU シミュレーションのためのハイパーバイザの設定」(『VEOS マニュアル』)を参照してください。

### 仮想マシンのセットアップに必要な知識

仮想マシンをセットアップするには、VM テクノロジーに関する知識が必要です。

仮想環境では、物理 PC と比較してはるかに高いレイテンシと低いネットワーク性能(ネットワークスループット)を想定する必要があります。

**仮想マシンの並列使用** 1 台の PC で複数の VM を同時に使用する場合、CPU、ネットワーク、ディスク I/O 帯域幅などのホストリソースの共有により、タイミングの問題が発生する場合があります。dSPACE は、アプリケーションで高性能が必要な場合は、物理的な PC の使用をお勧めします。

### ホスト PC 上での仮想マシンの使用

**システム要件** dSPACE ソフトウェアを使用する仮想マシンのホスト PC として使用されるコンピュータは、少なくとも「付録: システム要件」(『dSPACE ソフトウェアのインストール』)に記載された要件を満たす必要があります。ソフトウェアが VM 上でスムーズに実行できるようにリソースに余裕がある PC を使用することをお勧めします。これは、VM ソフトウェア自体がかなりのリソースを使用するためです。

- ホスト PC 上のオペレーティングシステムとソフトウェアだけでなく、VM 上のゲストオペレーティングシステムとアプリケーションソフトウェアを実行するのに十分な CPU 速度と RAM サイズが必要です。
- VM ソフトウェアとそこで実行するソフトウェアを PC に直接インストールするのと同じくらいの、十分なディスクの空きスペースも必要になります。

**ドングルベースのデバイスの接続** ドングルベースのシングルユーザライセンスを使用して dSPACE ソフトウェアを使用する場合、まずホスト PC に CmDongle を接続する必要があります。次に、ホスト PC 上の仮想マシンに WIBU-Systems CodeMeter-Stick デバイスを接続する必要があります。手順については、使用する VM ソフトウェアのマニュアルを参照してください。

**フローティングネットワークライセンスの使用** フローティングネットワークライセンスを使用する場合は、仮想マシンは dSPACE License Server へのアクセスが必要です。詳細については、「[クライアント/サーバ間の接続をセットアップする方法](#)」(『CodeMeter ライセンステクノロジーの使用』)を参照してください。

**dSPACE Help の最適な表示** dSPACE Help の内容を最適な状態で表示するには、VM で ClearType 設定 (= デフォルト設定) を有効にする必要があります。この設定には、Windows のスタートメニュー ([スタート] – [コントロール パネル] – [外観と個人設定] – [表示] – [ClearType テキストの調整]) からアクセスすることができます。

#### ‘スナップショットへの復元’機能の使用

##### 通知

**‘スナップショットへの復元’機能を使用すると、ライセンスが無効になります。**

仮想マシンで‘スナップショットへの復元’機能を使用すると、ご使用のホスト PC 上のソフトウェアベースの CmContainer (dSPACE Activation Container および dSPACE Borrow Container) はすべて無効になり、これらに含まれているライセンスは失われます。

- ライセンスをアクティブ化したソフトウェアベースの CmContainer を含む仮想マシンで‘スナップショットへの復元’機能を使用しないでください。
- ライセンス情報は CmDongles に格納します。この場所では、‘スナップショットへの復元’機能を使用した後も CmContainers は無効になりません。

#### ハードウェア設定が異なるホスト PC への仮想マシンの移動

##### 通知

**ハードウェア設定が異なるホスト PC に仮想マシンを移動すると、ライセンスが無効になります。**

ハードウェア設定が異なるホスト PC に仮想マシンを移動すると、ご使用のホスト PC 上のソフトウェアベースの CmContainer (dSPACE Activation Container および dSPACE Borrow Container) はすべて無効になり、これらに含まれているライセンスは失われます。これは、たとえば、物理的な PC の CPU タイプが変わった場合に起こります。

- ライセンスを有効にしたソフトウェアベースの CmContainer を含む仮想マシンは、ハードウェア設定が異なるホスト PC に移動しないでください。
- ライセンス情報は CmDongles に格納します。この場所では、仮想マシンを移動した後も CmContainers が無効になりません。
- ただし、仮想マシンをどうしても移動する必要がある場合は、長いダウンタイムを避けられるソリューションを相談するために、事前に dSPACE サポート ([www.dspace.jp/go/jpn\\_supportrequest](http://www.dspace.jp/go/jpn_supportrequest)) にお問い合わせください。

## dSPACE ソフトウェアのランタイム互換性

|                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>定義</b>                           | <p>ランタイム互換性とは、以下のことを意味します。</p> <ul style="list-style-type: none"> <li>▪ 別々のフォルダにインストールされている場合でも、複数の dSPACE 製品の同時使用が可能</li> <li>▪ 相互作用なく個別に dSPACE 製品を使用可能</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>dSPACE Release 2023-B の製品互換性</b> | <p>dSPACE では、同一の dSPACE Release のソフトウェア製品のみ使用することをお勧めしています。これにより、最大限のランタイム互換性が保証されます。</p> <p>以下の点を確認してください。</p> <ul style="list-style-type: none"> <li>▪ 異なる dSPACE Release の製品を併用した場合、dSPACE ツールチェーンでランタイム互換性に関連する制限が生じる可能性があります。</li> </ul> <p>dSPACE 製品が(自動化インターフェースを介して)直接連携する場合や、(A2L のような共通のファイルタイプを介して)間接的に連携する場合は、制限事項が適用されることがあります。詳細な制限事項については、該当する製品のマニュアルを参照してください。主要な制限事項については、以下を参照してください。</p> <p>まれに、ランタイム互換を実行するために製品に追加のパッチをインストールする必要がある場合があります。パッチに関する情報およびパッチの必要性については、<a href="https://www.dspace.jp/go/CompPatch">https://www.dspace.jp/go/CompPatch</a> を参照してください。</p> <ul style="list-style-type: none"> <li>▪ Release 2023-B の RCP and HIL Software 製品は、それより前の dSPACE Release の RCP and HIL Software 製品と併用することはできません。</li> </ul> |
| <b>以前のリリースの dSPACE 製品との併用</b>       | <p>以前のリリースの複数の製品を併用する場合の注意事項については、<a href="https://www.dspace.jp/goto.cfm/ja_0501">https://www.dspace.jp/goto.cfm/ja_0501</a> を参照してください。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

## Windows 機能の使用に関する制限事項

|                                                     |                                                                                                                                            |
|-----------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| <b>目的</b>                                           | <p>dSPACE ソフトウェアを Windows の機能と組み合わせて使用する場合には、いくつかの制限事項が適用されます。</p>                                                                         |
| <b>Windows サービスアカウント内での dSPACE ソフトウェアのインストールと実行</b> | <p>非サービススペースの dSPACE ソフトウェアは、あらかじめ設定された Windows サービスアカウント (LocalService、NetworkService、LocalSystem) のコンテキストでインストールまたは実行するように設計されていません。</p> |
| <b>ユーザの簡易切り替えのサポートなし</b>                            | <p>dSPACE ソフトウェアは、Windows のユーザの簡易切り替えをサポートしません。</p>                                                                                        |
| <b>PC をシャットダウンする前に dSPACE ソフトウェアを閉じる</b>            | <p>Windows オペレーティングシステムのシャットダウンプロセスでは、いくつかの必要なプロセスが、dSPACE ソフトウェアによって利用されている状態であっても中断さ</p>                                                 |



れることがあります。データの損失を回避するため、PC をシャットダウンする前に dSPACE ソフトウェアを手動で閉じることをお勧めします。

#### ユーザアカウント制御

dSPACE ソフトウェアをインストールするときは、Windows のユーザアカウント制御 (UAC) を無効にすることをお勧めします。UAC を無効にできない場合は、Windows の次の動作に注意してください: UAC を有効にしていると、セットアッププログラムはユーザのアカウントではなく管理者アカウントを使用します。そのため、管理者アカウントが必要なドライブ、特に必要なネットワークドライブへのアクセス権を持つことが重要です。

#### USB デバイス

光絶縁対応ケーブルを使用する dSPACE USB デバイスを初めて PC に接続すると、デバイスドライバソフトウェアが正常にインストールされなかったことを示すメッセージが表示されます。ただし、dSPACE デバイスはその後正常に動作します。

#### 高 DPI モニターの使用

Real-Time Testing には、高 DPI モニターでの作業に関する制限があります。RTT シーケンスを取り扱うユーザインターフェース、Real-Time Test Manager では、高 DPI モニターモニターの使用はサポートされていません。

#### ハイコントラストモード非対応

dSPACE ソフトウェアは、Windows のハイコントラストモード用に開発されたものでも、テストされたものでもありません。dSPACE ソフトウェアを使用する場合は、ハイコントラストモードを無効にすることをお勧めします。無効にしない場合、ユーザインターフェースのビジュアル表示が正常に機能しない可能性があります。

#### FIPS のサポート

dSPACE ソフトウェアは、FIPS PUB 140-2 米国政府コンピュータセキュリティ規格 (暗号モジュールに関するセキュリティ要件) に準拠して開発またはテストされていません。FIPS の詳細については、<https://docs.microsoft.com/en-us/windows/security/threat-protection/fips-140-validation> を参照してください。

#### 長いパス

dSPACE ソフトウェアは、Windows API の長いパス構文をサポートしていません。260 文字を超えるパスが直接または間接的に使用された場合の dSPACE ソフトウェアの動作は未確認です。

#### Windows の [8dot3name の作成] オプションの有効化

##### 注記

MATLAB®/Simulink®などのサードパーティ製ソフトウェアおよび dSPACE ソフトウェアをインストールする前に、Windows の [8dot3name の作成] オプションをすべてのドライブ (インストールに使用するドライブと作業に使用するドライブ) で有効にすることを強くお勧めします。

ソフトウェアのインストール時にこのオプションが無効な場合、dSPACE ソフトウェアの実行時に重大なエラーが発生する可能性があります。たとえば、ビルドプロセスの中断などが起きる可能性があります。[8dot3name の作成] オプションが無効な間にインストールしたインストレーションを修復するには、dSPACE ソフトウェアと必要なサードパーティ製ソフトウェアを再インストールする必要があります。

設定の確認とオプションの有効化の手順については、<https://www.dspace.com/ja/jpn/home/support/kb/faqs/faq346.cfm> または Microsoft Windows のマニュアルを参照してください。

---

#### Windows でユーザロケールとシステムロケールの設定は一致する必要がある

MATLAB は、Windows オペレーティングシステムで指定されているユーザロケールとシステムロケールを読み取ります。ユーザロケールとシステムロケールは一致する必要があります。これらの設定が同じでない場合、システムは MATLAB および dSPACE ソフトウェアの使用時に予期しない動作をすることがあります。

この設定の確認および変更については、[https://www.mathworks.com/help/matlab/matlab\\_env/setting-locale-on-windows-platforms.html?s\\_tid=gn\\_loc\\_drop](https://www.mathworks.com/help/matlab/matlab_env/setting-locale-on-windows-platforms.html?s_tid=gn_loc_drop) を参照してください。

これは、dSPACE でサポートされる、すべての MATLAB バージョンおよびすべての Windows オペレーティングシステムに影響します。

---

#### シングルユーザライセンスの制限事項

Windows Server 2019 などの Windows Server オペレーティングシステムを搭載した PC を使用し、Microsoft Remote Desktop Connection を介してこの PC で dSPACE ソフトウェアを使用する場合、フローティングネットワークライセンスを使用して、ライセンス保護されたソフトウェアをアクティブ化する必要があります。この場合、シングルユーザライセンスはサポートされません。

## Linux 機能の使用に関する制限事項

---

#### FIPS のサポート

dSPACE ソフトウェアは、FIPS PUB 140-2 米国政府コンピュータセキュリティ規格 (暗号モジュールに関するセキュリティ要件) に準拠して開発またはテストされていません。FIPS の詳細については、以下を参照してください。

- Ubuntu 20.04 LTS: <https://ubuntu.com/blog/fips-certification-ubuntu-20-04-lts>

---

#### 長いパス

dSPACE ソフトウェアは、4,096 文字の使用可能なパスの長さをサポートしていません。260 文字を超えるパスが直接または間接的に使用された場合の dSPACE ソフトウェアの動作は未確認です。

---

#### ロケール

dSPACE ソフトウェアは、US English ロケールでのみシステムのテストを実施しています。



**A**

ASM Battery Blockset  
新機能 43  
移行 44

ASM Diesel Engine Blockset  
移行 45

ASM Drivetrain Basic Blockset  
新機能 46  
移行 46

ASM Electric Components Blockset  
新機能 47  
移行 47

ASM Engine Gasoline Blockset  
移行 50

ASM Fuel Cell Blockset  
新機能 48  
移行 49

ASM Trailer Blockset  
新機能 52

ASM Truck Blockset  
新機能 54

ASM Vehicle Dynamics Blockset  
新機能 56  
移行 57

ASM ブロックセット  
移行 42

AutomationDesk  
新機能 39  
移行 40

**B**

Bus Manager (スタンドアロン)  
プロジェクトとアプリケーションの移行 62  
新機能 59  
移行 62

**C**

ConfigurationDesk  
新機能 67  
移行 79

ConfigurationDesk の Bus Manager  
新機能 74

ConfigurationDesk プロジェクトとアプリケーション  
の移行 83

ControlDesk  
新機能 88  
移行 104

**D**

DCI Configuration Tool  
新機能 107

dSPACE ECU Flash Programming Tool  
新機能 109

dSPACE FlexRay Configuration Package  
新機能 111

dSPACE Python Extensions  
新機能 117

dSPACE XIL API  
新機能 119  
移行 120

**E**

ECU Interface Manager  
新機能 123  
移行 124

**F**

Firmware Manager  
新機能 127

FPGA Programming Blockset  
新機能 129

**L**

Linux  
制限事項 272

Linux 機能の使用に関する制限事項 272

**M**

MATLAB  
サポートされるリリース 261  
要件 261

MicroAutoBox III のファームウェア  
新機能 135

Model and Sensor Interface  
移行 139

Model Compare  
新機能 137  
移行 138

Model Interface Package for Simulink  
新機能 143  
移行 144

ModelDesk  
新機能 141  
移行 142

MotionDesk  
新機能 145  
移行 145

**R**

Real-Time Testing  
移行 147

RTI Bypass Blockset  
移行 151

RTI CAN MultiMessage Blockset  
新機能 153  
移行 153

RTI FPGA Programming Blockset  
移行 131

RTI LIN MultiMessage Blockset  
新機能 155  
移行 155

RTI Synchronized Time Base Manager Blockset  
新機能 157

RTI/RTI-MP  
新機能 149

RTLib  
新機能 149

**S**

SCALEXIO Firmware  
新機能 159

移行 160

Signal List Editor  
新機能 163

SystemDesk  
新機能 170

**T**

TargetLink Data Dictionary  
移行 201  
ライブラリとモデルの手動アップグレード 204  
廃止されたドキュメント 201  
既存のデータディクショナリのアップグレード 204

**V**

VEOS  
新機能 253

**W**

Windows  
制限事項 270

Windows 機能の使用に関する制限事項 270

**サ**

サポートしている MATLAB リリース 261

**シ**

システム要件  
Linux オペレーティングシステム 264  
Windows オペレーティングシステム 263

**ト**

ドキュメントフォルダ 12

**ハ**

バージョン履歴 17

**ホ**

ホスト PC のソフトウェア  
Linux オペレーティングシステム 264  
MATLAB 261  
Windows オペレーティングシステム 263

**ロ**

ローカルプログラムデータフォルダ 12

一般的な機能拡張および変更 13

主な機能 20

共通プログラムデータフォルダ 12

廃止 15

ソフトウェアのサポート 15

予定 (ソフトウェアサポート) 15

予定 (ハードウェア) 16

新しいハードウェア 14

新機能

ASM Battery Blockset 43

ASM Drivetrain Basic Blockset 46

ASM Electric Components Blockset 47

- ASM Fuel Cell Blockset 48
- ASM Trailer Blockset 52
- ASM Truck Blockset 54
- ASM Vehicle Dynamics Blockset 56
- AutomationDesk 39
- Bus Manager(スタンドアロン) 59
- ConfigurationDesk 67
- ConfigurationDesk の Bus Manager 74
- ControlDesk 88
- DCI Configuration Tool 107
- dSPACE ECU Flash Programming Tool 109
- dSPACE FlexRay Configuration Package 111
- dSPACE Python Extensions 117
- dSPACE XIL API 119
- ECU Interface Manager 123
- Firmware Manager 127
- FPGA Programming Blockset 129
- MicroAutoBox III のファームウェア 135
- Model Compare 137
- Model Interface Package for Simulink 143
- ModelDesk 141
- MotionDesk 145
- RTI CAN MultiMessage Blockset 153
- RTI LIN MultiMessage Blockset 155
- RTI Synchronized Time Base Manager Blockset 157
- RTI/RTI-MP 149
- RTLib 149
- SCALEXIO Firmware 159
- Signal List Editor 163
- SystemDesk 170
- VEOS 253
- 移行**
  - ASM Battery Blockset 44
  - ASM Diesel Engine Blockset 45
  - ASM Drivetrain Basic Blockset 46
  - ASM Electric Components Blockset 47
  - ASM Engine Gasoline Blockset 50
  - ASM Fuel Cell Blockset 49
  - ASM Vehicle Dynamics Blockset 57
  - ASM ブロックセット 42
  - AutomationDesk 40
  - Bus Manager(スタンドアロン) 62
    - プロジェクトとアプリケーション 62
  - ConfigurationDesk 79
  - ConfigurationDesk のプロジェクトとアプリケーション 83
  - ControlDesk 104
  - dSPACE XIL API 120
  - ECU Interface Manager 124
  - Model and Sensor Interface 139
  - Model Compare 138
  - Model Interface Package for Simulink 144
  - ModelDesk 142
  - MotionDesk 145
  - Real-Time Testing 147
  - RTI 149
  - RTI Bypass Blockset 151
  - RTI CAN MultiMessage Blockset 153
  - RTI FPGA Programming Blockset 131
  - RTI LIN MultiMessage Blockset 155
  - SCALEXIO Firmware 160
- 要件**
  - ホスト PC の Linux オペレーティングシステム 264
  - ホスト PC の Windows オペレーティングシステム 263
  - ホスト PC のソフトウェア
    - MATLAB 261
- 製品の概要** 17