

dSPACE Release

# 新機能と移行手順

Release 2022-B – 2022 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 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.com/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 - 2022  
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 2022-B の概要	13
一般的な機能拡張および変更	13
dSPACE Help の新機能	16
廃止	16
製品バージョンの概要	18
製品の主な新機能	21
以前のリリースからの移行について	27
dSPACE Release 2022-B への移行	27
AutomationDesk	29
AutomationDesk 2022-B の新機能	29
AutomationDesk 2022-B への移行	29
Automotive Simulation Model (ASM)	33
すべての ASM 製品	34
すべての ASM ブロックセットの新機能	34
ASM Battery	35
ASM Battery Blockset 2022-B の新機能	35
ASM Electric Components	36
ASM Electric Components デモモデルの変更	36
ASM Environment	37
ASM Environment Blockset 2022-B の新機能	37
ASM Environment Blockset 2022-B への移行	37
ASM Fuel Cell	38
ASM Fuel Cell Blockset 2022-B の新機能	38
ASM Fuel Cell デモモデルの変更	38
ASM Fuel Cell Blockset 2022-B への移行	39
ASM Utils	40
ASM Utils 2022-B の新機能	40
ASM Utils 2022-B への移行	40

ASM Traffic.....	41
ASM Traffic デモモデルの変更.....	41
ASM Traffic Blockset 2022-B への移行.....	42
ASM Truck.....	43
ASM Truck デモモデルの変更.....	43
ASM Vehicle Dynamics.....	44
ASM Vehicle Dynamics Blockset 2022-B の新機能.....	44
<b>Bus Manager (スタンドアロン)</b> .....	<b>45</b>
Bus Manager (スタンドアロン) 2022-B の新機能.....	45
Bus Manager (スタンドアロン) 2022-B への移行.....	47
Bus Manager (スタンドアロン) の廃止.....	49
<b>ConfigurationDesk</b> .....	<b>51</b>
ConfigurationDesk 2022-B の新しい一般機能.....	51
I/O 機能に関する新機能とハードウェアサポート.....	54
ConfigurationDesk の Bus Manager の新機能.....	55
サポートされるコンテナファイルバージョン.....	57
ConfigurationDesk 2022-B への移行.....	60
ConfigurationDesk の廃止.....	62
<b>ControlDesk</b> .....	<b>63</b>
ControlDesk 2022-B の新機能.....	64
新しいユーザインターフェース処理機能 (ControlDesk 2022-B).....	64
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 2022-B).....	65
新しい変数管理機能 (ControlDesk 2022-B).....	65
新しい計測および記録機能 (ControlDesk 2022-B).....	66
新しい Bus Navigator 機能 (ControlDesk 2022-B).....	67
新しい自動化機能 (ControlDesk 2022-B).....	70
ControlDesk 2022-B への移行.....	71
ControlDesk 2022-B への移行.....	71
<b>dSPACE AUTOSAR Compare</b> .....	<b>75</b>
dSPACE AUTOSAR Compare 2022-B の新機能.....	75
dSPACE AUTOSAR Compare 1.1 の新機能.....	75

DCI Configuration Tool	79
DCI Configuration Tool 2022-B の新機能.....	79
dSPACE FlexRay Configuration Package	81
dSPACE FlexRay Configuration Package 2022-B の新機能.....	81
dSPACE Installation Manager	83
dSPACE Installation Manager 22.2 の新機能.....	83
dSPACE Installation Manager 22.2 への移行.....	84
dSPACE Python Extensions	85
dSPACE Python Extensions 2022-B の新機能.....	85
dSPACE XIL API .NET	87
dSPACE XIL API .NET 2022-B の新機能.....	87
dSPACE XIL API .NET 2022-B への移行.....	88
ECU Interface Manager	91
ECU Interface Manager 2022-B の新機能.....	91
ECU Interface Manager 2022-B の互換性.....	92
ECU Interface Manager 2022-B への移行.....	93
Firmware Manager	95
Firmware Manager 22.2 の新機能.....	95
FPGA Programming Blockset	97
FPGA Programming Blockset 2022-B の新機能.....	97
FPGA Programming Blockset 2022-B への移行.....	100
MicroAutoBox III のファームウェア	103
MicroAutoBox III Firmware 22.2 の新機能.....	103
Model Compare	105
Model Compare 2022-B の新機能.....	105
Model Compare 2022-B への移行.....	107

Model Container Utility	109
モデルコンテナユーティリティについて.....	109
Model および Sensor Interface Blockset	111
Model and Sensor Interface Blockset 2022-B の新機能.....	111
Model and Sensor Interface Blockset 2022-B への移行.....	112
ModelDesk	115
ModelDesk 2022-B の新機能.....	115
ModelDesk 2022-B への移行.....	115
Model Interface Package for Simulink	117
Model Interface Package for Simulink 2022-B の新機能.....	117
Model Interface Package for Simulink 2022-B への移行.....	118
MotionDesk	119
MotionDesk 2022-B の新機能.....	119
MotionDesk 2022-B への移行.....	119
Real-Time Testing	121
Real-Time Testing 2022-B の新機能.....	121
Real-Time Testing 2022-B への移行.....	122
RTI/RTI-MP および RTLib	123
RTI/RTI-MP および RTLib の新機能.....	123
RTI/RTI-MP および RTLib の移行上の注意点.....	123
RTI Bypass Blockset	125
RTI Bypass Blockset 2022-B への移行.....	125
RTI CAN MultiMessage Blockset	127
RTI CAN MultiMessage Blockset 2022-B への移行.....	127
RTI LIN MultiMessage Blockset	129
RTI LIN MultiMessage Blockset 2022-B への移行.....	129

<b>RTI Synchronized Time Base Manager Blockset</b>	<b>131</b>
RTI Synchronized Time Base Manager Blockset 2022-B の新機能.....	131
<b>SCALEXIO Firmware</b>	<b>133</b>
SCALEXIO Firmware 22.2 の新機能.....	133
SCALEXIO Firmware 22.2 への移行.....	134
SCALEXIO ファームウェアの廃止.....	134
<b>センサシミュレーション</b>	<b>135</b>
Sensor Simulation 2022-B の新機能.....	135
Sensor Simulation 2022-B への移行.....	135
<b>SYNECT</b>	<b>137</b>
SYNECT 2022-B の新機能.....	138
新機能.....	138
SYNECT 2022-B への移行.....	141
SYNECT 2.13 からの移行.....	141
データベースの移行.....	142
SYNECT 2.13 から SYNECT 2022-B へのデータモデルの変更.....	143
<b>SystemDesk</b>	<b>145</b>
SystemDesk 2022-B の新機能.....	146
新しい一般機能.....	146
新しい V-ECU 生成機能.....	148
SystemDesk 2022-B への移行.....	151
dSPACE Release 2023-B 以降の廃止.....	151
SystemDesk 2022-B への移行.....	152
<b>TargetLink</b>	<b>153</b>
TargetLink 2022-B の新機能.....	154
Simulink または Stateflow でのモデリング.....	154
Simulink ツールストリップの[TargetLink]タブ.....	155
バス処理の改善.....	156
構造体配列のモデリングの改善.....	157
Stateflow での時相論理のサポートの改善.....	158
Interpolation Using Prelookup ブロック.....	158

MATLAB コード.....	158
MATLAB コード関数と演算の拡張.....	158
Adaptive AUTOSAR.....	159
Adaptive AUTOSAR での INOUT オペレーション引数のサポート.....	159
Adaptive AUTOSAR Data Dictionary インポートおよびエクスポートに 関する変更.....	159
構造体配列のサポート (Adaptive AUTOSAR).....	160
Adaptive AUTOSAR データタイプ.....	160
Classic AUTOSAR.....	161
サポートされる Classic AUTOSAR リリース.....	161
Classic AUTOSAR での INOUT オペレーション引数のサポート.....	162
Classic AUTOSAR Data Dictionary インポートおよびエクスポートに関 する変更.....	162
AUTOSAR バージョン R21-11 に準拠した変更.....	162
構造体配列のサポートの改善 (Classic AUTOSAR).....	163
コード生成のコア機能.....	164
構造体配列のコード効率性の改善.....	164
モジュールのコード分類.....	165
MIL (Model-in-the-Loop) シミュレーション.....	165
オーバーフロー検出を無効にしてパフォーマンスを向上.....	165
ターゲットシミュレーション (PIL).....	166
2022-B でのターゲットシミュレーションモジュールの変更.....	166
ユーザビリティ.....	166
ユーザビリティに関する改良点.....	166
コードジェネレータオプション.....	167
新しいコードジェネレータオプション.....	167
API 関数とフックスクリプト.....	168
新しい API 関数.....	168
新しいフックスクリプト.....	168
その他.....	169
TargetLink のデモと例.....	169
TargetLink 2022-B および TargetLink Data Dictionary 2022-B への移行.....	171
移行に関する一般的な情報.....	171
モデル、ライブラリ、Data Dictionary のアップグレード.....	171
TargetLink バージョン間の移行に関する基礎.....	172
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.....	174
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.....	175
Data Dictionary の CodeDecorationSets への移行.....	176



正確に表現可能な浮動小数点数の丸め動作の変更 (Microsoft)	179
正確に表現可能な浮動小数点数の丸め動作の変更 (Microsoft)	179
TargetLink 5.2 から 2022-B への移行	179
コードジェネレータオプション	180
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	180
AUTOSAR	182
AUTOSAR に関する移行上の注意点	182
API 関数とフックスクリプト	185
TargetLink と TargetLink Data Dictionary API 関数の変更	185
フックスクリプトの変更	188
ルックアップテーブルブロックに関する移行上の注意点	188
Prelookup ブロックと Interpolation Using Prelookup ブロック	188
最適化	189
最適化に関する移行上の注意点	189
移行に関するその他の注意点	190
移行に関するその他の注意点	190
TargetLink 5.2 と TargetLink 2022-B 間でのコードの変更	196
構造関連	196
パス構造体関連	199
コード順序とループ構造	203
スコープの縮小関連	204
コード効率性の向上	205
Adaptive AUTOSAR 関連	208
Classic AUTOSAR 関連	211
Simulink Function サブシステム関連	214
その他	215
TargetLink 2022-B での廃止項目	218
廃止された TargetLink の機能	219
廃止された制限事項	219
廃止された API 関数	221
廃止されたフックスクリプト	221
今後の TargetLink バージョンでの変更予定	222
廃止予定の機能	222
廃止予定の API 関数	223
廃止されたコードジェネレータオプション	223
<b>VEOS</b>	<b>225</b>
VEOS 2022-B の新機能	225

VEOS 2022-B の互換性.....	229
VEOS 2022-B への移行.....	232
VEOS 2023-A での廃止項目.....	233

## 互換性情報 235

サポートしている MATLAB リリース.....	235
オペレーティングシステム.....	236
仮想マシン (VM) での dSPACE ソフトウェアの使用.....	239
dSPACE ソフトウェアのランタイム互換性.....	243
Windows 機能の使用に関する制限事項.....	244
Linux 機能の使用に関する制限事項.....	246





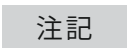



## 索引 249

# 本書について

**内容** 本書では、Release 2022-B に含まれるすべての dSPACE ソフトウェア製品の新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
 <b>危険</b>	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。
 <b>警告</b>	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。
 <b>注意</b>	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。
 <b>注意</b>	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。
 <b>注記</b>	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。
 <b>ヒント</b>	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 2022-B の概要

**概要** Release 2022-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
dSPACE Help の新機能.....	16
廃止.....	16
製品バージョンの概要.....	18
製品の主な新機能.....	21

## 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

dSPACE Release 2022-B に付属する Python 3.9 ディストリビューションには、次のパッケージが含まれています。

パッケージ	Python 3.9 Release 2022-B
comtypes	1.1.11
Core	3.9.13
cycler	0.11.0
fonttools	4.33.3
future	0.18.2
grpcio	1.43.0
grpcio_tools	1.43.0
kiwisolver	1.4.3
lxml	4.9.0
matplotlib	3.5.2
numpy	1.23.0
packaging	21.3
pillow	9.1.1
pip	22.1.2
protobuf	3.19.3
pyparser	2.21
pyglet	1.5.26
pyparsing	3.0.9
pypubsub	4.0.3
Python-dateutil	2.8.2
pythonnet	2.5.3 <sup>1)</sup>
pytz	2022.1
pywin32	302.10 <sup>1)</sup>
scipy	1.8.1
six	1.16.0
wxPython	4.1.1
yapsy	1.12.2

<sup>1)</sup> このパッケージには、dSPACE 固有のバグフィックスがいくつか含まれています。そのため、Python パッケージサーバの標準パッケージと置き換えないでください。

**注記****AutomationDesk または ModelDesk を使用するときの注意**

dSPACE Release 2022-B に付属する Python 3.9 ディストリビューションには、アプリケーションのバグフィックスやセキュリティアップデートを提供する最新の Python パッケージが含まれています。新しい numpy パッケージでは、いくつかの廃止されたメソッド (alen メソッドや asscalar メソッドなど) が削除されています。

影響を受ける dSPACE 製品は、AutomationDesk と ModelDesk (テスト機能付き) です。dSPACE Release 2022-B では、これらの製品は新しい numpy パッケージを使用するように準備されています。ただし、dSPACE Release 2022-B から Python 3.9 ディストリビューションをインストールし、dSPACE Release 2021-A、2021-B、2022-A の AutomationDesk または ModelDesk Testing を使用した場合、削除されたメソッドのいずれかが内部で呼び出されると、これらの製品が例外をスローするかエラーで停止することがあります。

dSPACE は、この問題を解決するために、影響を受けるバージョンの AutomationDesk 用のパッチを提供する予定です。ソフトウェアのアップデートとやパッチ自体、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。

上記のいずれかのバージョンの ModelDesk 用のパッチが必要な場合は、dSPACE サポートにお問い合わせください。

**新しいバージョン管理方式**

dSPACE 製品に採用された新しいバージョン管理は、アジャイルソフトウェア開発の原則と推奨事項に対応しています。カレンダーベースのバージョン管理方式では、バージョン名は年と連続した数字で構成されます。たとえば、22.3 は 2022 年にリリースされた 3 番目の製品バージョンを表します。

中間バージョンには、バージョンの 3 桁目を使用します。パッチとホットフィックスは、それぞれサフィックス *p*、*HF* と、それに続く数字を付けて表します。たとえば、次のような製品バージョン表記が可能です。22.2.1、22.2p1、22.2HF12345。

Release 製品には、製品バージョンの主要な識別子として Release バージョンが追加されます (例: 2022-B)。これにより、現在の製品バージョンの最初にリリースされた dSPACE Release を識別することができます。

Release 製品の API でバージョン情報が必要な場合にも、Release バージョンが使用されます。たとえば、登録されていない COM サーバをインスタンス化するには、*ControlDesk.Application.2022-B* といったように、特定のバージョンの COM サーバを呼び出す必要があります。

**プラットフォームサポート**

SCALEXIO システムに 64 ビット Linux オペレーティングシステムが搭載されました。これは、関連するすべての製品でサポートされています。

**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 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 6.3.0))
- Microsoft Visual Studio 2019 Professional

## dSPACE Help の新機能

**dSPACE Help に関する一般的な  
フィードバックの送信**

画面下部の[フィードバック]ボタンから dSPACE Help の一般的なフィードバックを送信することができます。

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)。。

The screenshot shows a feedback form with the following elements:

- Title: Feedback
- Text: Tell us about your experience with dSPACE Help. What do you like and/or how do you think we can improve?
- Label: Feedback category:
- Dropdown menu: -- Select a category -- (with a list of options: Home page, Layout, Search & Filters, Technical, Usability, None of the above)
- Button: Send Feedback

## 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。



<p><b>ソフトウェアサポートの廃止</b></p>	<p><b>オペレーティングシステム</b> dSPACE Release 2022-B 以降、Ubuntu 18.04 LTS はサポートされません。</p> <p><b>dSPACE CAN API およびその他のバス固有の API</b> dSPACE Release 2022-B では、dSPACE CAN API およびその他のバス固有の API は、32 ビット版アプリケーションで使用できなくなりました。 64 ビット版アプリケーションでの使用は、引き続きサポートされます。</p> <p><b>レーダー、LiDAR、レーザーセンサによるセンサシミュレーション</b> dSPACE Release 2022-B では、レーダー、LiDAR、およびレーザーセンサを使用した MotionDesk ベースのセンサシミュレーションが廃止されます。</p>
-----------------------------	---

<p><b>ソフトウェアサポートの廃止予定</b></p>	<p><b>MotionDesk および Sensor Simulation</b></p> <ul style="list-style-type: none"> <li>▪ MotionDesk をベースにしたセンサシミュレーション すべてのセンサタイプに対する MotionDesk をベースにした Sensor Simulation は、Release 2022-B を最後に廃止されます。詳細については、<a href="https://www.dspace.com/en/pub/home/support/pli/elas/elasw/elasensim.cfm">https://www.dspace.com/en/pub/home/support/pli/elas/elasw/elasensim.cfm</a> を参照してください。</li> <li>▪ MotionDesk MotionDesk は、Release 2023-B を最後に廃止されます。詳細については、<a href="http://www.dspace.jp/go/elamd">http://www.dspace.jp/go/elamd</a> を参照してください。</li> </ul> <p><b>AURELION</b> dSPACE AURELION は、カメラ、レーダー、および LiDAR センサを使用した ADAS/AS 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新製品で、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、<a href="https://www.dspace.com/en/pub/home/support/kb/faqs/faq433.cfm">https://www.dspace.com/en/pub/home/support/kb/faqs/faq433.cfm</a> を参照するか、dSPACE サポートにお問い合わせください。</p>
-------------------------------	---

<p><b>dSPACE ハードウェアの廃止</b></p>	<p>Release 2022-A では、次の製品のソフトウェアサポートが廃止されます。</p> <ul style="list-style-type: none"> <li>▪ Intel® Core™ i7-860 メインボード搭載の SCALEXIO プロセッサユニット (Real-Time PC Version 1.0)。 新規プロジェクトには、SCALEXIO プロセッサユニットの別のバリエーションを使用することをお勧めします。「<a href="#">SCALEXIO プロセッサユニットのバージョン</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。</li> </ul>
--------------------------------	---

<p><b>dSPACE ハードウェアの廃止予定</b></p>	<p><b>PHS バスハードウェア</b> 2021 年 12 月で、DS1006 Processor Board、DS1007 PPC Processor Board、およびすべての PHS バス I/O ボードなどの PHS バスベースシステムのハードウェアコンポーネントは廃止されました。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。dSPACE ソフトウェアの新規リリースでは、PHS バスハードウェアコンポーネントのサポートを 2023 年末まで継続します。 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。</p> <p><b>MicroAutoBox II</b> 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</p>
----------------------------------	---

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予定

dSPACE Release 2023-A 以降、dSPACE ソフトウェア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 の WibuKey、フローティングネットワークライセンス用の Flexera の FlexNet)をサポートしなくなります。これには、以下のような意味があります。

- dSPACE Release 2023-A 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 Release 2023-A 以降、dSPACE Installation Manager 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 は、dSPACE Release 2017-A 以前と同じ PC にインストールすることができません。
- 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す。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。

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

dSPACE Release 2023-A 以降、dSPACE Installation Manager では、古いドングル(dSPACE Release 2017-A 以前で配信)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。

## 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			
	2021-A	2021-B	2022-A	2022-B <sup>1)</sup> を参照してください。
AutomationDesk	6.5	6.6	6.7	2022-B <a href="#">「AutomationDesk」</a> (29 ページ)を参照。
Automotive Simulation Model	9.7	10.0	10.1	2022-B <a href="#">「Automotive Simulation Model(ASM)」</a> (33 ページ)を参照。
Bus Manager(スタンドアロン)	6.7	6.8	22.1	2022-B <a href="#">「Bus Manager(スタンドアロン)」</a> (45 ページ)を参照。
ConfigurationDesk for RapidPro	6.7	- <sup>2)</sup>	-	-
ConfigurationDesk	6.7	6.8	22.1	2022-B <a href="#">「ConfigurationDesk」</a> (51 ページ)を参照。

製品名	dSPACE Release			
	2021-A	2021-B	2022-A	2022-B <sup>1)</sup> を参照してください。
Container Manager	5.2	5.2	5.2	–
ControlDesk	7.4	7.5	7.6	2022-B 「ControlDesk」(63 ページ)を参照。
DCI Configuration Tool	3.13	3.13	3.13	2022-B 「DCI Configuration Tool」(79 ページ)を参照。
dSPACE AUTOSAR Compare	1.0	1.1	1.1	2022-B 「dSPACE AUTOSAR Compare」(75 ページ)を参照。
dSPACE CAN API Package	4.0.6	4.0.7	4.0.8	2022-B
dSPACE ECU Flash Programming Tool	2.8	2.8	2.8	2022-B
dSPACE FlexRay Configuration Package	4.7	4.8	4.9	2022-B 「dSPACE FlexRay Configuration Package」(81 ページ)を参照。
dSPACE Installation Manager	5.7	5.8	5.9	22.2 「dSPACE Installation Manager」(83 ページ)を参照。
dSPACE Python Extensions	4.0	4.1	4.2	2022-B 「dSPACE Python Extensions」(85 ページ)を参照。
dSPACE XIL API .NET	2021-A	2021-B	2022-A	2022-B 「dSPACE XIL API .NET」(87 ページ)を参照。
ECU Interface Manager	2.9	2.10	2.11	2022-B 「ECU Interface Manager」(91 ページ)を参照。
Firmware Manager	3.3	3.4	3.5	22.2 「Firmware Manager」(95 ページ)を参照。
FPGA Programming Blockset <sup>3)</sup>	3.11	3.12	3.13	2022-B 「FPGA Programming Blockset」(97 ページ)を参照。
MicroAutoBox III のファームウェア	5.1	5.2	6.0	22.2 「MicroAutoBox III のファームウェア」(103 ページ)を参照。
Model and Sensor Interface Blockset	1.1	1.2	1.3	2022-B 「Model および Sensor Interface Blockset」(111 ページ)を参照。
Model Compare	3.1	3.2	3.2	2022-B 「Model Compare」(105 ページ)を参照。
Model Container Utility	–	–	–	2022-B 「モデルコンテナユーティリティについて」(109 ページ)を参照。
ModelDesk	5.5	5.6	5.7	2022-B 「ModelDesk」(115 ページ)を参照。

製品名	dSPACE Release			
	2021-A	2021-B	2022-A	2022-B <sup>1)</sup> を参照してください。
Model Interface Package for Simulink	4.5	4.6	22.1	2022-B 「 <a href="#">Model Interface Package for Simulink</a> 」(117 ページ)を参照。
MotionDesk	4.8	4.9	4.10	2022-B 「 <a href="#">MotionDesk</a> 」(119 ページ)を参照。
MotionDesk Blockset	2.6.3	2.6.4	2.6.5	2022-B 「 <a href="#">MotionDesk</a> 」(119 ページ)を参照。
Real-Time Testing	5.0	5.1	5.2	2022-B 「 <a href="#">Real-Time Testing</a> 」(121 ページ)を参照。
RTI <sup>4)</sup>	7.16	7.17	7.18	2022-B 「 <a href="#">RTI/RTI-MP および RTLib</a> 」(123 ページ)を参照。
RTI-MP <sup>5)</sup>	7.16	7.17	7.18	2022-B 「 <a href="#">RTI/RTI-MP および RTLib</a> 」(123 ページ)を参照。
RTI Bypass Blockset	3.16	3.17	3.18	2022-B 「 <a href="#">RTI Bypass Blockset</a> 」(125 ページ)を参照。
RTI CAN Blockset	3.4.12	3.4.13	3.4.14	2022-B
RTI CAN MultiMessage Blockset	5.6	5.7	5.8	2022-B 「 <a href="#">RTI CAN MultiMessage Blockset</a> 」(127 ページ)を参照。
RTI Electric Motor Control Blockset	1.4.3	1.4.4	1.4.5	2022-B
RTI Ethernet Blockset	1.2.5	1.2.6	1.2.7	2022-B
RTI Ethernet (UDP) Blockset	1.4.5	1.4.6	1.4.7	2022-B
RTI LIN MultiMessage Blockset	3.6	3.7	3.8	2022-B 「 <a href="#">RTI LIN MultiMessage Blockset</a> 」(129 ページ)を参照。
RTI RapidPro Control Unit Blockset	2.2.5	2.2.6	2.2.7	2022-B
RTI Synchronized Time Base Manager Blockset	1.4.2	1.4.3	1.4.4	2022-B 「 <a href="#">RTI Synchronized Time Base Manager Blockset</a> 」(131 ページ)を参照。
RTI USB Flight Recorder Blockset	1.2.4	1.2.5	1.2.6	2022-B
RTI Watchdog Blockset	2.1.3	2.1.4	2.1.5	2022-B
センサシミュレーション	1.5	1.6	1.7	2022-B 「 <a href="#">センサシミュレーション</a> 」(135 ページ)を参照。
SCALEXIO Firmware	5.1	5.2	6.0	2022-B 「 <a href="#">SCALEXIO Firmware</a> 」(133 ページ)を参照。
SYNECT	2.11	2.12	2.13	2022-B 「 <a href="#">SYNECT</a> 」(137 ページ)を参照。
SystemDesk	5.5	5.6	5.6	2022-B 「 <a href="#">SystemDesk</a> 」(145 ページ)を参照。

製品名	dSPACE Release			
	2021-A	2021-B	2022-A	2022-B <sup>1)</sup> を参照してください。
TargetLink	5.1	5.2	5.2	2022-B 「TargetLink」(153 ページ)を参照。
VEOS	5.2	5.3	5.4	2022-B 「VEOS」(225 ページ)を参照。

1) dSPACE 製品の新しいバージョン管理方式については、「一般的な機能拡張および変更」(13 ページ)

2) dSPACE Release 2021-B 以降、ConfigurationDesk for RapidPro は dSPACE Release に含まれなくなりました。dSPACE の Web サイトから、専用のセットアッププログラムを含む個別のソフトウェアパッケージとしてダウンロードできます。<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。

- ユーザドキュメントの拡張

新機能の詳細については、「AutomationDesk 2022-B の新機能」(29 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。

- 多重化 IPDU の送信をその静的部分 IPDU でトリガ
- ISignal グループのアップデートビット用 ISignal の生成
- バス設定テーブルの拡張

詳細については、「Bus Manager(スタンドアロン) 2022-B の新機能」(45 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。

- 新しい DS6330M3 Automotive Ethernet Module のサポート。
- HIL 環境でデータプレイのステータス情報を提供するようにファンクションブロックタイプを拡張。
- linux64/linux32 共有オブジェクトでのバイナリ FMU のサポート。

- FMI 2.0 規格に準拠したバイナリフォルダの拡張サポート
- 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](#)」(51 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

- [Variables]ペインと Measurement Data Pool のナビゲーションの改善

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 \(ControlDesk 2022-B\)](#)」(64 ページ)を参照してください。

### プラットフォーム/デバイスの機能強化

- cmdloader: プラットフォームのフラッシュメモリを完全に消去

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 \(ControlDesk 2022-B\)](#)」(65 ページ)を参照してください。

### 変数管理の強化

- 変数の URI (Uniform Resource Identifier)

詳細については、「[新しい変数管理機能 \(ControlDesk 2022-B\)](#)」(65 ページ)を参照してください。

### 計測および記録の強化

- データロギングの改善

詳細については、「[新しい計測および記録機能 \(ControlDesk 2022-B\)](#)」(66 ページ)を参照してください。

### Bus Navigator の機能強化

- Bus Navigator (スタンドアロン)
- Bus Navigator ユーザインターフェースの改善
- R21-11 ARXML ファイルのサポート

詳細については、「[新しい Bus Navigator 機能 \(ControlDesk 2022-B\)](#)」(67 ページ)を参照してください。

### 自動化の強化

- Table Editor に接続された変数の個々の関数と軸の値の書き込み
- レイアウトのデータソースの切り替え
- 計算変数の出力パラメータの指定

詳細については、「[新しい自動化機能 \(ControlDesk 2022-B\)](#)」(70 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。

- ソース切り替え変数のデフォルト値の定義。

新機能の詳細については、「[dSPACE FlexRay Configuration Package 2022-B の新機能](#)」(81 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。

- ASAM XIL API 規格で定義されている ECUPort (ECUCPort と ECUMPort の組み合わせ)での初めての機能のサポート
- MAPort および ECUPort の実装で ASAM XIL API 2.2.0 に準拠して VariableInfo インターフェースを強化

新機能の詳細については、「[dSPACE XIL API .NET 2022-B の新機能](#)」(87 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。

- ECU 変数への動的アクセスの準備
- XCP on CAN: 特定の CAN ID の割り当て
- XCP on CAN: PID 送信の無効化

新機能の詳細については、「[ECU Interface Manager 2022-B の新機能](#)」(91 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。

- ファームウェアアップデート後の自動再起動のサポート強化

詳細については、「[Firmware Manager 22.2 の新機能](#)」(95 ページ)を参照してください。

---

**FPGA Programming Blockset**

FPGA Programming Blockset 2022-B の主な新機能は、次のとおりです。

- Xilinx®ソフトウェアのサポートの拡張。
- プロセッサ信号の前処理および後処理を行うための SCALEXIO および MicroAutoBox III 用スケールリングサブシステムのサポート。
- SCALEXIO と MicroAutoBox III のビルド結果用の新しいコンテナファイル形式。

新機能の詳細については、「[FPGA Programming Blockset 2022-B の新機能](#)」(97 ページ)を参照してください。

---

**MicroAutoBox III のファームウェア**

MicroAutoBox III のファームウェアの主な新機能は次のとおりです。

- DS1403 Processor Board でデータロギング設定を有効にしたリアルタイムアプリケーションの使用をサポート。
- DS1521 Bus Board で RS485 全二重モードをサポート。

新機能の詳細については、「[MicroAutoBox III Firmware 22.2 の新機能](#)」(103 ページ)を参照してください。

---

**Model and Sensor Interface Blockset**

Model and Sensor Interface Blockset の主な新機能は次のとおりです。

- Software-in-the-Loop シミュレーションのための VEOS の TCP サーバ接続。
- 通常モードおよび加速モードでの Simulink シミュレーションのサポート。

- ASM Traffic モデルデモに Model and Sensor Interface Blockset の Animation Interface を含めて、AURELION を使用したアニメーションとセンサシミュレーションをサポート。

ASM Traffic デモモデルの新機能の詳細については、「[ASM Traffic デモモデルの変更](#)」(41 ページ)を参照してください。

ブロックセットの新機能の詳細については、「[Model and Sensor Interface Blockset 2022-B の新機能](#)」(111 ページ)を参照してください。

---

## Model Compare

Model Compare 2022-B では、いくつかの改善が行われています。

新機能の詳細については、「[Model Compare 2022-B の新機能](#)」(105 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。

- Road Generator:トラフィックオブジェクトの位置を絶対座標で指定。

新機能の詳細については、「[ModelDesk 2022-B の新機能](#)」(115 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。

- 設定したバス通信の Simulink モデルへの伝播に関する機能強化
- MATLAB® R2022b のサポート

新機能の詳細については、「[Model Interface Package for Simulink](#)」(117 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® R2022b のサポート。

詳細については、「[RTI/RTI-MP および RTLib の新機能](#)」(123 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 STBM\_SYNCHRONOUS\_TRIGGER ブロック。

新機能の詳細については、「[RTI Synchronized Time Base Manager Blockset 2022-B の新機能](#)」(131 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。

- MACsec(メディアアクセス制御セキュリティ)機能を内蔵した DS6330M4 Automotive Ethernet Module のサポート。

新機能の詳細については、「[SCALEXIO Firmware 22.2 の新機能](#)」(133 ページ)を参照してください。



---

**SYNECT**

SYNECT 2022-B の主要新機能は、次のとおりです。

- 最新のユーザインターフェースフレームワーク
- グローバル検索の改善

新機能の詳細については、「[SYNECT 2022-B の新機能](#)」(138 ページ)を参照してください。

---

**SystemDesk**

SystemDesk 2022-B の主要新機能は、次のとおりです。

- AUTOSAR R21-11 リリースのサポート
- V-ECU FMU に関する FMI 3.0 規格のサポート。
- 生成されたベーシックソフトウェアモジュールのコントラクトヘッダーファイルのエクスポート。

新機能の詳細については、「[SystemDesk 2022-B の新機能](#)」(146 ページ)を参照してください。

---

**TargetLink**

TargetLink 2022-B の主要新機能は、次のとおりです。

- n 次元補間ブロック
  - 5 次元までのルックアップテーブルをサポート。
- AUTOSAR INOUT パラメータ
  - ARGINOUT の種類のオペレーション引数をサポート。
- Bus Creator ブロックがバス配列に対応
  - さまざまな種類のバス配列信号を bus-with-array-of-bus-as-payload 信号に接続。
- Adaptive AUTOSAR データタイプ
  - コード生成で C++ データタイプをサポート。
- Stateflow の temporalCount
  - 時相論理演算子 temporalCount のサポート。
- ヘッドレス Linux のサポート (dSPACE Release から分離)
  - Linux ベースのコンテナで TargetLink を実行してコード生成。

これらの新機能の詳細については、「[TargetLink 2022-B の新機能](#)」(154 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 2022-B および TargetLink Data Dictionary 2022-B への移行](#)」(171 ページ)を参照してください。

---

**VEOS**

VEOS の主要新機能は、次のとおりです。

- FMI 3.0 リリースのサポート
- モデルコンテナユーティリティのサポート
- VEOS を使用した協調シミュレーション
- V-ECU SDK
- Linux での V-ECU および SIC のビルド (VEOS Build Console)

- SIMPHERA ワークフローのコンテキストでの ISO 26262 認証の取得
- CAN/Ethernet データリプレイ用ステータス信号

新機能の詳細については、「[VEOS 2022-B の新機能](#)」(225 ページ)を参照してください。

# 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 2022-B への移行

概要	Release 2022-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 2022-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 2021-B 以前のリリースからの移行	<p>dSPACE Release 2021-B 以前のリリースから Release 2022-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 2022-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。</p> <p>必要な移行手順の詳細については、各 dSPACE Release バージョンの『<a href="#">新機能と移行手順</a>』ドキュメントを参照してください。</p>
------------------------------------	---

以前のリリースのドキュメント	<p>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。</p> <p>以前のリリースの『<a href="#">新機能と移行手順</a>』ファイルは次の場所にあります。</p> <ul style="list-style-type: none"> <li>最新の dSPACE Help インストールフォルダの、C:\Program Files\Common Files\dSPACE\Help 2022-B\Print\PreviousReleases を参照してください。</li> </ul>
----------------	---

- 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。

# AutomationDesk

## 次のステップ

## 本章の内容

AutomationDesk 2022-B の新機能.....	29
AutomationDesk 2022-B への移行.....	29

## AutomationDesk 2022-B の新機能

### 一般的な機能強化

**ユーザドキュメントの拡張** AutomationDesk のユーザドキュメントは、アニメーショングラフィックをいくつか追加して、ユーザインターフェースでの操作をビジュアル表示するように強化されました。

## AutomationDesk 2022-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、「AutomationDesk の移行」(『AutomationDesk 概要』)を参照してください。

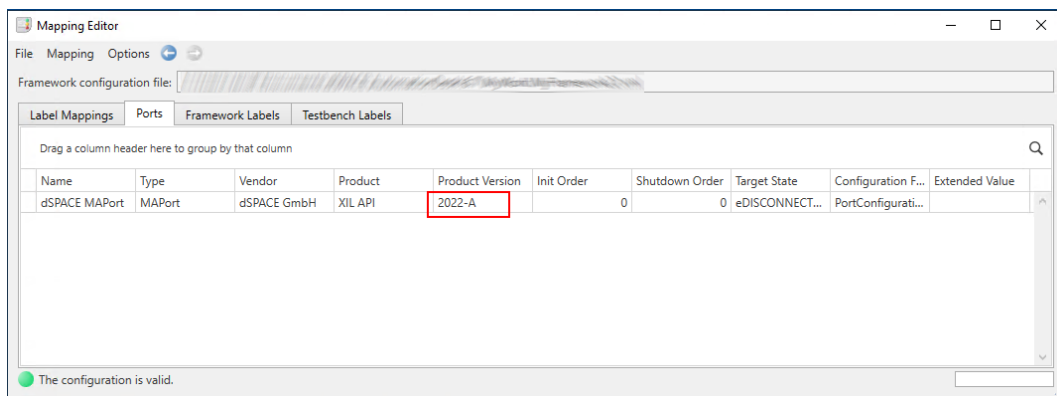
#### Release 2021-A 以降でのプロジェクトインポートの制限事項

dSPACE Release 2021-A 以降、AutomationDesk は、AutomationDesk 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。

#### XIL API 2.2.0 への移行

XIL API ライブラリや XIL API Convenience ライブラリのブロックを使用する場合、いくつかの設定を適応させる必要があります。

**XIL API フレームワークの使用** デフォルトベンダー *dSPACE* を使用していない場合、Mapping Editor でポート定義を移行する必要があります。XIL API のバージョンを 2.2.0 に切り替えるには、Release 2022-A 以降の dSPACE インプリメンテーションを指定する必要があります。



**XIL API テストベンチの使用** XIL API テストベンチを使用する場合、Vendor データオブジェクトを新しいベンダー識別子 *dSPACE GmbH\_XIL API\_2022-A* または *dSPACE GmbH\_XIL API\_2022-B* に移行する必要があります。

---

**STZ ファイルの使用**

STZ ファイルを使用する場合、手動での移行は必要ありません。ステイミュラスシグナルを実行すると、シグナルの記述が自動的に XIL API 2.2.0 に一時的に変換されます。STZ ファイルは、ステイミュラスの定義が変更された場合のみ、新しいフォーマットで保存されます。

---

**廃止の予定**

**Automation Server** *Automation Server* は、2022 年 12 月に廃止されます。ソフトウェア保守サービスをご契約のお客様に限り、2022 年 6 月まで AutomationDesk Automation Server の新しいリリースをご利用いただけます。*AutomationDesk Basic* ライセンスでは、Automation Server に相当する技術的内容が提供されます。詳細については、<http://www.dspace.jp/go/discontinuation> を参照してください。





# Automotive Simulation Model (ASM)

## 次のステップ

## 本章の内容

すべての ASM 製品.....	34
ASM Battery.....	35
ASM Electric Components.....	36
ASM Environment.....	37
ASM Fuel Cell.....	38
ASM Utils.....	40
ASM Traffic.....	41
ASM Truck.....	43
ASM Vehicle Dynamics.....	44

## すべての ASM 製品

### すべての ASM ブロックセットの新機能

---

#### 新しい ASM 動画

いくつかの動画で、ASM と ModelDesk の操作方法を紹介しています。  
自動化によるパラメータ値の変更方法を説明した新しい動画もあります。  
これらの動画には、dSPACE Help および dSPACE Web サイトからアクセスすることができます。

#### ASM の MATLAB M コードの変更

ASM 内部の `find_system` 呼び出しでは、バリエーションサブシステムのすべてのバリエーション選択が考慮されるようになりました。

ASM の M コード (モデルマイグレーション、ASM Testbench、ASMSignalInterface など) では、ブロック、ライン、ポートなどを検出するために複数の `find_system` 呼び出しが使用されます。

Release 2022-A までは、検索時にバリエーションサブシステムのアクティブなバリエーション選択肢のみが考慮されていました。

Release 2022-B 以降、検索時にバリエーションサブシステムのすべてのバリエーション選択肢が考慮されます。

この変更は、MATLAB Release R2022b 以降で `find_system` 呼び出しでのバリエーションの扱いが変更されたことに対応したものです。詳細については、対応する MATLAB ドキュメントを参照してください。

この変更は、ASM でサポートされているすべての MATLAB リリースで使用される ASM モデルに適用されます。


# ASM Battery

## ASM Battery Blockset 2022-B の新機能

---

### 新しい ASM ライブラリと新しいデモモデル

ASM Battery Blockset が新しくなりました。

ASM Battery ライブラリには、バッテリーやバッテリーパック、およびそれらの熱挙動のモデリングとシミュレーションを可能にするブロックが含まれています。また、このライブラリには、バッテリーシステムの監視と制御を行うためのブロックも含まれています。このライブラリに含まれるブロックの詳細については、『[ASM Battery Reference](#)』を参照してください。

さらに、バッテリーが安全かつ効率的に運用されるように監視するバッテリー管理システム (BMS) を開発、テストする新しいデモプロジェクトが追加されています。デモモデルの詳細については、『[ASM Battery Model Description](#)』を参照してください。

## ASM Electric Components

### ASM Electric Components デモモデルの変更

#### Battery Electric Vehicle with Traffic デモ

Battery Electric Vehicle with Traffic デモモデルが次のように変更されました。

**AURELION のサポート** Battery Electric Vehicle with Traffic デモモデルでは、アニメーションツールとして AURELION がサポートされるようになりました。AURELION を使用したアニメーションをサポートするために、デモモデルの MotionDeskInterface は AnimationInterface に置き換えられました。それぞれの MotionDesk プロジェクトは廃止されます。

**SCALEXIO プラットフォームのサポート** 新しい AnimationInterface には、SCALEXIO ハードウェア固有の設定が必要です。そのため、SCALEXIO のリアルタイムオブジェクトは、Traffic デモモデルに含まれなくなりました。その代わりに、Traffic プロジェクトに SIC ファイルと Python スクリプトが含まれており、それを使用して特定の SCALEXIO システム用のリアルタイムオブジェクトを簡単に作成できるようになりました。

**デフォルトのシミュレーションプラットフォーム** Traffic モデルでは、go.m ファイルでモデルを初期化する際に選択されるデフォルトのプラットフォームが、SCALEXIO から VEOS に変更されました。

go.m ファイルの呼び出しで Traffic モデルを初期化する際に、プラットフォーム引数を指定しない場合は、デフォルトで VEOS プラットフォームでシミュレーション用のモデルを準備するようになりました。

# ASM Environment

## 次のステップ

## 本章の内容

ASM Environment Blockset 2022-B の新機能.....	37
ASM Environment Blockset 2022-B への移行.....	37

## ASM Environment Blockset 2022-B の新機能

### MANEUVER\_FCN\_CALL\_ GENERATOR ブロック

連続した呼び出し間のサンプル数を、入力ポートを使用してオプションで指定できるようになりました。

## ASM Environment Blockset 2022-B への移行

### MANEUVER\_FCN\_CALL\_ GENERATOR ブロック

新しい入力ポートは外部サンプル数に関連付けられており、移行中に定数値に接続されます。

## 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 Fuel Cell

## 次のステップ

## 本章の内容

ASM Fuel Cell Blockset 2022-B の新機能.....	38
ASM Fuel Cell デモモデルの変更.....	38
ASM Fuel Cell Blockset 2022-B への移行.....	39

## ASM Fuel Cell Blockset 2022-B の新機能

### ELECTROCHEMISTRY\_ PHYSICAL ブロック

このブロックには、燃料電池の分極曲線を定義するパラメータを計算するための新しい ModelDesk 処理関数が含まれています。

### HYDROGEN\_EJECTOR ブロック

これはエジェクタを通過する質量流量とエンタルピー流量を計算する新しいブロックです。エジェクタは、水素の取り込みとアノード側からの水素の再循環の両方の役割を担っています。このブロックは、エジェクタの質量流量の測定値やエジェクタに影響を与えるその他の測定値に基づいて、3つのパラメータのデフォルト値を調整する処理関数も提供します。

## ASM Fuel Cell デモモデルの変更

### 再循環モードパラメータ

ASM\_FuelCell/FuelCell/FuelCell\_Slow/Plant 内の UserInterface に新しい Recirculation Mode パラメータが追加されました。

このパラメータは、2つの再循環モードのいずれかをアクティブにします。

- 1つ目のモードでは、水素の取り込みにインジェクタバルブ、水素の再循環に再循環ブローを使用します。
- 2つ目のモードでは、水素エジェクタがこの両方の役割を担います。

### 定常質量パワーバランスの計算

ASM\_FuelCell/FuelCell/FuelCell\_Slow/Plant 内の UserInterface での効率計算が変更されました。現在は、低位発熱量に基づいて1つの効率のみが計算され、BoP コンポーネント(エアコンプレッサなど)のすべての損失と電力消費も考慮されるようになりました。また、モデルでは、エジェクタモードがアクティブである場合、質量とエンタルピーのバランスの計算に、エジェクタの質量流量とエンタルピー流量が考慮されるようになりました。

---

MEA_BP_Casing_Assembly ブロック	<p>以下の平均値が(すべてのアクティブなセグメントにわたって)ブロックに追加されました。</p> <ul style="list-style-type: none"><li>▪ 膜の湿度。</li><li>▪ カソード触媒層での液体水の体積分率。</li><li>▪ アノード触媒層での水素のガス分率。</li><li>▪ カソード触媒層での酸素と水蒸気のガス分率。</li></ul>
-----------------------------	--

## ASM Fuel Cell Blockset 2022-B への移行

---

HYDROGEN SUPPLY CONTROL ブロック	<p>モデルにエジェクタモデルが追加されたため、ブロックマイグレーションが実行されます。このブロックに、現在の再循環モード用の入力と、エジェクタ制御信号用の出力が追加されました。また、エジェクタコントローラ用の 2 つの新しいパラメータが含まれるようになりました。</p>
------------------------------	--

---

COOLANT_PUMP ブロックと FAN ブロック	<p>COOLANT_PUMP ブロックと FAN ブロックに体積流量で電力を計算する新しいパラメータが追加されたため、ブロックマイグレーションが実行されます。</p>
-----------------------------	--

### 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 Utils

### 次のステップ

### 本章の内容

ASM Utils 2022-B の新機能.....	40
ASM Utils 2022-B への移行.....	40

## ASM Utils 2022-B の新機能

ASM_FCNCALL_GENERATOR ブロック	連続した呼び出し間のサンプル数を、入力ポートを使用してオプションで指定できるようになりました。
-------------------------------	---

## ASM Utils 2022-B への移行

ASM_FCNCALL_GENERATOR ブロック	新しい入力ポートは外部サンプル数に関連付けられており、移行中に定数値に接続されます。
-------------------------------	--



# ASM Traffic

## 次のステップ

## 本章の内容

ASM Traffic デモモデルの変更.....	41
ASM Traffic Blockset 2022-B への移行.....	42

## ASM Traffic デモモデルの変更

### 2-D SensorParameters ブロックの拡張

2-D センサの SensorParameters ブロックは、DISCRETE\_OBJECTS ブロックの計算が不要なパラメータ設定のケースを決定するロジックで拡張されています。DISCRETE\_OBJECTS ブロックを有効または無効にするスイッチ信号が用意されています。

### AURELION のサポート

Vehicle Dynamics with Traffic デモモデルは、アニメーションツールとして AURELION をサポートするようになりました。

AURELION を使用したアニメーションをサポートするために、デモモデルの MotionDeskInterface は AnimationInterface に置き換えられました。それぞれの MotionDesk プロジェクトは廃止されます。

### SCALEXIO プラットフォームのサポート

新しい AnimationInterface には、SCALEXIO ハードウェア固有の設定が必要です。そのため、SCALEXIO のリアルタイムオブジェクトは、Traffic デモモデルに含まれなくなりました。その代わりに、Traffic プロジェクトに SIC ファイルと Python スクリプトが含まれており、それを使用して特定の SCALEXIO システム用のリアルタイムオブジェクトを簡単に作成できるようになりました。

### デフォルトのシミュレーションプラットフォーム

Traffic モデルでは、go.m ファイルでモデルを初期化する際に選択されるデフォルトのプラットフォームが、SCALEXIO から VEOS に変更されました。

go.m ファイルの呼び出しで Traffic モデルを初期化する際に、プラットフォーム引数を指定しない場合は、デフォルトで VEOS プラットフォームでシミュレーション用のモデルを準備するようになりました。

## ASM Traffic Blockset 2022-B への移行

### DISCRETE\_OBJECTS ブロック

このブロックは、DISCRETE\_OBJECTS S-function の計算を有効または無効にするために使用できる新しい入力ポートによって拡張されています。移行時に、このブロックを有効にするために新しい入力ポートが定数値 1 に接続されます。

### 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 Truck

## ASM Truck デモモデルの変更

---

### ピットマンアームステアリング

ASM Truck デモでは、ステアリングシステムに新しい STEERING\_PITMAN\_ARM ブロックを使用するようになりました。ステアリングシステムは、トラック部門で多く採用されているピットマンアームステアリングをベースとするように変更されました。

## ASM Vehicle Dynamics

### ASM Vehicle Dynamics Blockset 2022-B の新機能

---

#### STEERING\_PITMAN\_ARM ブロック

新しい STEERING\_PITMAN\_ARM ブロックは、ピットマンアームを使用するステアリングシステムをモデリングします。ステアリングホイールからサスペンションとの結合まで、ステアリングシステムの変換と動的挙動をシミュレートします。内部および外部のパワーステアリングシステムを考慮することができます。

# 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) 2022-B の新機能.....	45
Bus Manager (スタンドアロン) 2022-B への移行.....	47
Bus Manager (スタンドアロン) の廃止.....	49

## Bus Manager (スタンドアロン) 2022-B の新機能

### 多重化 IPDU の送信を、静的部分 IPDU でトリガ

Bus Manager (スタンドアロン) では、多重化 IPDU の送信をその静的部分 IPDU によってトリガできるようになりました。このために、静的部分 IPDU の通信マトリクスで指定されているタイミングを考慮することができます。また、PDU Cyclic Timing Control および PDU Trigger 機能を静的部分 IPDU に割り当てて、その送信と関連する多重化 IPDU の送信をトリガすることもできます。

詳細については、「[多重化 IPDU の送信のトリガ](#)」(『Bus Manager (スタンドアロン) 実装ガイド』[📖](#))を参照してください。

### ISignal グループのアップデートビット用 ISignal の生成

Bus Manager (スタンドアロン) は、ISignal グループのアップデートビット用 ISignal を自動生成するようになりました。通信マトリクスでこのようなアップデートビットが指定されている場合、通信マトリクスのインポート時に ISignals が生成されます。

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。

## バス設定テーブルの拡張

Bus Manager(スタンドアロン)では、バス設定テーブルが次のように拡張されました。

- 追加の列を使用できるようになりました。これらの列を使用して、フレームトリガ、ISignal の初期値、PDU User Code 機能のユーザ信号など、さらに多くのエレメントを表示することができます。
- 見やすくするために、デフォルトでは利用可能なすべての列は表示されません。テーブルに表示されていない列は、Column Chooser で使用可能になっています。Column Chooser から、ドラッグ & ドロップでこれらの列をテーブルに追加することができます。
- テーブルをデフォルトの状態にリセットするには、[Reset Columns]コンテキストメニューコマンドを使用します。
- 各列を明確に識別するため、一部の列は名前が変更されています。
- テーブルのヘッダーでは、各列についてのツールチップを表示することができます。ツールチップには、各列に関する簡単な情報と、ユーザドキュメントの詳細情報への参照が表示されます(該当する場合)。

詳細については、「[バス設定テーブル](#)」(『[Bus Manager\(スタンドアロン\)実装ガイド](#)』[📖](#))を参照してください。

## 実行アプリケーションでのグローバル時間同期(GTS)の実装に関する機能強化

Bus Manager(スタンドアロン)では、グローバル時間に同期して個々の PDU の送信をトリガできるようになりました。これを行うには、RTI Synchronized Time Base Manager Blockset で提供されるトリガを使用します。

詳細については、「[実行アプリケーションでのグローバル時間同期の実装の基礎](#)」(『[Bus Manager\(スタンドアロン\)実装ガイド](#)』[📖](#))を参照してください。

## SecOC Authenticator Invalidation 機能の強化

SecOC Authenticator Invalidation 機能をセキュア IPDU に追加する際に、[Recalculate SecOC Information]チェックボックスがデフォルトでオフになりました。このため、ほとんどのユースケースで必要となる認証情報の再計算は、デフォルトで無効になります。

## ConfigurationDesk の自動化インターフェースを介した親エレメントへのアクセス

Buses Browser またはバス設定テーブルで選択されているエレメントの親エレメントに、ConfigurationDesk の自動化インターフェースを介してアクセスできるようになりました。これを行うには、ICaRelation インターフェースの新しい BusParentElement 関係を使用します。

詳細については、「[Release 2022-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。


## 最新のユーザインターフェース

Bus Manager(スタンドアロン)のユーザインターフェースのスタイルが最新化されました。これは Bus Manager の使用方法や機能には影響しません。

## 設定したバス通信の Simulink モデルへの伝播に関する機能強化

Model Interface Package for Simulink の機能強化により、設定したバス通信を Simulink モデルに伝播する際に、バス設定エレメントの Simulink.Bus オブジェクトを作成できるようになりました。これを行うには、MATLAB コマンドウィンドウで次のコマンドを入力します。

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

これで、設定したバス通信を Simulink モデルに伝播する際に、モデルポートブロックと Simulink.Bus オブジェクトが作成されます。Simulink.Bus オブジェクトは MATLAB ワークスペースに作成され、関連するモデルポートブロックのポートにデータタイプとして割り当てられます。詳細については、「[構造化されたデータポートを持つデータポートブロックの基礎](#)」(『Model Interface Package for Simulink - モデリングガイド』)を参照してください。

## Bus Manager(スタンドアロン) 2022-B への移行

---

以前のバージョンの Bus Manager(スタンドアロン)で作成したプロジェクトを開く

現在の Bus Manager(スタンドアロン)バージョンのプロジェクトと同じように、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を開くことができます

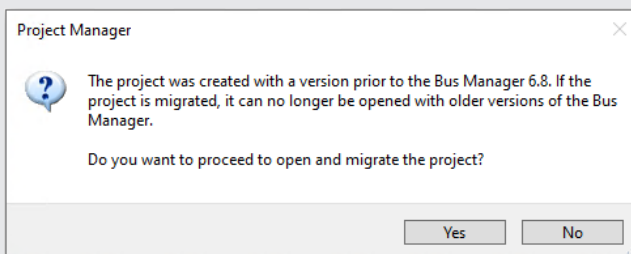
す。Bus Manager(スタンドアロン)では、プロジェクトを開く前にプロジェクトを移行します。

#### 注記

- dSPACE Release 2021-A 以降、Bus Manager(スタンドアロン)は、Bus Manager(スタンドアロン)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。
- プロジェクトを移行して保存すると、以前の Bus Manager(スタンドアロン)では開けなくなります。

Bus Manager(スタンドアロン)6.8(dSPACE Release 2021-B)で変更されたプロジェクトやアプリケーションのファイル形式について、以下の点にご注意ください。



- 以前のバージョンの Bus Manager(スタンドアロン)で作成したプロジェクトやアプリケーションファイル、バックアップも開くことができます。Bus Manager(スタンドアロン)では、プロジェクトを開く際に移行を確認するメッセージが表示されます。



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(スタンドアロン)では開けなくなります。

- 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 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。

Bus Manager(スタンドアロン)22.1(dSPACE Release 2022-A)で導入された新しいプロジェクト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プロジェクト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。これを回避するために、プロジェクトキャッシュフォルダのパスを短くすることができます。「[\[Paths\]ページ](#)」(『[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。



---


生成された ISignal グループのアップデートビット用 ISignal を既存の ConfigurationDesk アプリケーションで使用できない

Bus Manager によって生成される ISignal グループのアップデートビット用 ISignal は、Bus Manager(スタンドアロン)22.1 以前のバージョンで作成された ConfigurationDesk アプリケーションでは使用することができません。このような ISignal を使用できるようにするには、関連する通信マトリクス別のバージョン、つまり Content hash 値が異なる通信マトリクスを、影響する ConfigurationDesk アプリケーションに追加する必要があります。その後、関連するバス設定で古い通信マトリクスを置き換えて、必要に応じて新しく使用可能になった ISignal を手動で割り当てます。

通信マトリクスの置き換えの詳細については、「[割り当てられた通信マトリクスの交換](#)」(『Bus Manager(スタンドアロン)実装ガイド』)を参照してください。

---

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 2022-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)の廃止

---

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

アクティブな ConfigurationDesk アプリケーションのコンフィギュレーションデータを 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、dSPACE Release 2023-B で削除されます。

対応するツール自動化 API コマンド ICaAlgorithms:ExportConfiguration も削除されます。



# ConfigurationDesk

**概要** ConfigurationDesk では、SCALEXIO ハードウェアまたは MicroAutoBox III 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 2022-B の新しい一般機能.....	51
I/O 機能に関する新機能とハードウェアサポート.....	54
ConfigurationDesk の Bus Manager の新機能.....	55
サポートされるコンテナファイルバージョン.....	57
ConfigurationDesk 2022-B への移行.....	60
ConfigurationDesk の廃止.....	62

## ConfigurationDesk 2022-B の新しい一般機能

### Ethernet BSC ファイルのサポート

ConfigurationDesk では、Ethernet Configuration Package で生成されたバスシミュレーションコンテナ (BSC ファイル) がサポートされるようになりました。これにより、次のようなモデルコンテナワークフローの利点を、Ethernet 通信を使用するリアルタイムアプリケーションでも利用できるようになります。

- モデリングとコンフィギュレーションの作業を、統合、シミュレーション、テストから分離。
- オプションで ConfigurationDesk の PrecompileBSC 自動化コマンドでプリコンパイルできるモデルコンテナを使用することで、ビルド時間を短縮。
- ソースコードを含まないコンパイル済みコンテナファイルの使用による知的財産保護。

詳細については、「[バスシミュレーションコンテナ \(BSC ファイル\) の使用](#)」(『ConfigurationDesk リアルタイム実装ガイド』)を参照してください。

## FMU サポートの強化

### linux64/linux32 共有オブジェクトを使用したバイナリ FMU のサポート

ConfigurationDesk では、Linux 64/Linux 32 共有オブジェクトとして実装できるバイナリ FMU がサポートされるようになりました。これには、次の利点があります。

- バイナリ形式の共有オブジェクトでエクスポートされた、ソースコードを含まない FMU を、ConfigurationDesk で直接使用することができます。これでハイパーバイザベースのソリューションは必要なくなります。
- 異なる Linux ベースのシミュレータで同じ FMU を使用することができます。
- 知的財産保護シナリオをより簡単に実現することができます。

**バイナリフォルダのサポートの拡張** ConfigurationDesk では、ソースコードと追加のバイナリを含む FMU のサポートが拡張され、FMI 2.0 規格の定義に準拠した以下のフォルダをサポートするようになりました。


- <FMU root>/binaries/linux64
- <FMU root>/binaries/linux32

これにより、FMU の移植性が向上します。

**マルチプロセッサコアを採用した FMU のサポート** ConfigurationDesk では、複数のプロセッサコアに複数の並列スレッドを作成してモデル計算を並列化する FMU がサポートされるようになりました。新しい[Number of required processor cores]プロパティにより、アプリケーションプロセス用に予約して FMU を割り当てるプロセッサコアの数を指定することができます。

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。

## ハンドコードした FMU を使用する新しいデモ

ConfigurationDesk には、ハンドコードした FMU を使用する新しいデモが用意されています。FMU のソースコードと modelDescription.xml ファイルには、FMI 規格への対応をより簡単に開始できるようにするためのコメントが含まれています。この FMU を、Simulink®を使用せずにカスタム C コードを ConfigurationDesk のリアルタイムアプリケーションに統合するための出発点として使用することができます。詳細については、「[CfgCustomFMUDemo プロジェクト: 初めての FMI 規格の使用](#)」([『ConfigurationDesk デモプロジェクトの使用』](#) )を参照してください。

## 新しいモデル実装プロパティ

モデルコンテナ (FMU、SIC ファイル、BSC ファイル) は、インターフェース機能を C ソースコードファイルやバイナリファイルとして提供することができます。さらに、モデルコンテナには、追加のバイナリを含めることができます。ConfigurationDesk では、モデルコンテナの内容に関する情報を表示するために、以下のモデル実装プロパティが追加されました。

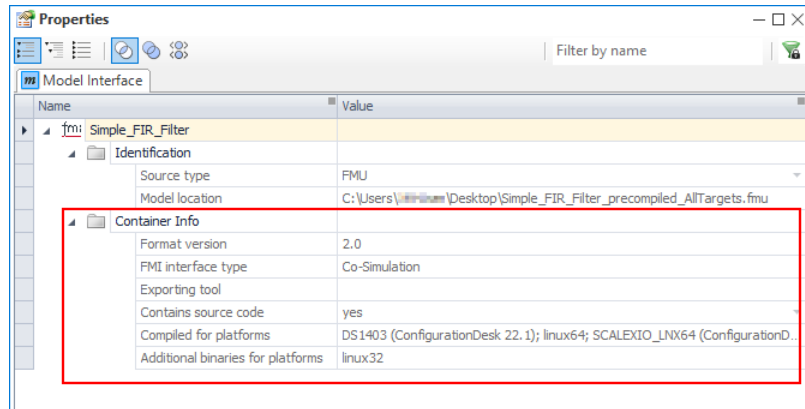
**Contains source code** [Contains source code]プロパティは、FMU、BSC ファイル、SIC ファイルにソースコードファイルが含まれているかどうかを示します。

**Compiled for platforms** [Compiled for platforms]プロパティは、モデルコンテナをコンパイルしたプラットフォームを表示します。PrecompileFMU、PrecompileSIC、または PrecompileBSC コマンドを使用してコンパイルしたモデルコンテナファイルの場合、このプロパティは FMU、BSC ファイル、または SIC ファイルをコンパイルした ConfigurationDesk バージョンをリストします。

**Additional binaries for platform** FMU、SIC ファイル、BSC ファイルの [Additional binaries for platform]プロパティは、FMU、SIC ファイル、BSC ファイ

ルに含まれる追加のバイナリと互換性のあるターゲットプラットフォームをリストします。

次の図に、FMU のプロパティの例を示します。



#### 注記

モデルコンテナの以前の[Precompiled for]プロパティは、[Compiled for platforms]に名前が変更されました。

#### アプリケーションプロセスのビルド時間の表示

ConfigurationDesk の Build Log Viewer に各アプリケーションプロセスのビルド時間とビルドプロセスの状態(成功または失敗)が表示されるようになりました。これにより、ビルド時間が長いモデル実装を特定することができます。

#### Build Log Viewer での特定フォルダへのハイパーリンク

ConfigurationDesk では、特定のフォルダに移動しやすいように、ビルドプロセス中に Build Log Viewer に Simulink コード生成フォルダ、Compile フォルダ、Build Results フォルダへのハイパーリンクを表示するようになりました。

#### ECU インターフェース処理のサポートに関する新機能

ConfigurationDesk では、リアルタイムアプリケーションに動的データアクセスを実装できるようになりました。動的データアクセスにより、リアルタイムアプリケーションがアクセスする ECU 変数を実行時に指定することができます。動的データアクセスは、ECU Interface Manager で指定する必要があります。動的データアクセスを含む EIC ファイルを ECU Interface Configuration ファンクションブロックにインポートすると、各動的データアクセスに対してファンクションポートが使用可能になります。このファンクションポートを使用して、リアルタイムアプリケーションでアクセスする ECU 変数を指定することができます。

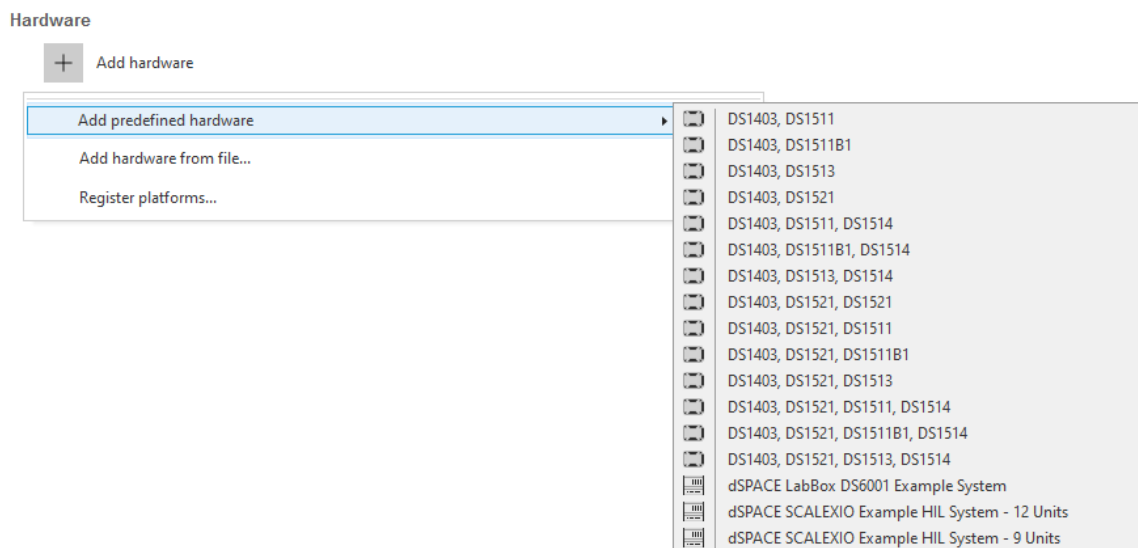
詳細については、「[ポートおよび基本プロパティの概要 \(ECU Interface Configuration\)](#)」(『[ConfigurationDesk I/O ファンクションインプリメンテーションガイド](#)』[📖](#))を参照してください。

#### ユーザインターフェースの最新化

ConfigurationDesk のグラフィカルユーザインターフェースのスタイルが最新化されました。これは、ConfigurationDesk の使用方法や機能には影響しません。

## リストから設定済みハードウェアをインポート

ConfigurationDesk アプリケーションにハードウェアをインポートする際に、複数の定義済みハードウェアトポロジのいずれかをリストから選択できるようになりました。



## 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 2022-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

### 拡張されたファンクションブロックタイプ

**データストリーミングのサポートの強化** Ethernet Setup および CAN ファンクションブロックタイプは、割り当てられたデータストリーミングチャンネルに関するステータス情報をファンクションポートに提供できるようになりました。

ファンクションポートは以下の情報を提供します。

- データストリーミングチャンネルの状態。
- 割り当てられたコントローラに転送されたデータフレームの数。
- 無効なコンテンツのために割り当てられたコントローラに転送されなかったデータフレームの数。
- 割り当てられたコントローラへの送信が失敗したデータフレームの数。
- 停止時間に達した後にドロップされたデータフレームの数。

ファンクションポートは[Channel status]プロパティで有効にすることができます。

**SENT Out** SENT Out ファンクションブロックタイプは、SENT プロトコルが使用されていない場合に送信される SENT メッセージ用の CRC チェックサムの出力をサポートできるようになりました。CRC Value ファンクション出力ポートが、実行時に計算される CRC チェックサムを提供します。

詳細については、「[ビヘイビアモデルでの SENT メッセージのエンコード](#)」(『[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。

#### ハードウェアサポートに関する新機能

**SCALEXIO** Ethernet Switch ファンクションブロックタイプは、DS6330M3 Automotive Ethernet Module をサポートするようになりました。DS6330M3 モジュールは、10BASE-T1S Ethernet 規格に対応しています。10BASE-T1S 規格は、物理層衝突回避(PLCA)をサポートする 2 線式の車載 Ethernet ネットワークで、1 本のバスラインで複数のノードを接続することが可能です。

詳細については、「[I/O Ethernet ポートの PHY 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。

## ConfigurationDesk の Bus Manager の新機能

#### 多重化 IPDU の送信を、静的部分 IPDU でトリガ

Bus Manager では、多重化 IPDU の送信を静的部分 IPDU によってトリガできるようになりました。このために、静的部分 IPDU の通信マトリクスで指定されているタイミングを考慮することができます。また、PDU Cyclic Timing Control および PDU Trigger 機能を静的部分 IPDU に割り当てて、その送信と関連する多重化 IPDU の送信をトリガすることもできます。

詳細については、「[多重化 IPDU の送信のトリガ](#)」(『[ConfigurationDesk Bus Manager 実装ガイド](#)』[📖](#))を参照してください。

#### ISignal グループのアップデートビット用 ISignal の生成

Bus Manager は、ISignal グループのアップデートビット用 ISignal を自動生成するようになりました。通信マトリクスでこのようなアップデートビットが指定されている場合、通信マトリクスのインポート時に ISignals が生成されます。

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

#### バス設定テーブルの拡張

Bus Manager では、バス設定テーブルが次のように拡張されました。

- 追加の列を使用できるようになりました。これにより、フレームトリガ、ISignal の初期値、PDU User Code 機能のユーザ信号など、さらに多くのエレメントにアクセスすることができます。
- 見やすくするために、デフォルトでは利用可能なすべての列は表示されません。テーブルに表示されていない列は、Column Chooser で使用可能になっています。Column Chooser から、ドラッグ & ドロップでこれらの列をテーブルに追加することができます。
- テーブルをデフォルト状態にリセットするには、[Reset Columns]コンテキストメニューコマンドを使用します。
- 各列を明確に識別するために、一部の列は名前が変更されています。
- 各列のテーブルヘッダーで、ツールチップを表示することができます。ツールチップには、各列に関する簡単な情報と、ユーザドキュメントの詳細情報への参照(該当する場合)が表示されます。

詳細については、「[バス設定テーブル](#)」(『[ConfigurationDesk Bus Manager 実装ガイド](#)』[📖](#))を参照してください。

### 実行アプリケーションでのグローバル時間同期(GTS)の実装に関する機能強化

Bus Manager では、グローバル時間に同期して個々の PDU の送信をトリガできるようになりました。これを行うには、RTI Synchronized Time Base Manager Blockset で提供されるトリガを使用します。

詳細については、「[実行アプリケーションでのグローバル時間同期の実装の基礎](#)」(『[ConfigurationDesk Bus Manager 実装ガイド](#)』[📖](#))を参照してください。

### 一般的な機能強化

Bus Manager では、以下の一般的な機能強化が行われています。

- SecOC Authenticator Invalidation 機能をセキュア IPDU に追加すると、[Recalculate SecOC Information]チェックボックスはデフォルトでオフになります。このため、ほとんどのユースケースで必要となる認証情報の再計算は、デフォルトで無効になります。
- Buses ビューセットで、Hardware Resource Browser がデフォルトで使用できるようになりました。

### ConfigurationDesk の自動化インターフェースを介した親エレメントへのアクセス

Buses Browser またはバス設定テーブルで選択されているエレメントの親エレメントに、ConfigurationDesk の自動化インターフェースを介してアクセスできるようになりました。これを行うには、ICaRelation インターフェースの新しい BusParentElement 関係を使用します。

詳細については、「[Release 2022-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。

### 設定したバス通信の Simulink モデルへの伝播に関する機能強化

Model Interface Package for Simulink の機能強化により、設定したバス通信を Simulink モデルに伝播する際に、バス設定エレメントの Simulink.Bus オブジェクトを作成できるようになりました。これを行うには、MATLAB コマンドウインドウで次のコマンドを入力します。

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

これで、設定したバス通信を Simulink モデルに伝播する際に、モデルポートブロックと Simulink.Bus オブジェクトが作成されます。Simulink.Bus オブジェクトは MATLAB ワークスペースに作成され、関連するモデルポートブロックのポートにデータタイプとして割り当てられます。詳細については、「[構造化されたデータポートを持つデータポートブロックの基礎](#)」(『[Model Interface Package for Simulink - モデリングガイド](#)』[📖](#))を参照してください。




## 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 2022-B(22.2) 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 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>	R2022b、R2022a、R2021b、R2021a
dSPACE Release 2022-A: <ul style="list-style-type: none"> <li>Model Interface Package for Simulink 22.1</li> </ul>	R2022a、R2021b、R2021a、R2020b
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>	R2021b、R2021a、R2020b、R2020a
dSPACE Release 2021-A: <ul style="list-style-type: none"> <li>Model Interface Package for Simulink 4.5</li> </ul>	R2021a、R2020b、R2020a、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
dSPACE Release 2019-A: <ul style="list-style-type: none"> <li>Model Interface Package for Simulink 4.1</li> </ul>	R2017b
dSPACE Release 2018-B: <ul style="list-style-type: none"> <li>Model Interface Package for Simulink 4.0</li> </ul>	R2017a
dSPACE Release 2018-A: <ul style="list-style-type: none"> <li>Model Interface Package for Simulink 3.6</li> </ul>	R2016b
dSPACE Release 2017-B: <ul style="list-style-type: none"> <li>Model Interface Package for Simulink 3.5</li> </ul>	R2016a
dSPACE Release 2017-A: <ul style="list-style-type: none"> <li>Model Interface Package for Simulink 3.4</li> </ul>	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>SCALEXIO Linux 32 ビット(レガシー対応)</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> 』  )を参照してください。

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 2021-B 以前	dsrt64.tlc(dSPACE Release 2019-B 以降で使用可能)	SCALEXIO Linux 64 ビット
	dsrt.tlc	<ul style="list-style-type: none"> <li>▪ SCALEXIO 32 ビット(レガシー対応)</li> <li>▪ MicroAutoBox III</li> </ul>

**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 2022-B(22.2)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 1.11)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 Functional Mock-Up Unit(FMU)

ConfigurationDesk 2022-B(22.2)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

## サポートされる EIC ファイルバージョン

ConfigurationDesk 2022-B(22.2)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 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
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 システムはでは、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 2022-B への移行

以前の ConfigurationDesk バージョンで作成したプロジェクトを開く

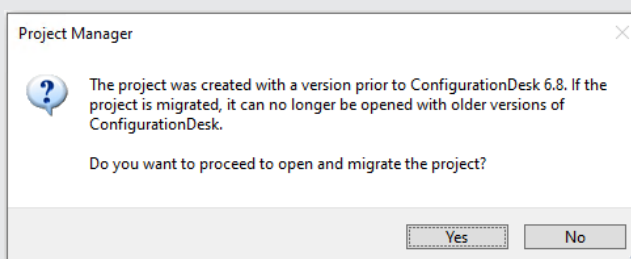
現在の ConfigurationDesk バージョンのプロジェクトと同じように、以前の ConfigurationDesk バージョンで作成したプロジェクトを開くことができます。プロジェクトは、開くと自動的に移行されます。

### 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。
- 移行して保存すると、そのプロジェクトを過去の ConfigurationDesk バージョンで開くことはできなくなります。

ConfigurationDesk 6.8 (dSPACE Release 2021-B) で変更されたプロジェクトやアプリケーションのファイル形式について、以下の点にご注意ください。

- 以前のバージョンの ConfigurationDesk で作成したプロジェクトやアプリケーションファイル、バックアップも開くことができます。ConfigurationDesk では、プロジェクトを開く際に移行を確認するメッセージが表示されます。



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とアプリケーションファイルは、それぞれの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 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。

ConfigurationDesk 22.1 (dSPACE Release 2022-A) で導入された新しいプロジェクト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プロジェクト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。これを回避するために、プロジェクトキャッシュフォルダのパスを短くすることができます。詳細については、「[\[Paths\]ページ](#)」(『[ConfigurationDesk ユーザインタフェースリファレンス](#)』)を参照してください。

2022 年 11 月

**SCALEXIO ターゲットアーキテクチャのモデルインプリメンテーションコンテナとの互換性**

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 64 ビットターゲットアーキテクチャと互換性がなくなりました。モデルインプリメンテーションコンテナを Linux 64 ビット互換のバイナリで再生成するか、Linux 32 ビットターゲットアーキテクチャを指定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。

#### Bus Manager に関する変更

**生成された ISignal グループのアップデートビット用 ISignal を既存の ConfigurationDesk アプリケーションで使用できない場合** ISignal グループのアップデートビット用に Bus Manager によって生成される ISignal は、Bus Manager (スタンドアロン) 22.1 以前のバージョンで作成された ConfigurationDesk アプリケーションでは使用できません。このような ISignal グループを使用できるようにするには、関連する通信マトリクスの別のバージョン、つまり Content hash 値が異なる通信マトリクスを、影響を受ける ConfigurationDesk アプリケーションに追加する必要があります。その後、関連するバスコンフィギュレーションで古い通信マトリクスを置き換えて、必要に応じて新しく使用可能になった ISignal を手動で割り当てます。

通信マトリクスの置き換えの詳細については、「[割り当てられた通信マトリクスの交換](#)」(『Bus Manager (スタンドアロン) 実装ガイド』)を参照してください。

#### 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 2022-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、「[検索パスの管理](#)」(『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。

## ConfigurationDesk の廃止

---

<b>V-ECU インプリメンテーションコンテナのサポートの廃止</b>	ConfigurationDesk 2022-B(22.2)では、V-ECU インプリメンテーションコンテナ (VECU ファイル)のサポートが廃止されました。以前の dSPACE Release で作成された、V-ECU インプリメンテーションコンテナが含まれている ConfigurationDesk アプリケーションを開くと、V-ECU インプリメンテーションコンテナは自動的に削除されます。
<b>ConfigurationDesk アプリケーションの Excel エクスポートの廃止予定</b>	アクティブな ConfigurationDesk アプリケーションの設定データを 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 で削除されます。  また、対応するツール自動化 API コマンド ICaAlgorithms:ExportConfiguration も削除されます。
<b>カスタムデバイスプロパティの廃止予定</b>	カスタムデバイスプロパティをデバイストポロジの元素に追加する機能は、ConfigurationDesk 2023-B で削除されます。
<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。

---

# ControlDesk

---

## 次のステップ

## 本章の内容

ControlDesk 2022-B の新機能.....	64
ControlDesk 2022-B への移行.....	71

## ControlDesk 2022-B の新機能

### 次のステップ

### 本章の内容

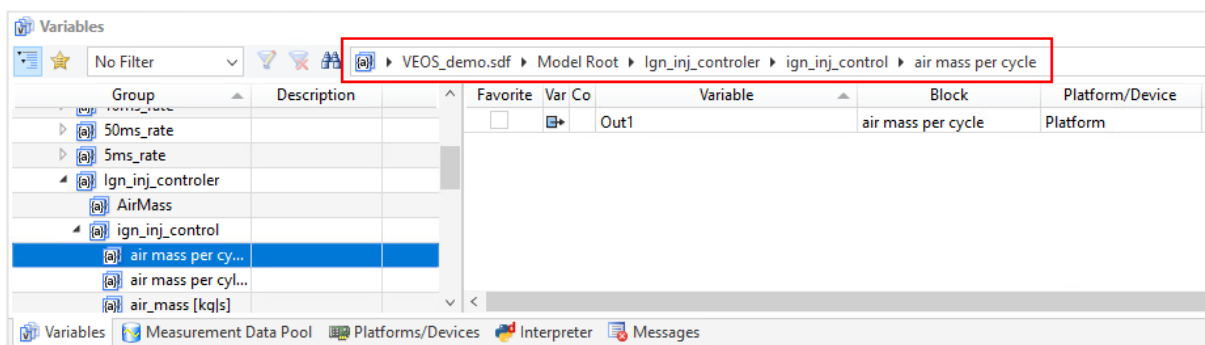
新しいユーザインターフェース処理機能 (ControlDesk 2022-B).....	64
ControlDesk 2022-B の新しいユーザインターフェース処理機能について概要を説明します。	
プラットフォーム管理およびプラットフォーム/デバイスの新機能 (ControlDesk 2022-B).....	65
ControlDesk 2022-B のプラットフォーム管理およびプラットフォーム/デバイスの新機能について概要を説明します。	
新しい変数管理機能 (ControlDesk 2022-B).....	65
ControlDesk 2022-B の新しい変数管理機能について概要を説明します。	
新しい計測および記録機能 (ControlDesk 2022-B).....	66
ControlDesk 2022-B の新しい計測および記録機能について概要を説明します。	
新しい Bus Navigator 機能 (ControlDesk 2022-B).....	67
ControlDesk 2022-B の新しい Bus Navigator 機能について概要を説明します。	
新しい自動化機能 (ControlDesk 2022-B).....	70
ControlDesk 2022-B の新しい自動化インターフェース機能について概要を説明します。	

## 新しいユーザインターフェース処理機能 (ControlDesk 2022-B)

### [Variables]ペインと Measurement Data Pool のナビゲーションの改善

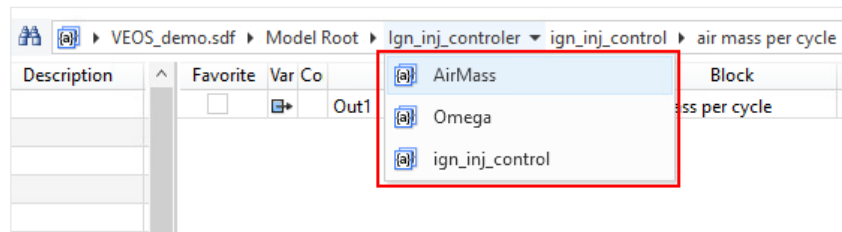
[Variables]ペインと Measurement Data Pool で使用できる新しいパンくずリストコントロールにより、特にネストされた構造を持つ変数記述ファイルを扱うときには、ナビゲーションが容易になります。

次の図は、[Variables]ペインの新しいコントロールの例を示しています。

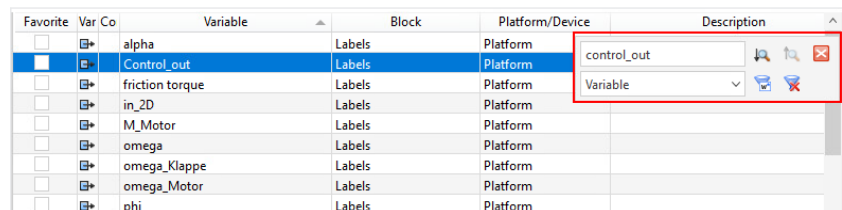




パンくずリストコントロールが有効な場合、つまりパンくずリストの項目をクリックできる場合は、次の図に示すように下位の項目にアクセスすることができます。



Ctrl + F キーを押すと、変数の検索やフィルタリングを行うことができます。



詳細については、以下を参照してください。

- 「[Variables]ペインの基礎」(『ControlDesk 変数管理』)
- 「Measurement Data Pool の基礎」(『ControlDesk 計測および記録』)

## 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 2022-B)

**cmdloader: プラットフォームのフラッシュメモリを完全に消去**

cmdloader コマンドラインツールにより、プラットフォームのフラッシュメモリを完全に消去できるようになりました。

次の例は、「ds1403」という名前のプラットフォームのフラッシュメモリを完全に消去する方法を示しています。

```
cmdloader -clear_flash -p ds1403
```

詳細については、「コマンドラインでのアプリケーションの取り扱い」(『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 2022-B)

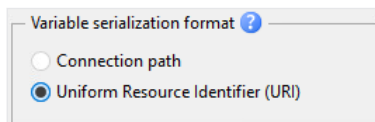
**変数の URI (Uniform Resource Identifier)**

変数のテキスト表現に、接続パスの他に、URI (Uniform Resource Identifier) 形式を選択できるようになりました。

この形式は、変数をクリップボードにコピーして自動化スクリプトに貼り付ける場合などに使用します。

URI 形式を選択するには、次の 2 つの方法があります。

- [ControlDesk Options]ダイアログの[Variables]ページで[Variable serialization format]オプションを選択します。



詳細については、「Variables Page」([『ControlDesk 変数管理』](#))を参照してください。

- 内部インタプリタで次の Python コードを入力します。

```
Application.VariablesManagement.VariableSerializationFormat = 1
```

詳細については、「VariablesManagement / IXaVariablesManagement <<Interface>>」([『ControlDesk 自動化』](#))を参照してください。

## 関連トピック

### 参考文献

[VariablesManagement / IXaVariablesManagement <<Interface>>](#) ([『ControlDesk 自動化』](#))

## 新しい計測および記録機能(ControlDesk 2022-B)

### データロギングの改善

ControlDesk 2022-B では、データロギング設定は、関連する リアルタイムアプリケーション(RTA)ファイルに含まれます。

そのため、ControlDesk PC をリアルタイムハードウェアに接続してデータロギング設定をロードする必要がなくなります。たとえば、RTA ファイルを含む USB マスストレージデバイスをハードウェアに接続すると、RTA ファイル(データロギング設定を含む)をリアルタイムハードウェアにロードすることができます。

これにより、データロギング設定を簡単に配布できるようになります。

ControlDesk 2022-B でのデータロギングワークフロー。

1. ControlDesk で、サポートされているプラットフォームと変数記述ファイルを含む実験を開きます。
2. 「データロギングの設定の基礎」([『ControlDesk 計測および記録』](#))の説明に従って、データロギングを設定します。
3. データロギング設定を有効にします。  
これで設定がリアルタイムアプリケーション(RTA)ファイルにロードされます。
4. データロギングを実行するには、次の方法があります。
  - ControlDesk で、接続されたリアルタイムハードウェアのフラッシュメモリまたは RAM に RTA ファイルをロードします。リアルタイムアプリケーションを起動すると、データロギングセッションが自動的に開始されます。  
または

- リアルタイムハードウェアの Web インターフェースを使用して、データロギング設定を含む RTA ファイルをリアルタイムハードウェアにロードします。  
または
  - (MicroAutoBox III のみ) RTA ファイルを USB マスストレージデバイスにコピーし、そのデバイスを MicroAutoBox III に接続します。次に MicroAutoBox III を起動したときにデータロギング設定を含む RTA ファイルがロードされ、データロギングが開始されます。
5. データロギングを停止します。

詳細については、「[データロギング](#)」(『ControlDesk 計測および記録』)を参照してください。

## 関連トピック

### 基本概要

[データロギングの設定の基礎](#) (『ControlDesk 計測および記録』)

## 新しい Bus Navigator 機能 (ControlDesk 2022-B)


### Bus Navigator (スタンドアロン)

新しい Bus Navigator (スタンドアロン) では、ControlDesk Bus Navigator の機能をいくつかの制限付きで使用することができます。たとえば、オンライン化と計測はバスモニタリングデバイスにのみ対応しています。

Bus Navigator (スタンドアロン) を使用するために必要なのは、有効な ControlDesk Bus Navigator Module (CONTROLDESK\_BNV) ライセンスのみです。ControlDesk メインバージョンのライセンスは必要ありません。後で CONTROLDESK ライセンスを追加することで、フル機能のメインバージョンの ControlDesk にアップグレードすることができます。


Bus Navigator (スタンドアロン) は以下の機能をサポートしています。

- バスモニタリングデバイスを使用した、CAN、Ethernet、LIN バス上での通信のモニタリングとロギング。
- ログ記録した CAN および Ethernet 通信の再生。
- CAN、LIN、Ethernet のメッセージ、フレーム、PDU を処理する計器。
- バスモニタリングデバイスと接続するバスインターフェース。
  - dSPACE およびサポートされているサードパーティの PC ベースのバスインターフェース。
  - SCALEXIO、MicroAutoBox III、または VEOS のバスチャンネル。

詳細については、「[通信バスにアクセスするためのインターフェースのサポート](#)」(『ControlDesk プラットフォーム管理』)を参照してください。

**注記**

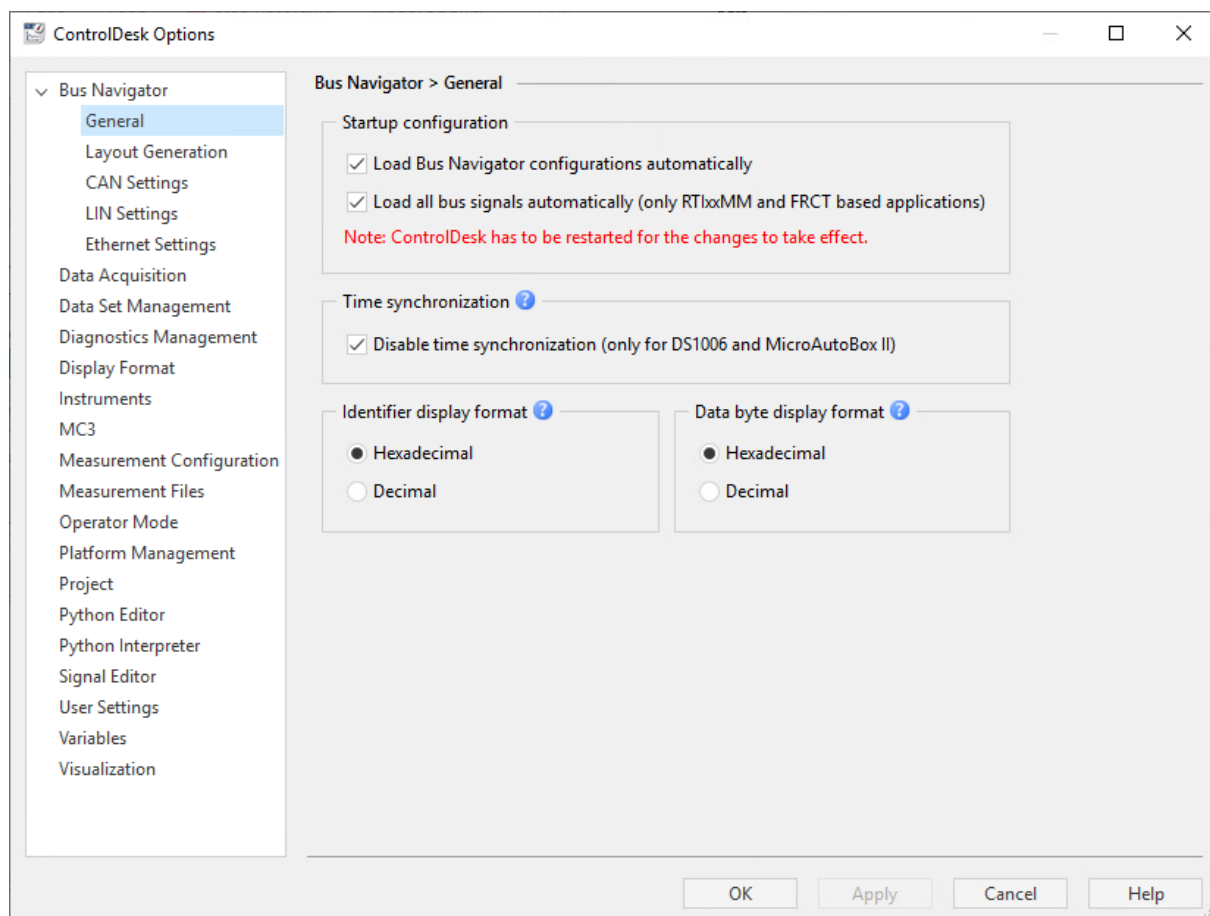
ControlDesk 2022-B 以降、CAN メッセージの処理および CAN バス通信のモニタリングとロギングのために、CAN ベースの dSPACE リアルタイムハードウェアおよび PC ベースの CAN インターフェース (DCI-CAN2 など) にアクセスするには、Bus Navigator Module の有効なライセンスが必要になります。この機能は、メインバージョンのライセンスから Bus Navigator Module のライセンスに移行されました。


Bus Navigator (スタンドアロン)の詳細については、「[Bus Navigator\(スタンドアロン\)の使用](#)」(『ControlDesk Bus Navigator』)を参照してください。

**Bus Navigator ユーザーインターフェースの改善****[Bus Navigator]ページ([ControlDesk Options]ダイアログ)の変更**

[ControlDesk Options]ダイアログに含まれる[Bus Navigator]ページが変更されました。

次の図は新しい構造を示しています。



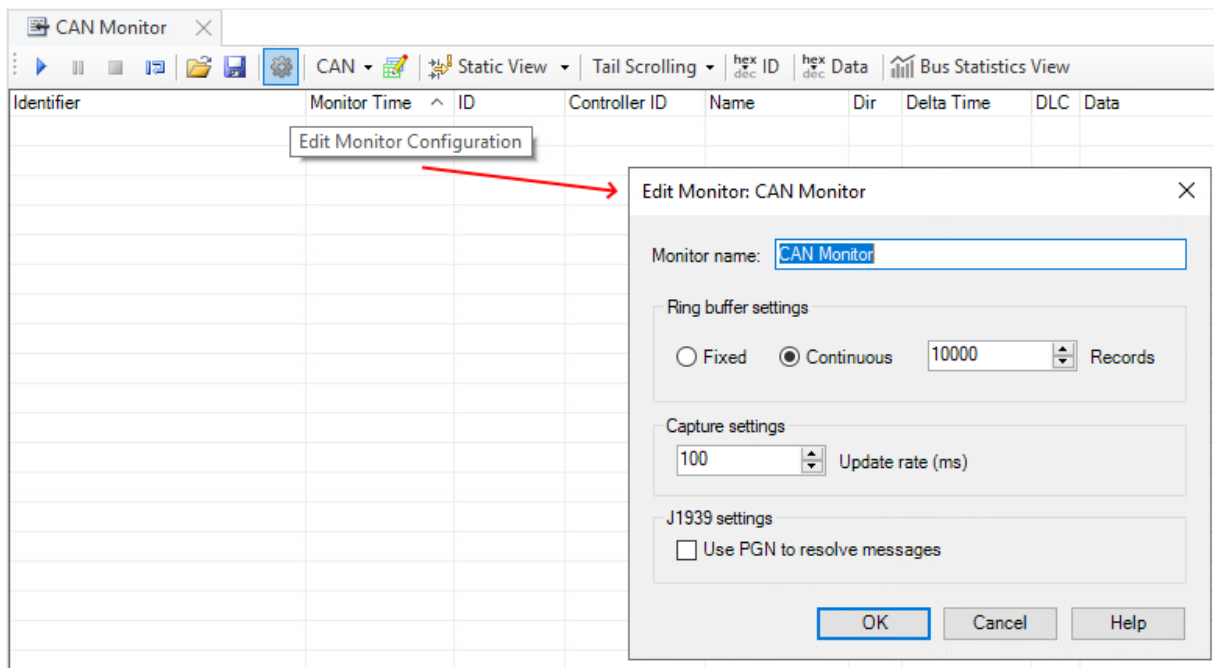
詳細については、「[Bus Navigator Page](#)」(『ControlDesk Bus Navigator』)を参照してください。

### Bus Navigator エlementプロパティの表示

- ControlDesk の[Properties]ペインに、[Bus Navigator]ペインで選択されている次のElementに関する情報も表示されるようになりました。
  - チャンネル
  - コントローラ
  - グローバル時間ドメイン
- [Bus Navigator]ペインで選択されているElementのソース、つまりインプリメンテーションソフトウェアの名前や通信マトリクスファイルタイプが表示されるようになりました。
- コンテナ IPDU に固有の[PDU]プロパティが表示されます。

詳細については、「[Properties Related to Bus Navigator Elements](#)」(『ControlDesk Bus Navigator』)を参照してください。

**[Edit Monitor]ダイアログへのアクセスの改善** モニタリングリストのツールバーに表示される新しい  ボタンにより、[Edit Monitor]ダイアログに簡単にアクセスできるようになりました。



「[Add/Edit Monitor Dialog](#)」(『ControlDesk Bus Navigator』)を参照してください。

### R21-11 ARXML ファイルのサポート

ControlDesk Bus Navigator は、AUTOSAR システムテンプレートの R21-11 パーバージョンに準拠した AUTOSAR システム記述 (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。

## 新しい自動化機能(ControlDesk 2022-B)

Table Editor に接続された変数の個々の関数と軸の値の書き込み

- IViTableEditorInstrument 自動化インターフェースでは、SetSpecificFunctionValues メソッドにより、Table Editor に接続された変数の関数値を個別に設定できるようになりました。  
詳細については、「TableEditorInstrument / IViTableEditorInstrument <<Interface>>」(『ControlDesk 自動化』[📖](#))を参照してください。
- IViValueAxis 自動化インターフェースでは、SetSpecificValue メソッドにより、Table Editor に接続された変数の個々の軸の値を設定できるようになりました。  
詳細については、「ValueAxis / IViValueAxis <<Interface>>」(『ControlDesk 自動化』[📖](#))を参照してください。

レイアウトのデータソースの切り替え

レイアウトの自動化インターフェースでは、SwitchDataSource プロパティにより、レイアウト内の変数接続のデータソースを切り替えることができるようになりました。これにより、現在の計測値をさまざまな記録と比較することができます。

詳細については、「LayoutDocument / IXaLayoutDocument <<Interface>>」(『ControlDesk 自動化』[📖](#))を参照してください。

計算変数の出力パラメータの指定

IXaCalculatedVariable 自動化インターフェースでは、OutputParameters プロパティにより、計算変数に出力パラメータを指定できるようになりました。

詳細については、「CalculatedVariable / IXaCalculatedVariable <<Interface>>」(『ControlDesk 自動化』[📖](#))を参照してください。

すべての変数記述ファイルで変数を検索

IXaVariablesManagement 自動化インターフェースでは、FindVariable メソッドにより、現在ロードされている実験のすべての変数記述ファイルで変数を検索できるようになりました。

詳細については、「VariablesManagement / IXaVariablesManagement <<Interface>>」(『ControlDesk 自動化』[📖](#))を参照してください。

# ControlDesk 2022-B への移行

## ControlDesk 2022-B への移行

### 概要

ControlDesk 7.6 から ControlDesk 2022-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

7.6 より前のバージョンから ControlDesk 2022-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。

### CAN バス通信を扱うために必要な CONTROLDESK\_BNV ライセンス

ControlDesk 2022-B では、CAN ベースの dSPACE リアルタイムハードウェアや PC ベースの CAN インターフェース (DCI-CAN2 など) にアクセスして、CAN メッセージの処理や CAN バス通信のモニタリングおよびロギングを行うには、ControlDesk Bus Navigator Module の CONTROLDESK\_BNV ライセンスが必要になります。

### ControlDesk 2022-B での DSSIGCONV ツールの変更

**計測配列および値ブロックの単一要素の抽出なし** コマンドラインツール DSSIGCONV の `/s:list` および `/sl:filename` パラメータで、計測配列と値ブロックの単一要素を指定できなくなりました。つまり、計測配列と値ブロックの単一要素をソース計測データファイルから抽出できなくなりました。

**線形同スケールリングの削除** ソース宛先ファイル内の線形同スケールリング (係数 = 1.0、オフセット = 0.0) は、エクスポート時に削除されます。

**サポートされないスケールリング形式** 以下のスケールリング形式はサポートされません。

- %c
- %C
- %n
- %o
- %p
- %s
- %S
- %Z

以下の形式はフロート形式 (%f) に変更されます。

- %a
- %A
- %e
- %E
- %g
- %G

DSSIGCONV ツールの詳細については、「Using DSSIGCONV」(『ControlDesk 計測および記録』( ))を参照してください。

## ControlDesk 2022-B での Measurement Data API の変更

ControlDesk 2022-B には、新しい Measurement Data API が実装されています。そのため、既存スクリプトの調整が必要になる場合があります。

**Measurement オブジェクトが Measurements コレクションから削除後に無効になる** ControlDesk 2022-B では、Measurement を Measurements コレクションから削除すると、Measurement オブジェクトは無効になります。これにアクセスしようとすると、例外が発生します。

**Signals コレクションに含まれなくなった測定配列と値ブロックの要素** 測定配列と値ブロックを含む Signals コレクションのコンテンツ、すなわち vtMeasurementArray タイプと vtValueBlock タイプの信号は、次のようになります。

- ControlDesk 7.6 以前では、測定配列と値ブロックを含む Signals コレクションには、個々の配列要素も含まれていました。
- ControlDesk 2022-B では、測定配列と値ブロックを含む Signals コレクションには、個々の配列要素は含まれなくなりました。

そのため、個々の配列要素のデータにアクセスするには、インデックスを使用する必要があります。以下のコードの抜粋で例を示します。

```
>>> Signal = Measurement.Signals["PLATFORM/ArraySignal/5ms"]
>>> Signal.Data[1]
(0.01, 0.011, 0.012, 0.013, 0.014)
```

**スケールリングがオプションに** ControlDesk 2022-B で信号のスケールリングの仕様が変更されました。

- ControlDesk 7.6 以前では、Signals コレクションに信号を追加する場合、スケールリングの指定が必須でした。変換なしの信号については、線形同スケールリング(係数 = 1.0、オフセット = 0.0)が必要でした。
- ControlDesk 2022-B では、Signals コレクションに信号を追加する場合、変換なしの信号については None スケールリングを指定します。

```
>>> SignalWithoutSourceData = Measurement.Signals.Add('Without Source Data', 'Without Source Data', 'Device', '', SignalData, XAxis, None)
```

計測データファイルをロードすると、線形同スケールリングは削除されます。

**重複するテーブルスケールリングの追加不可** ControlDesk 2022-B では、テーブルの範囲指定が重複しているテーブルスケールリングを追加すると、例外が発生します。

**有理関数スケールリングがソースから変換値に変換** 有理関数スケールリング、すなわち stRationalFunction タイプのスケールリングの動作が ControlDesk 2022-B で変更されました。

- ControlDesk 7.6 以前では、有理関数スケールリングは変換値からソースに計測値を変換するように定義されていました。
- ControlDesk 2022-B では、ソースから変換値に計測値を変換するように定義されます。これは他のスケールリングタイプの動作と同じです。

**一意のラスタ名** ControlDesk 2022-B では、Measurement Data API でプラットフォームの計測ラスタの一意名が必要になります。そのため、計測データファイルをロードする場合、必要に応じて Measurement Data API がラスタ名の曖昧性を解消します。



次の表は、ラスタ名の例を示しています。

ControlDesk 7.6 以前	ControlDesk 2022-B 以降
PLATFORM/Signal/5ms (2 つの異なるラスタに同じ名前)	<ul style="list-style-type: none"> <li>▪ PLATFORM/Signal/5ms</li> <li>▪ PLATFORM/Signal/5ms(2) (曖昧性解消後)</li> </ul>

そのため、ControlDesk 2022-B の Measurement Data API でスクリプトを再利用する場合、X 軸とシグナルのキーが異なる場合があります。ラスタのキーは、Signal オブジェクトから入手できます。

**サポートされないスケーリング形式** 以下のスケーリング形式はサポートされません。

- %c
- %C
- %n
- %o
- %p
- %s
- %S
- %Z

以下の形式はフロート形式 (%f) に変更されます。

- %a
- %A
- %e
- %E
- %g
- %G

**GetDeviceOptions/SetDeviceOptions メソッドの廃止** Measurement オブジェクトの次のメソッドは廃止されました。ControlDesk 2022-B 以降、以下は使用できなくなります。

- GetDeviceOptions
- SetDeviceOptions

**Signals コレクションでのシグナルの順序の変更** ControlDesk 2022-B では、計測データファイルをロードする場合、Signal コレクションの Signals オブジェクトの順序が、以前のバージョンの Measurement Data API でファイルを読み込む場合の順序とは異なります。

**同じ計測データファイルを複数回ロードしない** ControlDesk 2022-B では、同じ計測データファイルを複数回ロードすると、例外が発生します。

ただし、同名の計測データファイルでも、別のフォルダにあり、Measurement オブジェクトに別の名前を指定している場合は、ロードすることができます。


LoadSaveAndExport.py デモスクリプトから抜粋した次のスクリプトで、例を示します。

```
Measurement = Measurements.Load(SubFolderPath, Name = 'Measurement from sub-folder')
```

Measurement Data API の詳細については、「[Measurement Data API の基礎](#)」([『ControlDesk Measurement Data API』](#))を参照してください。

---

### 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

[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』)

# dSPACE AUTOSAR Compare

## 次のステップ

## 本章の内容

dSPACE AUTOSAR Compare 2022-B の新機能.....	75
dSPACE AUTOSAR Compare は、新しい AUTOSAR Release をサポートしています。	
dSPACE AUTOSAR Compare 1.1 の新機能.....	75
dSPACE AUTOSAR Compare は、Git などのバージョン管理システムに統合して、ARXML ファイルの異なるリビジョン間の差分を表示したり、マージ競合の解決を支援することができます。	

## dSPACE AUTOSAR Compare 2022-B の新機能

### 全般的な改良点

新バージョンは、AUTOSAR Release R21-11 にも対応しています。

### 関連トピック

#### 基本概要

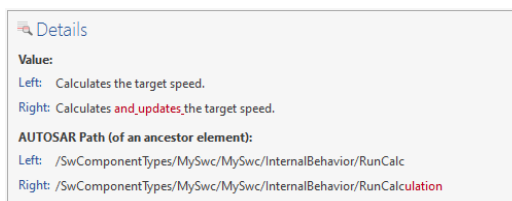
[Basics on dSPACE AUTOSAR Compare \(『dSPACE AUTOSAR Compare Manual』\)](#)

## dSPACE AUTOSAR Compare 1.1 の新機能

### 全般的な改良点

- 新バージョンは、AUTOSAR Release R20-11 にも対応しています。
- 比較ページの要素への移動に AUTOSAR パスを指定できるようになりました。

- 追加された[Details]ペインでは、選択された AUTOSAR エLEMENTの差分がハイライトされ、AUTOSAR のフルパスが表示されます。



## バージョン管理システムへの統合

dSPACE AUTOSAR Compare は、Git などのバージョン管理システムに統合することができ、ARXML ファイルの異なるリビジョン間の差分を表示したり、マージ競合を支援したりすることができます。以下のセクションでは、Git を中心に dSPACE AUTOSAR Compare の追加、差分の表示、およびマージ競合の解決について説明します。

## dSPACE AUTOSAR Compare の追加

dSPACE AUTOSAR Compare を Git の差分およびマージツールとして追加することができます。前述の目的のために、dSPACE AUTOSAR Compare をデフォルトのツールとして使用することができます。

## 表示の違い

コミットされていないファイルへのローカルな変更を検査するには、次のコマンドを使用します。

```
git difftool -t darc <file>
```

dSPACE AUTOSAR Compare では、最後のコミットとローカルでの変更の間でファイルの内容を比較できるセッションが開きます。

dSPACE AUTOSAR Compare をデフォルトの Git 差分ツールとして設定している場合は、'-t darc'を省略できます。ファイル名を指定しない場合、変更されたすべてのファイルに対してセッションが開かれます。

## マージ競合の解決

分岐をマージまたはリベースする場合には、Git はこのメッセージでマージ競合を通知します。

```
Auto-merging <file>
CONFLICT (content): Merge conflict in <file>
Automatic merge failed; fix conflicts and then commit the result.
```

次のコマンドを使用して、dSPACE AUTOSAR Compare でマージ競合を解決することができます。

```
git mergetool -t darc <file>
```

dSPACE AUTOSAR Compare をデフォルトの Git マージツールとして設定している場合は、'-t darc'を省略できます。

Git の設定で mergetool.prompt が'true'に設定されている場合、Git は詳細情報を表示し、dSPACE AUTOSAR Compare でマージ競合を手作業で解決できるようになりました。そうでない場合は、マージの競合は自動的に解決されます。

---

関連トピック

基本概要

[Integration in Other Tools \(『dSPACE AUTOSAR Compare Manual』📖\)](#)



# DCI Configuration Tool

## DCI Configuration Tool 2022-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 1.5.5 が  
DCI Configuration Tool 2022-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 FlexRay Configuration Package

## dSPACE FlexRay Configuration Package 2022-B の新機能

### FlexRay Configuration Package

**ソース切り替え変数のデフォルト値** FlexRay Configuration Package 4.9 以前では、ソース切り替え変数は常に Simulink モデルに初期設定されています。データソースを制御するソース切り替え変数の値は、実行時に、TRC ファイル変数ごとに個別にのみ変更することができます。

FlexRay Configuration Package 2022-B では、FlexRay Configuration Tool で、つまりコード生成前に、ソース切り替え変数のデフォルト値を指定できるようになりました。[General Properties]ダイアログの[Generators]ページで、すべてのソース切り替え変数のグローバルなデフォルト値を選択することができます。つまり、初期データソースをデフォルトで TRC 変数に設定するか、Simulink モデルに設定するかを指定することができます。

「[Generation]ページ」(『FlexRay Configuration Tool リファレンス』[📖](#))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 22.2 の新機能.....	83
dSPACE Installation Manager 22.2 への移行.....	84

## dSPACE Installation Manager 22.2 の新機能

### インストレーション管理のための新機能

dSPACE Installation Manager では、インストールされている dSPACE ソフトウェアのアップデート(パッチなど)が利用可能かどうかを確認して表示できるようになりました。これは手作業でも自動でも実行することができます。自動の場合、dSPACE Installation Manager は、通知機能を使用してアップデートが利用可能かどうかを通知します。

#### 注記

次の制限事項に注意してください。

- dSPACE Release 2020-A 以降でリリースされた dSPACE ソフトウェアのアップデートのみ表示されます。
- 以下のソフトウェアについては、アップデートが利用可能な場合でも、アップデートが表示されません。
  - dSPACE Firmware Manager、dSPACE Firmware Archives
  - dSPACE Installation Manager
  - dSPACE Release で提供されるサードパーティ製ソフトウェア (Wibu-Systems のソフトウェアなど)
  - dSPACE Release で提供されない dSPACE ソフトウェア (dSPACE ソリューションコレクションまたは AURELION に含まれるソフトウェア製品など)。
- Hotfix はサポートされていないため、通常は表示されません。

特定の管理者の使用事例など、詳細については「使用可能な dSPACE ソフトウェアのアップデートに関する情報の取得」(『dSPACE ソフトウェアインストレーションの管理』)を参照してください。

---

### ライセンス管理のための新機能

Microsoft Windows で実行している保護された dSPACE ソフトウェアと連携するフローティングネットワークライセンス用サーバで、Ubuntu Linux をオペレーティングシステムとして使用することもできるようになりました。詳細については 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。

## dSPACE Installation Manager 22.2 への移行


---

### CmDongles の使用

dSPACE Installation Manager 22.2 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。

# dSPACE Python Extensions

## dSPACE Python Extensions 2022-B の新機能

---

### 新機能

dSPACE Python Extensions には新機能はありません。



# dSPACE XIL API .NET

## 次のステップ

## 本章の内容

dSPACE XIL API .NET 2022-B の新機能.....	87
dSPACE XIL API .NET 2022-B への移行.....	88

## dSPACE XIL API .NET 2022-B の新機能

### 新機能

dSPACE XIL API.NET 2022-B には以下の新機能が含まれています。

- 新しい ECUPort の実装により、以下の初めての機能が提供されます。
  - 適合値の読み取り。
  - 適合値の書き込み。
  - パラメータセット (CDFX) のダウンロード。
  - サポートされる変数は、スカラー、配列、曲線、および任意のデータタイプのマップです。
  - V-ECU のみがサポートされます。
  - Windows と Linux で利用可能です。

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

- 新しい ECUPort の実装の要件を満たすために、VariableInfo インターフェースが強化されました。

以下の機能を使用できるようになりました。

- 曲線、行列、マップの各変数の軸タイプを取得することができます。
- 変換方法に関する情報を取得することができます。
- 説明と物理的な単位を取得することができます。

MAPort も VariableInfo インターフェースを使用するため、MAPort の実装でもこれらの機能を利用できるようになりました。

## dSPACE XIL API .NET 2022-B への移行

### XIL API 2.2.0 への移行

**ソースコードの変更** アプリケーションを ASAM XIL API 2.2.0 規格に移行するには、リファレンスを修正する必要があります。

- C#: Solution Explorer で[Add Reference]ダイアログを開き、DLL のリファレンスをバージョン 2.2 のリファレンスに置き換えます。  
その後、C#クライアントアプリケーションを再コンパイルしてください。
- Python: バージョン文字列と公開鍵トークンを修正します。ASAM XIL API 2.2.0 では、すべてのアセンブリに同じ公開鍵トークンがあります。  
更新後のリファレンスは、次の例のようになります。

```
# Load ASAM assemblies from the global assembly cache (GAC)
clr.AddReference("ASAM.XIL.Implementation.TestbenchFactory, Version=2.2.0.0, Culture=neutral,
PublicKeyToken=bf471dff114ae984")
clr.AddReference("ASAM.XIL.Interfaces, Version=2.2.0.0, Culture=neutral, PublicKeyToken=bf471dff114ae984")
```

**フレームワーク構成の変更** XIL API 2.1.0 と 2.2 のフレームワーク構成は互換性がありません。不正な XML スキーマが使用された場合、例外が発生し、XML ファイルが有効でないことを表示します。

ターゲット状態の構成方法が異なってしまったため、以下の例では SCALEXIO プラットフォームのフレームワーク構成全体を示しています。修正箇所はマークされています。

### XIL API 2.1.0

```
<?xml version="1.0" encoding="utf-8"?>
<FrameworkConfiguration xmlns="http://www.asam.net/XIL/FrameworkConfiguration/2.0">
  <MappingFileList>
    <MappingFile Name="DemoFrameworkLabels">..\FrameworkMappings\FrameworkLabels.xml</MappingFile>
    <MappingFile Name="DemoLabelMapping_SCALEXIO">..\FrameworkMappings\LabelMapping_SCALEXIO.xml</MappingFile>
    <MappingFile Name="DemoUnits">..\FrameworkMappings\Units.xml</MappingFile>
  </MappingFileList>
  <PortDefinitionList>
    <MAPortDefinition InstanceName="dSPACE SCALEXIO MAPort" InitOrder="0" ShutdownOrder="0" TargetState="eSIMULATION_RUNNING">
      <VendorName>dSPACE GmbH</VendorName>
      <ProductName>XIL_API</ProductName>
      <ProductVersion>2021-B</ProductVersion>
      <PortConfigurationFile>..\..\..\MAPort\Common\PortConfigurations\MAPortConfigSCALEXIO.xml</PortConfigurationFile>
    </MAPortDefinition>
  </PortDefinitionList>
</FrameworkConfiguration>
```

### XIL API 2.2.0

```
<?xml version="1.0" encoding="utf-8"?>
<FrameworkConfiguration xmlns="http://www.asam.net/XIL/FrameworkConfiguration/2.2.0">
  <MappingFileList>
    <MappingFile Name="DemoFrameworkLabels">..\FrameworkMappings\FrameworkLabels.xml</MappingFile>
    <MappingFile Name="DemoLabelMapping_SCALEXIO">..\FrameworkMappings\LabelMapping_SCALEXIO.xml</MappingFile>
    <MappingFile Name="DemoUnits">..\FrameworkMappings\Units.xml</MappingFile>
  </MappingFileList>
  <PortDefinitionList>
    <MAPortDefinition InstanceName="dSPACE SCALEXIO MAPort" InitOrder="0" ShutdownOrder="0">
      <VendorName>dSPACE GmbH</VendorName>
      <ProductName>XIL_API</ProductName>
      <ProductVersion>2022-A</ProductVersion>
      <TargetState>
        <eSIMULATION_RUNNING>
          <PortConfigurationFile>..\..\..\MAPort\Common\PortConfigurations\MAPortConfigSCALEXIO.xml</PortConfigurationFile>
          <ForceConfig>>false</ForceConfig>
        </eSIMULATION_RUNNING>
      </TargetState>
    </MAPortDefinition>
  </PortDefinitionList>
</FrameworkConfiguration>
```



**マッピングファイルの変更** XIL API 2.1.0 と 2.2 のマッピングファイルには互換性がありません。不正な XML スキーマが使用された場合、例外が発生し、XML ファイルが有効でないことを表示します。

次のエントリのみ変更する必要があります。

<b>XIL API 2.1.0</b>
<code>&lt;Mapping xmlns="http://www.asam.net/XILAPI/Mapping/2.1.0"&gt;</code>
<b>XIL API 2.2.0</b>
<code>&lt;Mapping xmlns="http://www.asam.net/XIL/Mapping/2.2.0"&gt;</code>



# ECU Interface Manager

## 次のステップ

## 本章の内容

ECU Interface Manager 2022-B の新機能.....	91
ECU Interface Manager 2022-B の新機能の概要。	
ECU Interface Manager 2022-B の互換性.....	92
ECU Interface Manager 2022-B の互換性に関する情報を提供します。	
ECU Interface Manager 2022-B への移行.....	93
ECU Interface Manager 2022-B への移行方法に関する情報。	

## ECU Interface Manager 2022-B の新機能

### ECU 変数への動的アクセスの準備

ECU Interface Manager を使用して、ECU 変数への動的アクセスを準備できるようになりました。その結果、関連するリアルタイムアプリケーションの実行中に、読み書きする変数の数、アドレス、データタイプ、変数変換を変更することができます。

詳細については、「動的データアクセスの設定の基礎」(『ECU Interface Manager マニュアル』[📖](#))を参照してください。

### XCP on CAN: 特定の CAN ID の割り当て

XCP on CAN では、DAQ リストを使用する XCP メッセージの送信に CAN ID のリストを指定できるようになりました。スタティックとダイナミックのどちらの DAQ リストを使用するか、および PID 送信が有効か無効かによって、A2L ファイルまたは ECU Interface Manager で指定された CAN ID が XCP メッセージの送信に使用されません。

以下を参照してください。

- 「データアクセス」(『ECU Interface Manager マニュアル』[📖](#))
- 「動的データアクセス」(『ECU Interface Manager マニュアル』[📖](#))
- 「ECU インターフェース」(『ECU Interface Manager マニュアル』[📖](#))
- 「関数」(『ECU Interface Manager マニュアル』[📖](#))

**XCP on CAN:PID 送信の無効化**

ECU Interface Manager では、XCP on CAN ECU インターフェースと接続したパケット識別子(PID)の送信を無効にできるようになりました。これにより、XCP データ転送オブジェクト(DTO)パケットの全バイトが DAQ データ転送に使用されるため、DAQ データのスループットが向上します。

以下を参照してください。

- 「データアクセス」(『ECU Interface Manager マニュアル』[📖](#))
- 「動的データアクセス」(『ECU Interface Manager マニュアル』[📖](#))
- 「関数」(『ECU Interface Manager マニュアル』[📖](#))

## ECU Interface Manager 2022-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 2.9 <sup>1)</sup>	バージョン 2.10 <sup>2)</sup>	バージョン 2.11 <sup>3)</sup>	バージョン 2022-B <sup>4)</sup>
ConfigurationDesk 2022-B <sup>4)</sup>	✓ <sup>5)</sup>	✓ <sup>5)</sup>	✓ <sup>6)</sup>	✓ <sup>6)</sup>
ConfigurationDesk 22.1 <sup>3)</sup>	✓ <sup>5)</sup>	✓ <sup>5)</sup>	✓ <sup>6)</sup>	✓ <sup>6)</sup>
ConfigurationDesk 6.8 <sup>2)</sup>	✓	✓	-	-
ConfigurationDesk 6.7 <sup>1)</sup>	✓	✓	-	-

1) dSPACE Release 2021-A

2) dSPACE Release 2021-B

3) dSPACE Release 2022-A

4) dSPACE Release 2022-B

5) 32 ビットターゲットアーキテクチャの場合のみ。

6) 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 2022-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 2022-B でプロジェクトを再利用することができます。  
ECU Interface Manager 2022-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 2022-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 2.7 より前のバージョンから ECU Interface Manager 2022-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 22.2 の新機能

### 一般的な機能強化

通常、Firmware Manager は、ファームウェアのアップデート後に必要な再起動を開始することができます。一部の dSPACE プラットフォームでは、この動作がサポートされていませんでした。ユーザには、ハードウェアを手動で再起動するように案内していました。

現在、ファームウェアバージョン 22.2 にアップデートしたときの自動再起動は、条件付きで以下のプラットフォームで利用できます。

- MicroAutoBox III  
以前にロードした DS1403 のファームウェアバージョンが 5.0 以上であること、つまりインストールされている Firmware Archives のセットアップバージョンが 3.2 (dSPACE Release 2020-B) 以上であることが必要です。
- SCALEXIO AutoBox/LabBox  
以前にロードした DS6001 のファームウェアバージョンが 5.0 以上であること、つまりインストールされている Firmware Archives のセットアップバージョンが 3.2 (dSPACE Release 2020-B) 以上であることが必要です。
- SCALEXIO Real-Time PC  
DS2502 IOCNET Link Board の CPLD バージョンが 4.0 以上、かつ FPGA バージョンが 5.1 以上であることが必要です。以前にロードしたファームウェアバージョンが 5.1 以上であること、つまりインストールされている Firmware Archives のセットアップバージョンが 3.3 (dSPACE Release 2021-A) 以上であることが必要です。





# FPGA Programming Blockset

## 次のステップ

## 本章の内容

FPGA Programming Blockset 2022-B の新機能.....	97
FPGA Programming Blockset 2022-B への移行.....	100

## FPGA Programming Blockset 2022-B の新機能

### Xilinx®のサポートの拡張

FPGA Programming Blockset で、Xilinx 設計ツールの以下の製品とバージョンがサポートされるようになりました。

Xilinx 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
Vivado 2022.1 <sup>2)</sup>	<ul style="list-style-type: none"> <li>MATLAB R2021a</li> <li>MATLAB R2021b</li> </ul>	<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。例外：FPGA Programming Blockset-Processor Interface のみ、Windows 11 をサポートしています。Windows 11 は、Xilinx をサポートしていないため、FPGA アプリケーションのモデリングやハンドコーディングには使用できません。</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>」(236 ページ)を参照してください。Windows Server オペレーティングシステムは、Xilinx 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。</p>

1) FPGA Programming Blockset の Processor Interface サブライブラリは、MATLAB R2022a および R2022b もサポートしています。

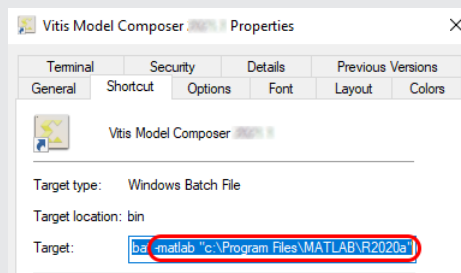
2) 一般的に、Vivado ML Enterprise エディションと Xilinx® Vitis™ Model Composer が必要です。Xilinx Vitis Model Composer 付属の Vivado ML Standard エディションは、DS2655 (7K160) および DS6601 FPGA Base Board でも使用することができます。Xilinx Vitis Model Composer は、RTI FPGA Programming Blockset を使用して FPGA アプリケーションをモデリングする場合にのみ必要です。

Vivado 2021.1 以降、Xilinx System Generator for DSP は Vitis Model Composer に含まれています。MATLAB/Simulink のための統一

された Xilinx アドオンです。この変更に関連するライセンスの問題については、<https://xilinx.com/support/answers/76039.html> を参照してください。

### ヒント

Vitis Model Composer を特定の MATLAB バージョンで使用するには、`-matlab "<matlabdir>"` パラメータを指定する必要があります。このパラメータは、たとえばデスクトップ上の Vitis Model Composer アイコンに追加することができます。



### MicroAutoBox III/SCALEXIO フレームワークの機能強化

**プロセッサ信号の前処理と後処理** SCALEXIO および MicroAutoBox III のブロックセットは、FPGA モデルの一部としてスケールングサブシステムをサポートするようになりました。スケールングサブシステムにより、FPGA とリアルタイムプロセッサ間での信号の前処理と後処理を指定することができます。スケールングサブシステムは、FPGA モデルの一部ですが、リアルタイムプロセッサで実行されます。スケールングサブシステムは、次の点に使用することができます。

- 信号をスケールングする。
- 複雑なアルゴリズムを実装して信号を処理する。たとえば、複数のプロセッサ出力信号から FPGA の入力信号を算出することができます。
- 信号をグループ化してわかりやすくする。

詳細については、「[プロセッサ通信のプリプロセス処理とポストプロセス処理の基礎](#)」(『FPGA Programming Blockset ガイド』)を参照してください。

**ビルド結果ファイル用の新しいコンテナファイル形式** FPGA Programming Blockset は、ビルド結果のファイル形式として FPGAC をサポートするようになりました。FPGAC ファイルにより、FPGA アプリケーションをリポジトリで管理することができます。FPGAC ファイルを ConfigurationDesk の検索パスに追加するか検索パスを更新すると、Functions ブラウザに FPGA カスタムファンクションブロックタイプが追加され、ConfigurationDesk アプリケーションを開くかカスタムファンクションをロードした場合にインスタンス化されたファンクションブロックが更新されます。

詳細については、「[FPGAC ファイルで提供される FPGA カスタムファンクションブロックタイプを追加する方法](#)」(『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。

**MicroAutoBox III/SCALEXIO 用の新しいスクリプト機能** スクリプトインターフェースは、既存の FPGAC ファイルを更新するスクリプト関数によって拡張され、たとえば、新しい FPGA アプリケーションをビルドせずに、変更したスケールングサブシステムを実装できるようになりました。

詳細については、「[変更されたスケールングサブシステムで FPGAC ファイルを更新する方法](#)」(『FPGA Programming Blockset ガイド』)を参照してください。

## SCALEXIO フレームワークの機能強化

**プロセッサ通信の 64 ビット固定小数点のサポート** Register64 および Buffer64 アクセスタイプのバイナリ位置のないデータタイプ (Fix\_64\_0 と UFix\_64\_0) は、64 ビット固定小数点のフル分解能をサポートするようになりました。

他のすべての 64 ビット固定小数点データタイプは、引き続き double に変換され、その固定小数点の分解能は 53 ビットに制限されます。これは Bus アクセスタイプにも適用されます。

### 注記

生成されたプロセッサインターフェースブロックのデフォルトのデータタイプは常に *double* です。64 ビット固定小数点データタイプの完全な分解能を使用するには、生成されるブロックのデータタイプを *int64* または *uint64* に設定する必要があります。「[64 ビットの固定小数点データを最大分解能で転送する方法 \(SCALEXIO\)](#)」(『FPGA Programming Blockset ガイド』[📖](#))を参照してください。

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

**ビルド結果のバックアップ** FPGA アプリケーションをビルドする前に、ビルドプロセスで生成および使用される FPGA アプリケーション、モデル、およびレポートの自動バックアップを有効にして設定できるようになりました。ビルドプロセスの終了後、MATLAB コマンドウインドウにバックアップフォルダへのリンクが表示されます。

自動バックアップの設定については、「[ビルド結果のバックアップ](#)」(『FPGA Programming Blockset ガイド』[📖](#))を参照してください。

**モデルとビルド結果に関する新しいレポート** このフレームワークでは、FPGA ビルドの最後にモデルとビルドのレポートを生成して提供するようになりました。

モデルレポートには、使用した FPGA チャンネル、ソフトウェア、ツールに関する情報が含まれます。ビルドレポートには、FPGA の使用率とビルド期間に関する情報が含まれます。

レポートへのリンクは、ビルドプロセスの終了後に MATLAB コマンドウインドウに表示されます。レポートは、ビルドフォルダまたはバックアップフォルダにあります。

**[word length calculator]ダイアログ** [word length calculator]を使用して、ユーザ定義の浮動小数点値に対して最適な固定小数点データタイプを計算できるようになりました。「[Word Length Calculator](#)」(『FPGA Programming Blockset - FPGA Interface Reference』[📖](#))を参照してください。

## FPGA Build Monitor の機能強化

FPGA Build Monitor 2022-B では、ビルドプロセス中に警告メッセージが発行された場合に、それを表示するようになりました。

詳細については、「[FPGA Build Monitor 2022-B の参考情報](#)」(『FPGA Programming Blockset ガイド』[📖](#))を参照してください。

## FPGA Programming Blockset 2022-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 以降の FPGA アプリケーションを実装しており、FPGA Programming Blockset 2022-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。

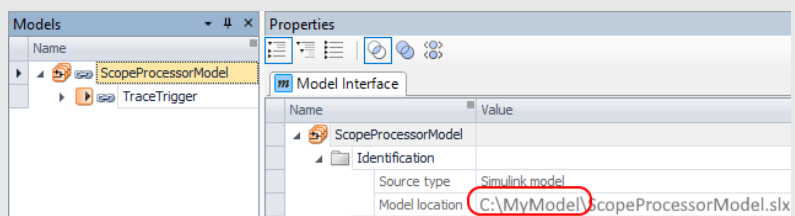
**FPGA スコープラスタの生成プロセスの変更** ConfigurationDesk で FPGA Programming Blockset 3.13 (Release 2022-A) を使用してビルドされた FPGA カスタムファンクションを使用する場合、ControlDesk で使用される FPGA スコープラスタを生成する必要があります。

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。

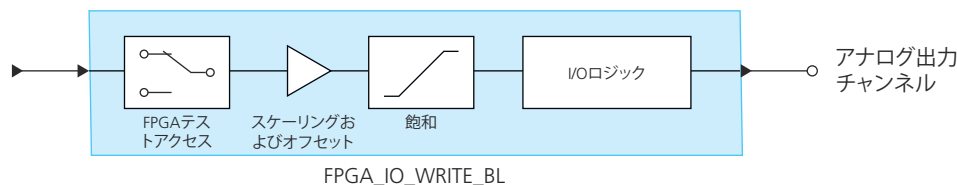
#### 注記

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リングします。

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リングし、その後、スケーリングされた値を 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。

**移行されるプロセッサインターフェースの表示** FPGA Programming Blockset 3.4 ...3.8 では、Processor Interface サブライブラリのプロセッサインターフェースブロックを使用して SCALEXIO システム/MicroAutoBox III のプロセッサインターフェースをモデル化していました。

Processor Interface サブライブラリのプロセッサインターフェースブロックを使用したモデルを移行すると、アップデートプロセスでこれらのブロックが Model Interface Package for Simulink のモデルポートブロックに移行されます。下の図に例を示します。

移行後の表示	移行前の表示

## 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 22.2 の新機能

---

### DS1403 Processor Board

DS1403 Processor Board は、データロギング設定を有効にしたリアルタイムアプリケーションの使用をサポートするようになりました。ControlDesk を使用して、データロギング設定と関連するリアルタイムアプリケーション (RTA) ファイルを MicroAutoBox III のフラッシュメモリにロードする必要がなくなりました。その結果、ControlDesk PC をリアルタイムハードウェアに接続して、データロガー設定をロードする必要がなくなりました。

詳細については、「データロギング」(『ControlDesk 計測および記録』)を参照してください。

### DS1521 Bus Board

DS1521 Bus Board が RS485 の全二重モードに対応。





# Model Compare

## 次のステップ

## 本章の内容

Model Compare 2022-B の新機能.....	105
Model Compare 2022-B への移行.....	107

## Model Compare 2022-B の新機能

### Difference Overview の改善

Model Compare 2022-B では、[Difference Overview]ダイアログが改善され、変更されたモデル部分の概要を簡単に確認できるようになりました。差異のある子エレメントを含むサブシステムは、太字で表示されるようになり、より簡単に識別することができます。[Difference Overview]ダイアログは、SLX ファイルの他に、MATLAB R2021b で新しく追加された OPC (Open Packaging Conventions) MDL ファイルもサポートするようになりました。

#### 関連ドキュメント

- 「Difference Overview」(『Model Compare リファレンス』)

### ワークフロー統合の改善

Model Compare 2022-B では、ワークフロー統合が改善されています。

- バッチモードでは、ダイアログは表示されません。
- バージョン管理システム用の CMD プロキシスクリプトが提供されます。
- Comfort Copy の間、ライブラリのロックは自動的に解除されます。
- Comfort Copy でマスクパラメータがサポートされます。
- サブシステムの追加/削除されたエレメントは、デフォルトで差分レポートにリストされます。
- textconv ドライバを使用すると、SLX や OPC MDL ファイルの Git 差分や Blame により、すばやく読みやすい結果を得ることができます。The ModelCompare.VCSTextConv.cmd は、Model Compare のインストールフォルダにあります。詳細については、dSPACE サポートにお問い合わせください。

### 関連ドキュメント

- 「ModelCompare [オプション] API コマンド」(『Model Compare リファレンス』📖)
- 「Model Compare 2022-B への移行」(107 ページ)

### フィルタリングの改善

Model Compare 2022-B で、Block Type および Property フィルタのフィルタリングが改善されました。「\*」や「?」などのワイルドカードを使用することで、より簡単かつ柔軟にフィルタを設定できるようになりました。

### 関連ドキュメント

- 「[Block Type Filter]ページ」(『Model Compare リファレンス』📖)
- 「[Property Filter]ページ」(『Model Compare リファレンス』📖)

### 大きなバスブロックのサポートの改善

Model Compare 2022-B では、Bus Creator ブロックなどの大きなバスブロックの変更を、比較結果や差分レポートで容易に確認できるようになりました。

### 関連ドキュメント

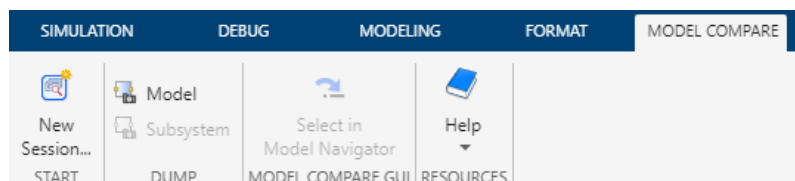
- 「バス信号の表示」(『Model Compare リファレンス』📖)

### 新しいチュートリアル動画

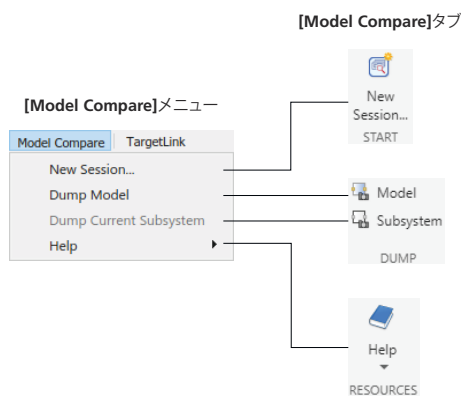
Model Compare では、日々の作業をサポートするために、いくつかのチュートリアル動画を用意しています。最新のチュートリアル動画「Model Compare3 者間マージ (4:17)」📺dSPACE Help を参照してください。

### Simulink ツールストリップの新しい[Model Compare]タブ

Model Compare 2022-B では、Simulink ツールストリップに新しい[Model Compare]タブが導入されました。



MATLAB R2021b 以前では、以下のコマンドは[Model Compare]メニューで使用することができました。MATLAB R2022a 以降では、これらのコマンドは Simulink ツールストリップの新しい[Model Compare]タブで使用できるようになりました。



[Model Compare]メニュー (MATLAB R2021b 以前)	[Model Compare]タブ (MATLAB R2022a 以降)
New Session	START - New Session
Dump Model	DUMP - Model
Dump Current Subsystem	DUMP - Subsystem
Help - dSPACE Help	RESOURCES - Help - Help
Help - Useful Links - dSPACE Website	RESOURCES - Help - dSPACE Website
Help - Useful Links - TargetLink Support Center	RESOURCES - Help - dSPACE Support Request
-	MODEL COMPARE GUI - Select in Model Navigator
-	RESOURCES - Help - New Features and Migration
-	RESOURCES - Help - Using dSPACE Help

## Model Compare 2022-B への移行

### 差分レポートに関する変更

Model Compare 2022-B では、差分レポートの表示内容が若干変更されました。追加または削除されたサブシステムの要素がデフォルトで表示されるようになりました。たとえば、プルリクエスト時に差分レポートを自動生成して、変更点を確認することができます。「[Change is the Only Constant. Continuously Reviewing Binary Resources via Pull Requests](#)」を参照してください。

以前の動作に戻すには、[Comparison Settings]ダイアログの[Basic Filters]ページで[Ignore content changes of subsystems without a corresponding system]チェックボックスをオンにします。「[\[Basic Filters\]ページ](#)」(『[Model Compare リファレンス](#)』)を参照してください。

#### 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 Container Utility

## モデルコンテナユーティリティについて

### 概要

モデルコンテナユーティリティを使用すると、既存のモデルインプリメンテーションコンテナとパッケージコンテナ (PKC ファイル) を修正することができ、生成し直す必要がなくなります。


### dSPACESIL テストツールチェーンの補完

モデルコンテナユーティリティは、dSPACE SIL テストツールチェーンを補完するものです。特に、OEM とサプライヤ間でモデルコンテナを交換するようなワークフローに有効です。

- このユーティリティを使用して、IP 関連のファイルを既存のモデルインプリメンテーションコンテナまたはパッケージから除外することで、*知的財産権を保護*することができます。
- このユーティリティを使用して、既存のモデルインプリメンテーションコンテナ やパッケージコンテナを *事前にコンパイル*することができます。また、それぞれの一部分、つまりパッケージのみをコンパイルすることもできます。
- モデルコンテナユーティリティのコマンドラインインターフェースにより、*一連のビルドシナリオの一部として、多数のユーザ向けのコンテナやパッケージを簡単にコンパイル*することができます。

### サポートされるコンテナタイプとシミュレーションプラットフォーム

- モデルコンテナユーティリティは現在、以下のコンテナタイプをサポートしています。
  - V-ECU インプリメンテーションコンテナ (VECU) ファイル
  - パッケージコンテナ (PKC) ファイル
- モデルコンテナユーティリティは現在、VEOS シミュレーションプラットフォームでのコンテナコンパイルをサポートしています。

詳細については、「[Basics on the Model Container Utility](#)」(『[Model Container Utility Manual](#)』)を参照してください。



# Model および Sensor Interface Blockset

## 次のステップ

## 本章の内容

Model and Sensor Interface Blockset 2022-B の新機能..... 111

Model and Sensor Interface Blockset 2022-B への移行..... 112

## Model and Sensor Interface Blockset 2022-B の新機能

### Model and Sensor Interface Blockset

dSPACE Release 2020-B 以降、SIL (Software-in-the-Loop) または HIL (Hardware-in-the-Loop) テスト環境においてセンサを使用したシミュレーションを行う場合、Model and Sensor Interface Blockset (MSI ブロックセット) を使用する必要があります。

Sensor Simulation では、シミュレーションデータの計算、ビジュアル表示用に MotionDesk PC へのデータの送信、接続されたセンサのセンサ構成と生データを生成する SensorSim アプリケーションへのデータの送信のためにこのブロックセットが必要となります。

データは、TCP/IP プロトコルを使用して連続するデータエレメントのデータストリームとして送信されます。Simulink モデルでは、このブロックセットを Ethernet 機能に接続して選択したサポートされているシミュレーションプラットフォームと通信し、TCP/IP プロトコルを使用して Sensor Simulation システムおよびアプリケーションに接続する必要があります。

シミュレーションの各ムービングオブジェクトでは、シミュレーションデータオブジェクトは親子関係を使用して単一のオブジェクトブロック内のキネマティックチェーン (運動学的連鎖) として定義することができます。接続されたシステムでステータスや統計情報を読み出すためにブロックを追加することもできます。

Environment Sensor Interface Unit を接続することができます。このユニットは、SensorSim アプリケーションから得られたセンサの生データを分割し、適切なセンサハードウェア、たとえば、カメラセンサのイメージセンサインターフェースに挿入します。Simulink モデルにブロックを追加して、接続されたセンサのフィードバックを

Environment Sensor Interface Unit から受け取ることができます。また、特定のセンサに関してセンサの欠陥(たとえば、カメラセンサでのピクセルエラーなど)を設定することもできます。

Model および Sensor Interface Solution を使用して ASM モデルを調整した場合は、これらのモデルを移行して Model and Sensor Interface Blockset を使用することができます。

Model and Sensor Interface Blockset の機能の詳細については、「[Model and Sensor Interface Blockset の機能](#)」([『Model and Sensor Interface Blockset マニュアル』](#) 〇〇)を参照してください。

## 新機能

Model and Sensor Interface Blockset 2022-B に以下の新機能が追加されました。

- VEOS 用の TCP Server 接続: SIL (Software-in-the-Loop) シミュレーションで VEOS シミュレーションプラットフォームに Sensor Simulation システムとアプリケーションを接続するために、Utilities サブシステムには Ethernet TCP Server ブロックが用意されています。
- Simulink シミュレーション: 通常モードおよび加速モードでの Simulink シミュレーションがサポートされています。
- AURELIO の ASM Traffic デモのサポート: Simulink、VEOS、SCALEXIO 用の ASM Traffic デモには、Model and Sensor Interface Blockset Animation Interface が含まれており、ModelDesk プロジェクトで使用することで AURELION によるアニメーションとセンサシミュレーションをサポートすることができます。SCALEXIO ハードウェア固有の設定の要件を含む、ASM Traffic Demo Model の新機能の詳細については、「[ASM Traffic デモモデルの変更](#)」(41 ページ)を参照してください。

## 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』](#) 〇〇)  
[MotionDesk 2022-B の新機能](#)..... 119  
[Simulink モデルの適合](#) ([『Model and Sensor Interface Blockset マニュアル』](#) 〇〇)

## Model and Sensor Interface Blockset 2022-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 2022-B の新機能.....	115
ModelDesk 2022-B への移行.....	115

## ModelDesk 2022-B の新機能

### Road Generator

**トラフィックオブジェクト**    トラフィックオブジェクトは、絶対 x、y、z 座標を使用して道路エレメントやジャンクションに配置することができます。

## ModelDesk 2022-B への移行

### プロジェクトの移行

ModelDesk 2022-B 以降では、ModelDesk バージョン 5.0 (dSPACE Release 2018-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。

### 移行したプロジェクトでの Release 2020-B の ASM の使用

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。

### ASM プロジェクト

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 2022-B の新機能..... 117

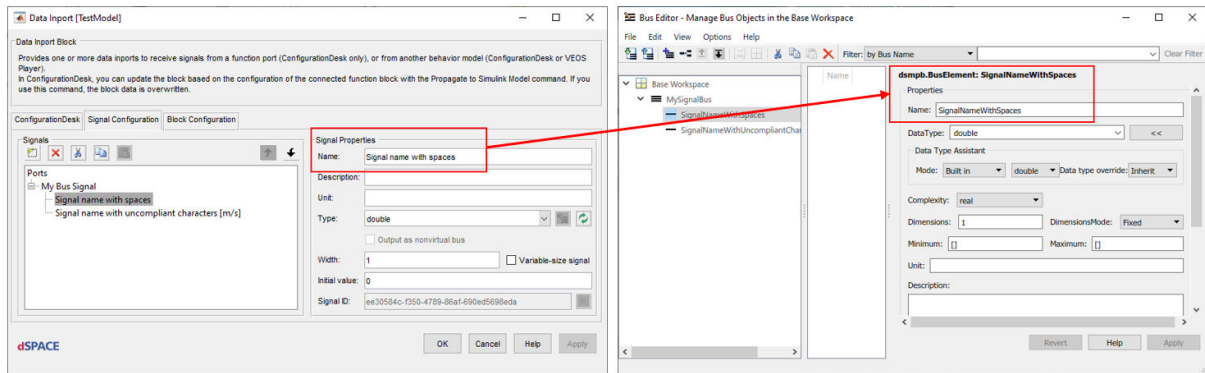
Model Interface Package for Simulink 2022-B への移行..... 118

## Model Interface Package for Simulink 2022-B の新機能

### Simulink.Bus オブジェクトでの非 標準拠文字を含む信号名のサポ ート

Simulink.Bus オブジェクトを作成する場合、Model Interface Package for Simulink では、スペースやブラケットを含む信号名や 63 文字を超える信号名など、有効な C 識別子または MATLAB 識別子ではない信号名をサポートしています。モデルポートブロックダイアログと ConfigurationDesk のモデルポートブロックでは、信号名は変更されません。このため、信号名に非標準拠文字を含む構造化された信号を使用して、バス設定に対して伝播操作やモデルポートの自動マッピングを行うことができます。

モデルポートブロックダイアログでこのような信号名を指定した場合、Model Interface Package for Simulink は、Simulink.BusElement オブジェクトの代わりに、MATLAB 変数の命名規則に適合した名前を使用する dsmpb.BusElement オブジェクトを提供します。次の図を参照してください。



**注記**

63 文字を超える信号名は、63 文字目で切り捨てられます。

**MATLAB の互換性**

Model Interface Package for Simulink 2022-B は、以下の MATLAB®リリースと互換性があります。

- MATLAB R2022b
- MATLAB R2022a
- MATLAB R2021b
- MATLAB R2021a

**Model Interface Package for Simulink 2022-B への移行**

**移行**

Model Interface Package for Simulink 2022-B では、移行作業は不要です。

# MotionDesk

## 次のステップ

## 本章の内容

MotionDesk 2022-B の新機能.....	119
MotionDesk 2022-B への移行.....	119

## MotionDesk 2022-B の新機能

### MotionDesk

MotionDesk 2022-B には新機能はありません。

## MotionDesk 2022-B への移行

### レーダー、LiDAR、レーザーセンサを使用したセンサシミュレーションの廃止

MotionDesk 2022-B 以降では、レーダー、LiDAR、レーザーセンサを使用した MotionDesk をベースにしたセンサシミュレーション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/AS 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新製品で、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 および Sensor Simulation

- MotionDesk をベースにしたセンサシミュレーション  
すべてのセンサタイプに対する MotionDesk をベースにした Sensor Simulation は、Release 2022-B を最後に廃止されます。詳細については、

<https://www.dspace.com/en/pub/home/support/pli/elas/elasw/elasensim.cfm> を参照してください。

- MotionDesk

MotionDesk は、Release 2023-B を最後に廃止されます。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、最初に Release 22-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 22-B 以降で COLLADA ファイルをインポートして使用することができます。

Library Manager Tool の Import Objects Dialog で、ファイルを変換します。

---

#### 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 2022-B の新機能.....	121
Real-Time Testing 2022-B への移行.....	122

## Real-Time Testing 2022-B の新機能


### データストリーミング

SCALEXIO プロセッサユニット、DS6001 Processor Board、MicroAutoBox III のみ：  
ホスト PC とリアルタイムシステム間のデータストリームに使用するバッファのサイズを指定することができます。これにより、デフォルトのバッファサイズでメモリ不足のエラーが発生する場合、より多くのデータをストリームすることができます。

### MicroAutoBox III の互換性が向上

MicroAutoBox III プラットフォームでは、ホスト PC とリアルタイムプラットフォーム間の互換性が拡張されています。最新バージョンの Real-Time Testing がインストールされているホスト PC から、ファームウェアがバージョン 6.0 以降 (Release 2022-A でリリースされ、Real-Time Testing バージョン 5.2 に含まれる) の MicroAutoBox III プラットフォームを管理することができます。

### ドキュメント

新しい Real-Time Testing Tutorial は、Real-Time Testing の最初の一步を踏み出すためのガイドです。『[Real-Time Testing Tutorial](#)』を参照してください。

## Real-Time Testing 2022-B への移行

---

### 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。

Real-Time Testing 4.0 以前のバージョンで生成された BCG ファイルは、Real-Time Testing 2022-B では使用することができません。RTT シーケンスの BCG ファイルを再度作成する必要があります。

SCALEXIO では Real-Time Testing 4.4 以降、VEOS では Real-Time Testing 4.2 以降のみで、次の移行を行う必要があります内部 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.....	123
RTI/RTI-MP および RTLib の移行上の注意点.....	123

## 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® R2022b 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。

**注記**

dSPACE Release 2013-B 以前を使用して MAT ファイルで保存した Simulink.ConfigSet オブジェクトを Release 2022-B に移行することはできません。

# RTI Bypass Blockset


## RTI Bypass Blockset 2022-B への移行

### バーチャル ECU でのオンターゲットバイパス処理のサポート終了

バージョン 2022-B では、RTI Bypass Blockset では、次の dSPACE ハードウェア上のバーチャル ECU (V-ECU) でのオンターゲットバイパス処理はサポートされなくなりました。

- SCALEXIO システム
- MicroAutoBox III

### 追加の移行手順が必要な場合

RTI CAN MultiMessage Blockset 3.18 より前のバージョンから RTI Bypass Blockset 2022-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 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 3.x および 2.x と互換性のある RTI Bypass Blockset 2022-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 2022-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 2022-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] をクリックした時に行われます。RTI Bypass Blockset 2022-

B 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 3.18 までのモデルを使用する場合*

RTI Bypass Blockset 2.6 から RTI Bypass Blockset 3.18 まででビルドされた Simulink モデルを RTI Bypass Blockset 2022-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 3.18 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

## RTI CAN MultiMessage Blockset 2022-B への移行

### 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 5.8 より前のバージョンから RTI CAN MultiMessage Blockset 2022-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

## RTI LIN MultiMessage Blockset 2022-B への移行

### 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 2022-B の新機能

---

### STBM\_SYNCHRONOUS\_TRIGGER ブロック

新しい STBM\_SYNCHRONOUS\_TRIGGER ブロックでは、dSPACE ECU タイムベースマネージャの時間に基づいてトリガを生成し、これを使用してイベントを設定することができます。また、このブロックは、基本スケジュールテーブルの GTS 同期ステータスの情報も提供します。

「STBM\_SYNCHRONOUS\_TRIGGER」(『RTI Synchronized Time Base Manager Blockset Reference』[📖](#))を参照してください。



# SCALEXIO Firmware

## 次のステップ

## 本章の内容

SCALEXIO Firmware 22.2 の新機能.....	133
SCALEXIO Firmware 22.2 への移行.....	134
SCALEXIO ファームウェアの廃止.....	134

## SCALEXIO Firmware 22.2 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。

#### ▪ DS6330M4 車載 Ethernet モジュール

DS6330M4 モジュールは、転送速度 100Mbit/s および 1000Mbit/s の車載 Ethernet に対応しており、MACsec (メディアアクセス制御セキュリティ) 機能を内蔵しています。ただし、ハードウェアは MACsec に対応していますが、この機能を利用するにはソフトウェアパッチをインストールする必要があります。

DS6330M4 モジュールは、車載 Ethernet 規格に対応した Ethernet ネットワーク接続用ポートを 2 つ搭載しています。DS6333-CS、DS6333-PE、および DS6335-CS Ethernet ボードで最大 2 個のモジュールを使用することができます。

### Rs485 全二重

DS2671 Bus Board と DS6321 UART Board の UART は、RS485 全二重をサポートしています。



### データロギング

DS6001 Processor Board は、データロギング設定を有効にしたリアルタイムアプリケーションの使用をサポートするようになりました。ControlDesk を使用して、データロギング設定と関連するリアルタイムアプリケーション (RTA) ファイルを DS6001 のフラッシュメモリにロードする必要がなくなりました。その結果、ホスト PC をリアルタイムハードウェアに接続して、データロガー設定をロードする必要がなくなりました。

詳細については、「データロギング」(『ControlDesk 計測および記録』)を参照してください。

## 関連トピック

### 基本概要

[Ethernet Board Configuration]ページ (『SCALEXIO ハードウェアの設置および設定』)  
 [Ethernet Board Configuration]ページ (『SCALEXIO ハードウェアの設置および設定』)

## SCALEXIO Firmware 22.2 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 メインボード搭載の SCALEXIO プロセッサユニット (Real-Time PC Version 1.0) は、SCALEXIO ファームウェアでサポートされていません。

# センサシミュレーション

## 次のステップ

## 本章の内容

Sensor Simulation 2022-B の新機能.....	135
Sensor Simulation 2022-B への移行.....	135

## Sensor Simulation 2022-B の新機能

### SensorSim アプリケーション

Sensor Simulation 2022-B の SensorSim アプリケーションには、新機能はありません。

### 関連トピック

#### 基本概要

[MotionDesk Sensor Simulation Control](#)  
[センサシミュレーション Manual](#)  
[センサシミュレーション Overview](#)

## Sensor Simulation 2022-B への移行

### レーダー、LiDAR、レーザーセンサを使用したセンサシミュレーションの廃止

Sensor Simulation Release 2022-B では、レーダー、LiDAR、およびレーザーセンサを使用した MotionDesk ベースのセンサシミュレーション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/AS 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新製品で、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 および Sensor Simulation

- MotionDesk をベースにしたセンサシミュレーション  
すべてのセンサタイプに対する MotionDesk をベースにした Sensor Simulation は、Release 2022-B を最後に廃止されます。詳細については、  
<https://www.dspace.com/en/pub/home/support/pli/elas/elasw/elasensim.cfm> を参照してください。
- MotionDesk  
MotionDesk は、Release 2023-B を最後に廃止されます。詳細については、  
<http://www.dspace.jp/go/elamd> を参照してください。



# SYNECT

---

## 次のステップ

## 本章の内容

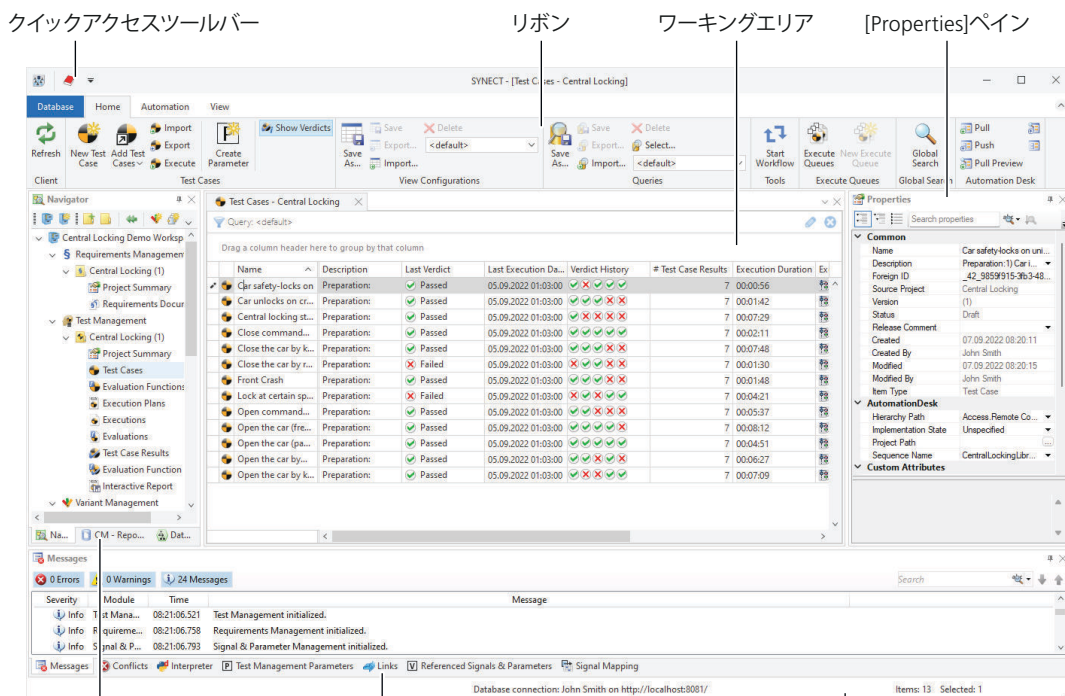
SYNECT 2022-B の新機能.....	138
SYNECT 2022-B への移行.....	141

## SYNECT 2022-B の新機能

## 新機能

最新のユーザインターフェースフレームワーク

SYNECT は、ページとペインの処理が改善された最新のユーザインターフェースフレームワークを提供するようになりました。次の図を参照してください。



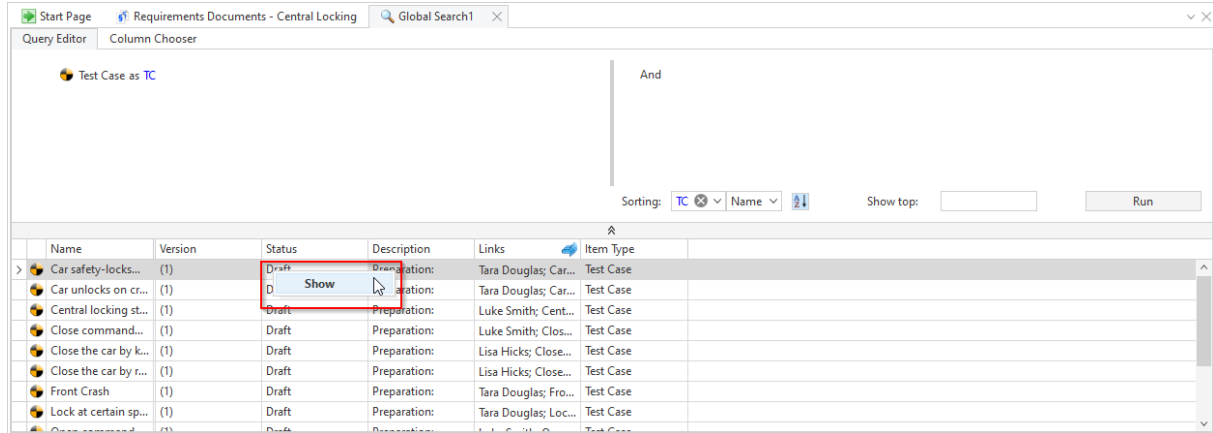
- ナビゲータ
- [Data Pools]ペイン
- CM Repository Browser

- Message Viewer
- [Conflicts]ペイン
- Interpreter
- [Test Management Parameters]ペイン
- [Links]ペイン
- [Signal Mapping]ペイン
- [Referenced Signals & Parameters]ペイン

ステータスバー

## グローバル検索の改善

検索クエリで見つかったアイテムに、結果アイテムのコンテキストメニューを使用して移動できるようになりました。次の図を参照してください。



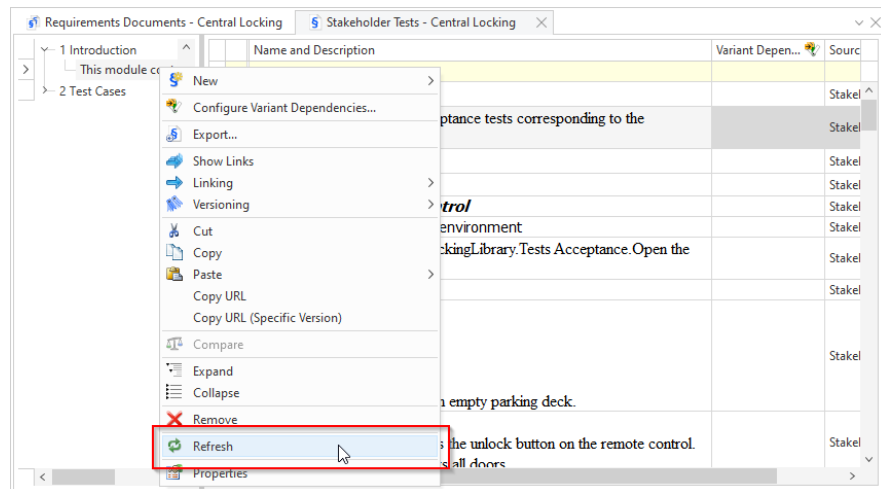
SYNECT は、データグリッド (Test Cases データグリッドなど) でアイテムを開きます。

## アイテムのリフレッシュ機能の改善

特定のアイテムをリフレッシュすることができるようになりました。これにより、データグリッドのパフォーマンスが向上しています。

特定のデータグリッド (Requirements Management Documents データグリッドなど) のアイテムのコンテキストメニューには、Refresh コマンドが用意されています。

要件の内容の更新を示す次の図を参照してください。



**SYNECT クライアント API** SYNECT クライアント API の Application インターフェースには、ID で指定したアイテムのリストをリフレッシュできる RefreshItems メソッドが用意されています。

詳細については、「[Application / ISnApplication <<Interface>>](#)」(『SYNECT Client API Reference』)を参照してください。

---

### SYNECT サーバ API によるアイテム URL のサポート

サーバ API を使用して、特定のアイテムタイプの SYNECT アイテムの URL を生成できるようになりました。この URL から、SYNECT クライアントの各アイテムに移動することができます。これを使用して、SYNECT のアイテムをリンクさせるレポートやカスタムアドオンの拡張機能を実装することができます。

ItemBase クラスは、FindSynectUrls メソッドを実装しています。このメソッドは、プロジェクト、テストケース、実行、モデルなど、ItemBase クラスから継承した各データベースアイテムで利用可能です。このメソッドは、URL のリストを返します。詳細については、「ItemBase」(『SYNECT Server API Reference』[📖](#))を参照してください。

---

### コンテキストメニューの拡張

SYNECT は、SYNECT ユーザーインターフェースのペインから利用できるアイテムのコンテキストメニューを拡張するためのクライアントイベントを提供します。

選択したアイテムの発信リンクと着信リンクを表示する、Links ペインのアイテムのコンテキストメニューを拡張できるようになりました。表示されているリンクアイテムのコンテキストメニューに、カスタムコマンドを追加することができます。

詳細については、「SYNECT クライアントのコンテキストメニューの拡張」(『SYNECT ガイド』[📖](#))を参照してください。

---

### SYNECT クライアント API の改善

SYNECT クライアント API では、ユーザーインターフェースのペインを開く、アイテムを選択する、ファイルをインポートまたはエクスポートするなどのクライアント操作を自動化することができます。

SYNECT クライアント API を使用して、2 つのアイテムの差分を表示する比較ビューを開くことができるようになりました。

詳細については、「CompareView / ISnCompareView <<Interface>>」(『SYNECT Client API Reference』[📖](#))を参照してください。

---

### 要求管理ドキュメントのインポートの改善

要求管理ドキュメントの一部をインポートできるようになりました。新しい DeleteUnspecifiedItems ECXML プロパティを使用して、プラグインのインポート動作を制御することができます。

有効にした場合、プラグインによって完全な要件ドキュメントがインポートされます。インポートされたドキュメントで利用できない要件内容は削除されます。これは以前の SYNECT バージョンでよく知られている動作であり、たとえば、DOORS のドキュメントをインポートする際に役立ちます。

無効にした場合、インポートプラグインによって要件ドキュメントの構造は変更されません。利用可能な要件内容が更新され、新しい要件内容が追加されます。新しい動作では、要件内容のリストをインポートすることができます。たとえば、Azure DevOps からアイテムをインポートしたり、Atlassian Jira から問題をインポートしたりする場合に役立ちます。

# SYNECT 2022-B への移行

## 次のステップ

## 本章の内容

SYNECT 2.13 からの移行.....	141
SYNECT の新バージョンにアップデートするための移行手順。	
データベースの移行.....	142
以前の SYNECT バージョンのデータを SYNECT 2022-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 2.13 から SYNECT 2022-B へのデータモデルの変更.....	143
SYNECT 2.13 から SYNECT 2022-B へのデータモデルの変更はありません。	

## SYNECT 2.13 からの移行

### SYNECT クライアント自動化スクリプトの移行

Internal Python インタープリタの SYNECT クライアント API は、最新のユーザーインターフェースフレームワークで変更されました。新しい API に合わせて自動化スクリプトを調整する必要があります。Internal Python インタープリタ API の詳細については、「Python」(『SYNECT Client API Reference』[📖](#))を参照してください。

**削除された自動化インターフェースおよび列挙** 以下の自動化インターフェースと列挙が削除されました。

- FoldingType 列挙
- IPiBookmarks
- IPiFolding
- IPiFoldmarks
- IPiLineWrapping
- IPiRightEdge
- IPiTokenStyle
- IPiTokenStyles コレクション
- IPiZoom
- LineWrappingMode 列挙
- RightEdgeMode 列挙
- VisualWrappingCue 列挙

削除されたプロパティおよびメソッド 以下のインターフェースのプロパティとメソッドが削除されました。

インターフェース	削除されたプロパティ/メソッド
PythonThreads / IPythonThreads <<Interface>>	プロパティ: <ul style="list-style-type: none"> <li>▪ IDs</li> <li>▪ Count</li> <li>▪ MainThreadID</li> </ul> メソッド: <ul style="list-style-type: none"> <li>▪ Stop</li> </ul>
InterpreterProperties / IPiInterpreterProperties <<Interface>>	プロパティ: <ul style="list-style-type: none"> <li>▪ BackgroundColor</li> <li>▪ ErrorTextColor</li> <li>▪ Font</li> <li>▪ HighlightBraceMatching</li> <li>▪ Keywords</li> <li>▪ OutputTextColor</li> <li>▪ TextColor</li> <li>▪ TokenStyles</li> </ul>
EditorProperties / IPiEditorProperties <<Interface>>	プロパティ: <ul style="list-style-type: none"> <li>▪ BackgroundColor</li> <li>▪ BookmarksColor</li> <li>▪ Folding</li> <li>▪ FoldingMargin</li> <li>▪ Font</li> <li>▪ HighlightBraceMatching</li> <li>▪ HScrollVisible</li> <li>▪ Keywords</li> <li>▪ LineNumberMargin</li> <li>▪ LineWrapping</li> <li>▪ MarkerMargin</li> <li>▪ RightEdge</li> <li>▪ ShowIndentationGuides</li> <li>▪ TextColor</li> <li>▪ TokenStyles</li> <li>▪ VScrollVisible</li> </ul>
EditorControl / IPiEditorControl <<Interface>>	プロパティ: <ul style="list-style-type: none"> <li>▪ Bookmarks</li> <li>▪ CodePage</li> <li>▪ Foldmarks</li> <li>▪ Zoom</li> </ul>

## 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 2022-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 - 2.13 のデータベースを SYNECT 2022-B に移行するには、SYNECT 2022-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 2.13 から SYNECT 2022-B へのデータモデルの変更

**概要**

SYNECT 2.13 から SYNECT 2022-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 2022-B の新機能.....	146
SystemDesk 2022-B への移行.....	151

# SystemDesk 2022-B の新機能

## 次のステップ

## 本章の内容

新しい一般機能.....	146
新しい一般機能について説明します。	
新しい V-ECU 生成機能.....	148
V-ECU を生成するための新しい SystemDesk 機能。	

## 新しい一般機能

### SystemDesk 2022-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 R21-11 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 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

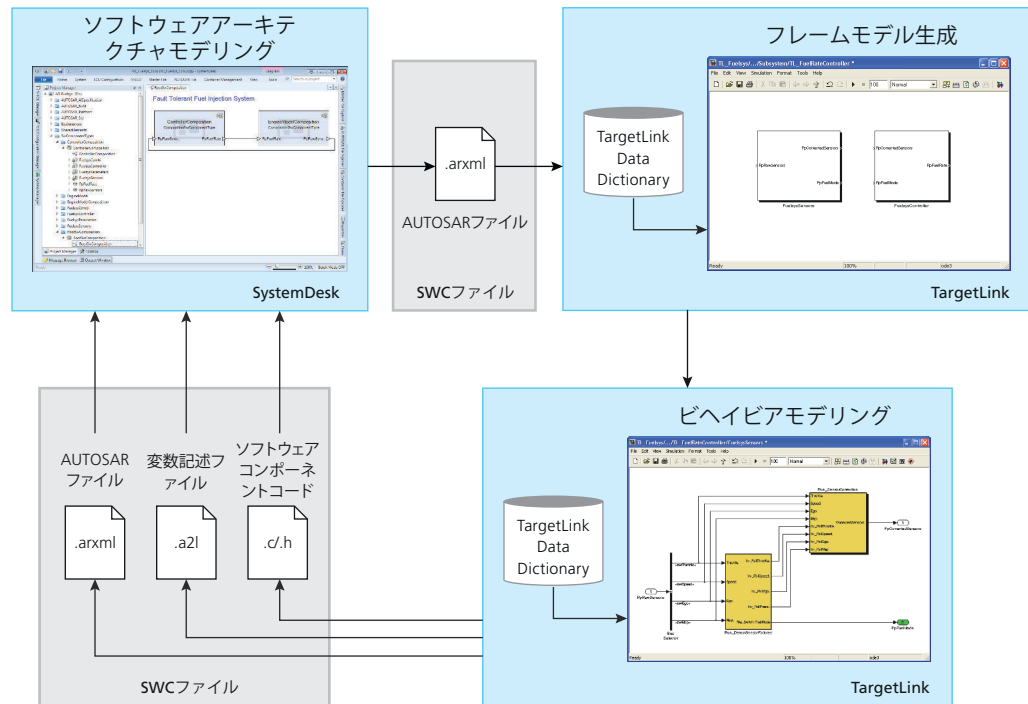
SystemDesk は、Adaptive Platform ソフトウェアを開発するために AUTOSAR R21-11 のサポートを開始しました。データ交換では、AUTOSAR R20-11、R19-11、および R19-03 がサポートされています。

### ソフトウェアコンポーネントを交換するためのベストプラクティス

AUTOSAR に準拠した ECU ソフトウェア開発で、SystemDesk はシステムレベル設計ツールとして機能します。ドメインモデリング用に、dSPACE は TargetLink AUTOSAR Module を提供しています。この使用事例に従う場合、AUTOSAR モデルの関連部分を SystemDesk から TargetLink に転送する必要があります。

dSPACE は、V-ECU および環境モデルの閉ループシミュレーションのコンテキストで、ソフトウェアコンポーネントの SIL テストをサポートします。この使用事例に従う場合、さらに AUTOSAR モデルのコードファイルを参照し、ここで実装固有の側面を指定する必要があります。これにより、SystemDesk でモデルベースの V-ECU を生成することができます。

下の図は、SystemDesk と TargetLink の相互運用性を示しています。



TargetLink と SystemDesk の間でソフトウェアコンポーネントを交換するための dSPACE ツールである *Container Manager* の廃止以降は、ベストプラクティスに従うことをお勧めします。これには、パッケージおよび AUTOSAR ファイルへの AUTOSAR エレメントの割り当て、SystemDesk および TargetLink Data Dictionary での特定のインポート/エクスポート関連の設定、交換された AUTOSAR ファイルのバージョン管理が含まれます。また、AUTOSAR ファイルの比較およびマージツールの使用をお勧めします。

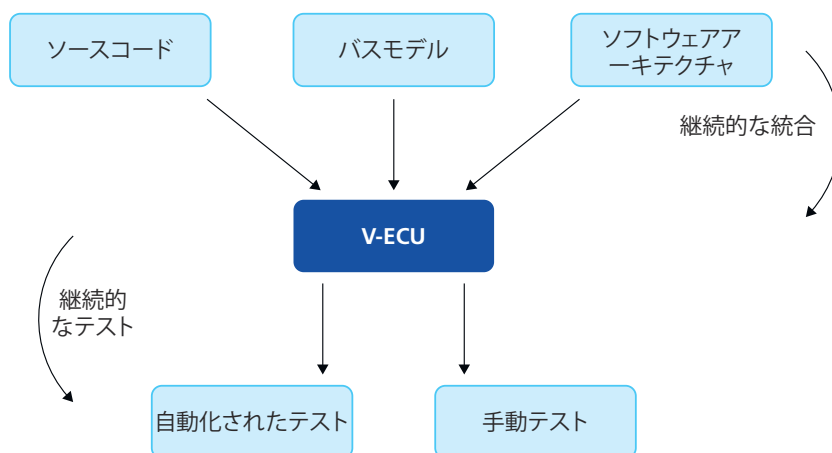
詳細については、お使いの dSPACE ツールのユーザマニュアルに記載されているベストプラクティスを参照してください。

dSPACE ツール	マニュアル
SystemDesk	「ソフトウェアコンポーネントをやり取りするためのベストプラクティス」(『SystemDesk マニュアル』📖)
TargetLink	「ソフトウェアコンポーネントをやり取りするためのベストプラクティス」(『TargetLink 相互運用およびデータ交換ガイド』📖)

## ビルドパイプラインでの SystemDesk の使用

継続的な統合や継続的なテストのために、ビルドパイプラインで SystemDesk を使用して V-ECU のインプリメンテーションを生成することができます。

この概念の特徴を示す次の図を参照してください。



詳細については、「ビルドパイプラインでの SystemDesk の使用」(『SystemDesk マニュアル』)を参照してください。

## 新しい V-ECU 生成機能

### SystemDesk 2022-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 3.2

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 2022-B を使用してバージョン 3.2 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 2022-B の互換性」(229 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。

### V-ECU FMU のサポートの改善

dSPACE Release 2022-B では、V-ECU FMU に次の機能が追加されました。

- FMI3.0 規格のサポート。
- V-ECU FMU への XCP アクセスのサポート。

V-ECU FMU に XCP サービスを統合して、dSPACE ControlDesk などの計測および適合ツールからアクセスできるようになりました。これにより、V-ECU の変数にアクセスできます。

- Linux での実行のサポート。  
32 ビットおよび 64 ビットの Linux システム用の V-ECU FMU をビルドできるようになりました。これにより、Linux 実行環境を使用して、FMI 準拠のシミュレータで V-ECU FMU を実行することができます。
- V-ECU FMU の再起動のサポート。

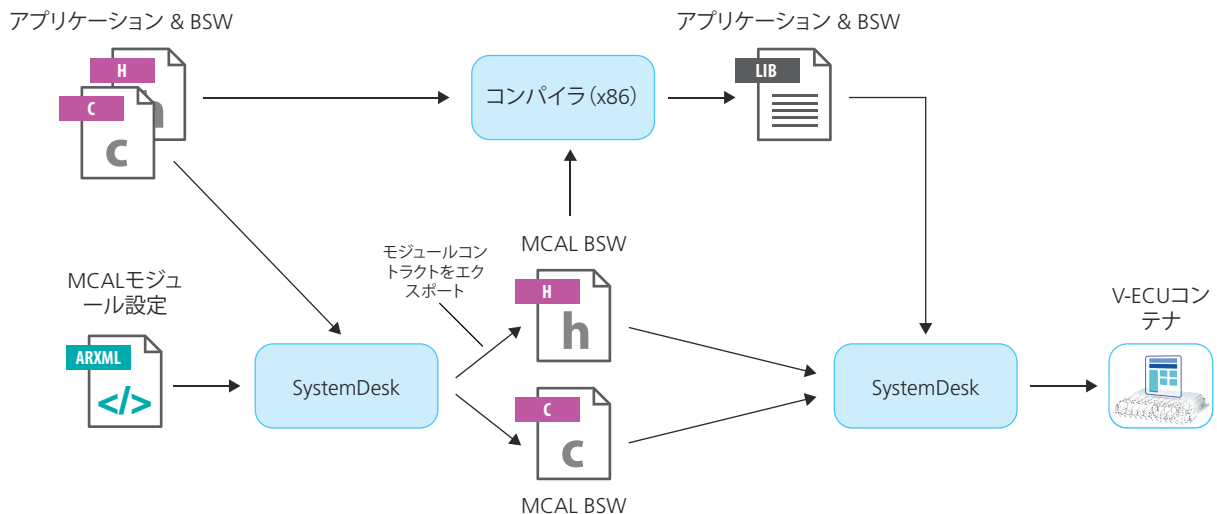
詳細については、「FMU エクスポート用の ECU コンフィギュレーション」(『SystemDesk マニュアル』)を参照してください。

## コントラクトヘッダーファイルのエクスポート

V-ECU の基本ソフトウェアを開発またはビルドする場合、dSPACE MCAL モジュールのヘッダーファイルが必要です。基本ソフトウェアをバイナリとして含める場合は、ビルドプロセス時にヘッダーファイルが必要です。

これは、ファイル数が多いなどの理由で、SystemDesk のすべてのソースコードファイルを参照することができないシナリオに適用されます。


次の図を参照してください。



SystemDesk V-ECU Manager で、生成した基本ソフトウェアのヘッダーファイルをエクスポートできるようになりました。これにより、開発用 IDE で簡単にヘッダーファイルをインクルードすることができます。

V-ECU のバイナリファイルを作成するには、以下の手順で行います。

1. 必要な dSPACE 基本ソフトウェアと開発した基本ソフトウェアを備えた V-ECU を準備します。
2. SystemDesk を使用して dSPACE ヘッダーファイルをエクスポートします。  
ヘッダーファイルのエクスポートを行う基本ソフトウェアモジュールを選択することができます。SystemDesk は、基本ソフトウェアモジュール間の依存関係を分析し、必要な追加のコントラクトヘッダーファイルをエクスポートすることができます。
3. 開発用 IDE でヘッダーファイルをインクルードします。
4. OEM サプライヤのワークフローで共有したいバイナリファイルをビルドします。

詳細については、「モジュールコントラクトのエクスポートの基礎」(『SystemDesk マニュアル』)を参照してください。

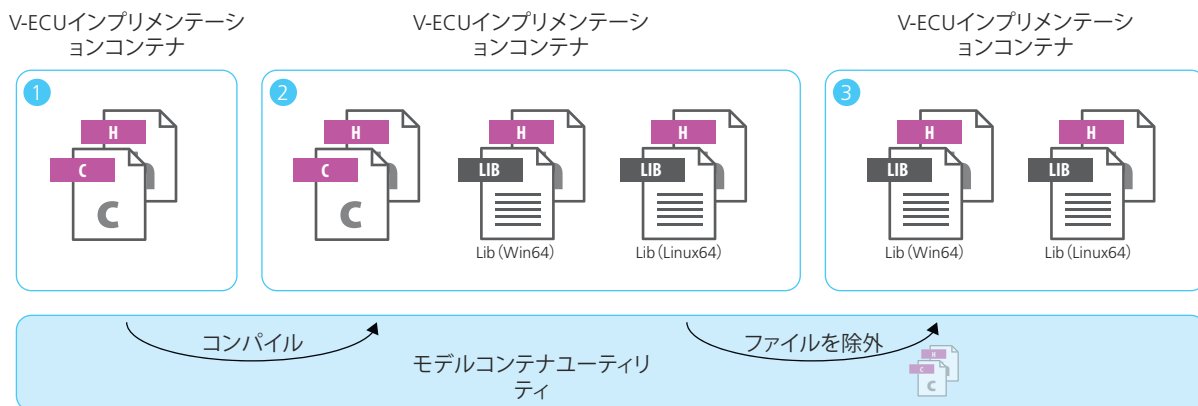
## モデルコンテナユーティリティの使用

VEOS では、既存の V-ECU インプリメンテーションコンテナ (VECU ファイル) やパッケージコンテナ (PKC ファイル) を再生成することなく変更できるモデルコンテナユーティリティを提供しています。

モデルコンテナユーティリティは、以下のような使用事例をサポートします。

- マルチユーザシナリオや継続的な統合シナリオでのビルド時間を節約するために、コンテナパッケージのソースコードを事前にコンパイルする。
- IP 保護を確保するために、ビルドされたコンテナパッケージのソースコードファイルを除外する。

次の図を参照してください。



詳細については、『Model Container Utility Manual』を参照してください。

# SystemDesk 2022-B への移行

## 次のステップ

## 本章の内容

dSPACE Release 2023-B 以降の廃止.....	151
SystemDesk 2023-B では、ソフトウェアの一部の機能が廃止されます。	
SystemDesk 2022-B への移行.....	152
SystemDesk 2022-B への移行について説明します。	

## dSPACE Release 2023-B 以降の廃止

### 廃止された部分的な AUTOSAR Import および Export コマンド

SystemDesk 2023-B では、ソフトウェアコンポーネントの以下の部分的な AUTOSAR Import および Export コマンドは使用できなくなります。

- 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lement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。

SystemDesk 2022-B でワークフローを移行することをお勧めします。

### ECU コンフィギュレーションに関するビューの廃止

dSPACE Release 2023-B 以降、ECU Configuration Manager では複数のビューはサポートされません。

AUTOSAR ビュー や Dependency ビューに切り替えることはできなくなります。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 ビューで利用可能です。

## SystemDesk 2022-B への移行

### プロジェクトの自動移行

SystemDesk2022-B では、SystemDesk 5.5 および 5.6 の SDP プロジェクトファイルは開いたときに自動的に移行されます。

#### 注記

SystemDesk 5.5 または 5.6 の最新の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 2022-B で開きます。

### SystemDesk を自動化するスクリプトの移行

SystemDesk 2022-B 以降、SystemDesk API が変更されました。SystemDesk 5.6 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 5.6 to SystemDesk 2022-B*」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

<a href="#">TargetLink 2022-B の新機能</a> .....	154
<a href="#">TargetLink 2022-B および TargetLink Data Dictionary 2022-B への移行</a> .....	171
<a href="#">今後の TargetLink バージョンでの変更予定</a> .....	222

### 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 2022-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.....	154
MATLAB コード.....	158
Adaptive AUTOSAR.....	159
Classic AUTOSAR.....	161
コード生成のコア機能.....	164
MIL (Model-in-the-Loop) シミュレーション.....	165
ターゲットシミュレーション (PIL).....	166
ユーザビリティ.....	166
コードジェネレータオプション.....	167
API 関数とフックスクリプト.....	168
その他.....	169

## Simulink または Stateflow でのモデリング

### 次のステップ

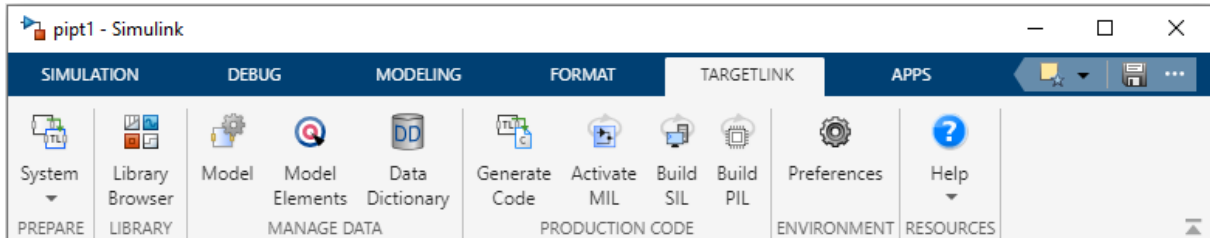
### 本章の内容

Simulink ツールストリップの[TargetLink]タブ.....	155
バス処理の改善.....	156
構造体配列のモデリングの改善.....	157
Stateflow での時相論理のサポートの改善.....	158
Interpolation Using Prelookup ブロック.....	158

## Simulink ツールストリップの[TargetLink]タブ

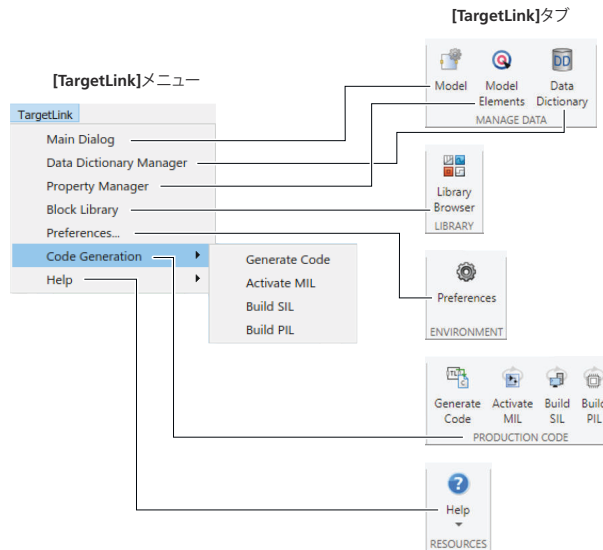
### Simulink ツールストリップの新しい[TargetLink]タブ

TargetLink 2022-B では、Simulink®ツールストリップに新しい[TargetLink]タブが導入されました。MATLAB®バージョン R2022a 以降では、[TargetLink]メニューは[TargetLink]タブに置き換わりました。



### 使用可能なコマンドの概要

MATLAB R2021b 以前では、以下のコマンドは[TargetLink]メニューにあります。MATLAB R2022a 以降では、これらのコマンドは Simulink ツールストリップの新しい[TargetLink]タブにあります。



[TargetLink]メニュー (MATLAB R2021b 以前)	[TargetLink]タブ (MATLAB R2022a 以降)
ブロックライブラリ	LIBRARY - Library Browser
Code Generation - Activate MIL	PRODUCTION CODE - Activate MIL
Code Generation - Build PIL	PRODUCTION CODE - Build PIL
Code Generation - Build SIL	PRODUCTION CODE - Build SIL
Code Generation - Generate Code	PRODUCTION CODE - Generate Code
Data Dictionary Manager	MANAGE DATA - Data Dictionary
Help - About TargetLink	RESOURCES - Help - About TargetLink
Help - Demos	RESOURCES - Help - Demos

[TargetLink]メニュー (MATLAB R2021b 以前)	[TargetLink]タブ (MATLAB R2022a 以降)
Help - dSPACE Help	RESOURCES - Help - Help
Help - New Features and Migration	RESOURCES - Help - New Features and Migration
Help - Useful links	RESOURCES - Help - Online
Help - Useful links - dSPACE Website	RESOURCES - Help - dSPACE Website
Help - Useful links - TargetLink Support Center	RESOURCES - Help - Visit the TargetLink Support Center
Help - Using dSPACE Help	RESOURCES - Help - Using dSPACE Help
Main Dialog	MANAGE DATA - Model
Make System/Library TargetLink-Compliant	PREPARE - System - Make System TargetLink-Compliant
Preferences...	Environment - Preferences
Property Manager	MANAGE DATA - Model Elements
-	PREPARE - System - Prepare System
-	PREPARE - System - Synchronize Data Types

## バス処理の改善

### Simulink.Bus オブジェクトと DD Typedef オブジェクト間のマッピング

TargetLink 2022-B では、バスオブジェクトと DD Typedef オブジェクトを DD SimulinkBusObject プロパティでマッピングできるようになりました。DD Typedef オブジェクトは、以下のコンテキストではマッピングされた構造体タイプと呼ばれます。

- モデルの準備中
- フレームモデルの生成中
- コードの生成中

インターフェースで指定されている場合、TargetLink は、これらのマッピングされた構造体タイプを、[nherit properties]チェックボックスをオンにしているバス対応ブロックの信号線で自動的に使用することができます。これは、マッピングされた構造体タイプを Bus Creator ブロックで使用する場合に特に便利です。「[構造体配列のモデリングの改善](#)」(157 ページ)を参照してください。

#### 関連ドキュメント

- 「[インターフェースにバスオブジェクトを設定する](#)」(『TargetLink 準備およびシミュレーションガイド』📖)
- 「[バスオブジェクトと DD Typedef オブジェクトのマッピング](#)」(『TargetLink 準備およびシミュレーションガイド』📖)
- 「[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』📖)
- 「[バス信号またはバス配列信号のマッピングされた構造体タイプを作成する方法](#)」(『TargetLink 準備およびシミュレーションガイド』📖)
- TargetLink デモモデル BUS\_CC

## 関連トピック

## 基本概要

BUS\_CC (『TargetLink デモモデル』[📄](#))  
 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』[📄](#))

## 構造体配列のモデリングの改善

### Bus-with-array-of-bus-as-payload 信号を作成する Bus Creator ブロックのサポート

TargetLink 2022-B では、Bus Creator ブロックを使用して、plain-array-of-bus や bus-with-array-of-bus-as-payload などの信号を bus-with-array-of-bus-as-payload 信号に接続できるようになりました。

#### 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』[📄](#))
- 「バス配列にアクセスするためのモデリングパターンの概要」(『TargetLink 準備およびシミュレーションガイド』[📄](#))
- 「バスポートブロックを使用した Bus-with-Array-of-Bus-as-Payload および Plain-Array-of-Bus 信号への動的アクセスのモデリングの例」(『TargetLink 準備およびシミュレーションガイド』[📄](#))

### DD Typedef オブジェクトを使用した構造体配列変数の実装

TargetLink 2022-B では、TargetLink ブロックで構造体配列変数を直接参照する以外に、DD Typedef オブジェクトを使用して構造体配列タイプを参照することで、構造体配列変数を指定できるようになりました。

#### 関連ドキュメント

- 「バス信号またはバス配列信号のマッピングされた構造体タイプを作成する方法」(『TargetLink 準備およびシミュレーションガイド』[📄](#))

### 参照パラメータとしての構造体配列変数

TargetLink 2022-B では、構造体配列変数を関数の参照パラメータとして使用することができます。

#### 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』[📄](#))

### インクリメンタルコード生成ユニットのインターフェースでの構造体配列変数

TargetLink 2022-B では、インクリメンタルコード生成ユニットのインターフェースで構造体配列変数をサポートしています。

#### 構造体配列変数またはバス配列信号に関するその他の新機能

- 「構造体配列のサポートの改善(Classic AUTOSAR)」(163 ページ)

## Stateflow での時相論理のサポートの改善

### temporalCount 演算子のサポート

TargetLink では、時相論理演算子 temporalCount をサポートするようになりました。ただし、以下のキーワードはサポートされていません。

- sec
- msec
- usec

#### 関連ドキュメント

- 「Stateflow(制限事項)」(『TargetLink 制限事項リファレンス』[📖](#))

## Interpolation Using Prelookup ブロック

### 5 次元までのテーブルデータのサポート

TargetLink 2022-B では、Interpolation Using Prelookup ブロックは 5 次元までのテーブルデータをサポートします。

#### 関連ドキュメント

- 「サポートされるルックアップテーブルの次元」(『TargetLink 準備およびシミュレーションガイド』[📖](#))
- 「ルックアップテーブルの値を指定する方法」(『TargetLink 準備およびシミュレーションガイド』[📖](#))
- 「3 次元テーブルの Prelookup ブロックと Interpolation Using Prelookup ブロックによるモデリングの例」(『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

Interpolation Using Prelookup ブロック (『TargetLink モデルエレメントリファレンス』[📖](#))

## MATLAB コード

### MATLAB コード関数と演算の拡張

#### サポートされる MATLAB コード関数と演算

TargetLink 2022-B では、以下の MATLAB®コード関数と演算のサポートが追加されました。

- bitxor
- norm

## 関連ドキュメント

- 「サポートされる MATLAB コード関数ステートメントと関数演算子」(『TargetLink コード生成リファレンス (Simulink®モデルの MATLAB®コード用)』)

## Adaptive AUTOSAR

## 次のステップ

## 本章の内容

Adaptive AUTOSAR での INOUT オペレーション引数のサポート.....	159
Adaptive AUTOSAR Data Dictionary インポートおよびエクスポート に関する変更.....	159
構造体配列のサポート (Adaptive AUTOSAR).....	160
Adaptive AUTOSAR データタイプ.....	160

## Adaptive AUTOSAR での INOUT オペレーション引数のサポート

### ARGINOUT オペレーション引数のサポート

TargetLink 2022-B では、ARGIN および ARGINOUT の種類のオペレーション引数の他に、メソッドで ARGINOUT の種類のオペレーション引数もサポートするようになりました。

## 関連ドキュメント

- 「メソッド呼び出しのモデル化方法」(『TargetLink Adaptive AUTOSAR モデリングガイド』)
- 「メソッド実装をモデル化する方法」(『TargetLink Adaptive AUTOSAR モデリングガイド』)
- 「Adaptive AUTOSAR の通信対象を作成する方法」(『TargetLink Adaptive AUTOSAR モデリングガイド』)

## Adaptive AUTOSAR Data Dictionary インポートおよびエクスポートに関する変更

### インポートおよびエクスポートでの DD ArTrigger オブジェクトのサポート

TargetLink 2022-B では、DD AUTOSAR インポートおよびエクスポートで DD ArTrigger オブジェクトをサポートしています。

**関連ドキュメント**

- 「Adaptive AUTOSAR の通信対象を作成する方法」(『TargetLink Adaptive AUTOSAR モデリングガイド』📖)

**InternalBehavior の AdminData のインポート/エクスポート**

TargetLink 2022-B では、ARXML のインポート時に、Data Dictionary で表現されない InternalBehavior オブジェクトの AdminData を格納するためのデータコンテナとして、新しい DD AdminDataOfInternalBehavior オブジェクトを提供しています。格納された AdminData は、再度エクスポートすることができます。

## 構造体配列のサポート (Adaptive AUTOSAR)

**Adaptive AUTOSAR での構造体配列のサポート**

TargetLink 2022-B では、以下の Adaptive AUTOSAR 通信サブジェクトでの構造体配列の使用がサポートされます。

- フィールド
- イベント
- メソッド

## Adaptive AUTOSAR データタイプ

**ara::core::Array および std::array データタイプのネイティブサポート**

TargetLink 2022-B は、ara::core::Array および std::array データタイプのコード生成をサポートしています。これらのデータタイプは、Classic AUTOSAR 配列データタイプと同じように使用することができます。

TargetLink 2022-B 以前では、ara::core::Array データタイプは ARA アダプタコードでのみ使用されていました。

**関連ドキュメント**

- 「Adaptive AUTOSAR のデータタイプの概要」(『TargetLink Adaptive AUTOSAR モデリングガイド』📖)

**C++名前空間のサポート**

TargetLink 2022-B は、C++名前空間で定義されたデータタイプのコードの生成をサポートしています。

TargetLink 2022-B 以前では、C++名前空間は ARA アダプタコードでのみ使用されていました。

**関連ドキュメント**

- 「Adaptive AUTOSAR のデータタイプの概要」(『TargetLink Adaptive AUTOSAR モデリングガイド』📖)



# 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.....	161
Classic AUTOSAR での INOUT オペレーション引数のサポート.....	162
Classic AUTOSAR Data Dictionary インポートおよびエクスポートに関する変更.....	162
AUTOSAR バージョン R21-11 に準拠した変更.....	162
構造体配列のサポートの改善 (Classic AUTOSAR).....	163

## サポートされる Classic AUTOSAR リリース

### サポートされる Classic AUTOSAR リリース

次の Classic AUTOSAR リリースがサポートされます。

Classic AUTOSAR リリース	リビジョン
R21-11	21-11 <sup>1)</sup>
R20-11	20-11
R19-11	19-11
4.4	4.4.0
4.3	4.3.1 4.3.0
4.2	4.2.2 4.2.1
4.1	4.1.3 4.1.2 4.1.1
4.0	4.0.3 4.0.2

<sup>1)</sup> TargetLink 2022-B で新たにサポート

## Classic AUTOSAR での INOUT オペレーション引数のサポート

### ARGINOUT オペレーション引数のサポート

TargetLink 2022-B では、ARGIN および ARGOUT の種類のオペレーション引数の他に、通信の種類 Operation 用の ARGINOUT の種類のオペレーション引数もサポートするようになりました。

#### 関連ドキュメント

- 「クライアントサーバ通信の基礎」(『TargetLink Classic AUTOSAR モデリングガイド』[📖](#))
- 「同期オペレーション呼び出しとサーバオペレーションのインプリメンテーションの組み合わせの基礎」(『TargetLink Classic AUTOSAR モデリングガイド』[📖](#))
- 「Classic AUTOSAR 通信のために通信対象を作成する方法」(『TargetLink Classic AUTOSAR モデリングガイド』[📖](#))

## Classic AUTOSAR Data Dictionary インポートおよびエクスポートに関する変更

### InternalBehavior の AdminData のインポート/エクスポート

TargetLink 2022-B では、ARXML のインポート時に、Data Dictionary で表現されない InternalBehavior オブジェクトの AdminData を格納するためのデータコンテナとして、新しい DD AdminDataOfInternalBehavior オブジェクトを提供しています。格納された AdminData は、再度エクスポートすることができます。

## AUTOSAR バージョン R21-11 に準拠した変更

### コード生成時の SWC 名の競合の回避

1 つの ECU で同じ名前の SWC を使用した場合に発生する名前の競合を避けるために、TargetLink 2022-B では DD SymbolicName プロパティを設定することができます。このプロパティを設定すると、TargetLink は生成したコード内の SWC の名前に、DD SoftwareComponent オブジェクトの名前ではなく、このプロパティの値を使用します。

さらに、TargetLink では新しい名前マクロ\$(SwComponentSymbolicName)を提供しています。AUTOSAR バージョンが R21-11 以上の場合、これはコード生成時に DD SymbolicName プロパティの値に置き換えられます。このプロパティを設定していない場合、または AUTOSAR バージョンが R20-11 以下の場合、名前マクロは DD SoftwareComponent オブジェクトの名前に置き換えられます。

#### 関連ドキュメント

- 「ソフトウェアコンポーネントを作成する方法」(『TargetLink Classic AUTOSAR モデリングガイド』[📖](#))
- 「名前マクロの展開の詳細」(『TargetLink カスタマイズおよび最適化ガイド』[📖](#))

ユーザ指定のデータタイプに対応したカスタム定義のヘッダーファイルの提供

AUTOSAR に準拠して、TargetLink 2022-B および AUTOSAR バージョン R21-11 では、ユーザ指定のデータタイプの定義を個別のヘッダーファイルで提供することができます。これは DD Typedef オブジェクトの DD TypeEmitter プロパティで指定されます。

#### 関連ドキュメント

- 「ユーザ指定の Classic AUTOSAR データタイプにカスタム定義されたヘッダーファイルを提供する方法」(『TargetLink Classic AUTOSAR モデリングガイド』[🔗](#))

## 構造体配列のサポートの改善 (Classic AUTOSAR)

フレームモデルの生成およびアップデート

TargetLink 2022-B では、構造体配列を含む Classic AUTOSAR データからフレームモデルを生成およびアップデートすることができます。TargetLink がポートまたはデータストアブロックを生成します。以下の通信の種類がサポートされます。

- InterRunnable
- Sender-Receiver
- NvData
- Operation

複雑なバス配列信号のモデリングを高速化するために、TargetLink ではアダプタブロックを生成することができます。

#### 関連ドキュメント

- 「Classic AUTOSAR におけるバス配列信号を使用したフレームモデルの生成と更新の詳細」(『TargetLink Classic AUTOSAR モデリングガイド』[🔗](#))

ポートブロックでより多くの通信の種類をサポート

TargetLink 2022-B では、ポートブロックベースのモデリングスタイルを使用して、構造体配列を含む Classic AUTOSAR データ用の RTE API 関数を生成することができます。TargetLink 5.2 以降でサポートされている通信の種類 Operation に加えて、以下の通信の種類もサポートされるようになりました。

- InterRunnable
- Sender-Receiver
- NvData

#### 関連ドキュメント

- 「TargetLink で Classic AUTOSAR をモデリングする場合の構造体の配列の使用の基礎」(『TargetLink Classic AUTOSAR モデリングガイド』[🔗](#))

データストアブロックでより多くの通信の種類をサポート

TargetLink 2022-B では、構造体配列を含む Classic AUTOSAR データと組み合わせ、データストアブロックを介して以下をモデリングすることができます。

- 明示的な通信
- 暗黙的な通信: Rte\_IWrite および Rte\_IrvIWrite のサポートを追加

**関連ドキュメント**

- 「TargetLink で Classic AUTOSAR をモデリングする場合の構造体の配列の使用の基礎」(『TargetLink Classic AUTOSAR モデリングガイド』📖)

## コード生成のコア機能

**次のステップ****本章の内容**

構造体配列のコード効率性の改善.....	164
モジュールのコード分類.....	165

## 構造体配列のコード効率性の改善

TargetLink 2022-B では、構造体配列変数に対して生成されるコードの効率性が向上しました。構造体全般が、構造体でない変数と同じように最適化されるようになりました。

これは特にバス信号から生成される構造体に関係するもので、この場合、構造体の代入が他の最適化より優先されます。

構造体の代入を優先すると、結果としてコードの効率性が低下する可能性があります。TargetLink では、これを緩和するために、以下のコードジェネレータオプションを提供しています。

- StructAssignmentDecompositionAuxVarMemoryUsageThreshold
- StructAssignmentDecompositionStatementMultiplicationThreshold

**その他の変更** 上記の改善とあわせて、以下の最適化も行われました。

- 状態更新ステートメントの再スケジューリング
- 構造体のスコープの縮小

**移行の問題** 特定のモデリング状況や使用事例によっては、一時的な回避策をとることで、以前の TargetLink バージョンでの量産コードの取得が可能になる場合があります。TargetLink については dSPACE 製品サポートにお問い合わせください。

**関連ドキュメント**

- 「AllowStructAssignments」(『TargetLink モデルエレメントリファレンス』📖)
- 「ExtendedLifeTimeOptimization」(『TargetLink モデルエレメントリファレンス』📖)
- 「StructAssignmentDecompositionAuxVarMemoryUsageThreshold」(『TargetLink モデルエレメントリファレンス』📖)
- 「StructAssignmentDecompositionStatementMultiplicationThreshold」(『TargetLink モデルエレメントリファレンス』📖)

## モジュールのコード分類

モジュールをシミュレーションコードとしてマーク

TargetLink 2022-B では、CodeClassification プロパティにより、DD Module オブジェクトをシミュレーションコードとしてマークすることができます。このプロパティが SimulationCode に設定されている場合、結果のコードファイルは、シミュレーションコードフォルダ (TLSim など) に配置されます。

### 関連ドキュメント

- ModuleInfo の説明  
( TargetLink Data Dictionary リファレンスを参照)

## MIL (Model-in-the-Loop) シミュレーション

### オーバーフロー検出を無効にしてパフォーマンスを向上

オーバーフロー検出を無効にして MIL シミュレーションモードのパフォーマンスを向上

MIL シミュレーションモードのパフォーマンスを向上させるために、TargetLink 2022-B ではオーバーフロー検出を無効にできるようになりました。つまり、シミュレーションの開始時に、非常に時間のかかる TargetLink ブロックデータの読み込みは、信号のロギングを有効にした TargetLink ブロックに対して行うだけで済みます。

特に、大規模なモデルで少数の TargetLink ブロックのみをログ記録したい場合は、オーバーフロー検出を無効にするとパフォーマンスが向上する可能性があります。モデルのオーバーフロー検出を無効にするには、`t1_set(<model>, 'logopt.detectoverflows', 'off')` コマンドを使用します。

### 関連ドキュメント

- 「オーバーフロー検出の基礎」(『TargetLink 準備およびシミュレーションガイド』[📖](#))
- 「移行に関するその他の注意点」(190 ページ)

## ターゲットシミュレーション (PIL)

### 2022-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 2022-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	新規	廃止
Freescale MPC5700VLE	GreenHill	2022	2021
Renesas RH850	GreenHill	2022	2021

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

Data Dictionary と Property Manager

TargetLink 2022-B では、Data Dictionary Manager と Property Manager のいくつかのユーザビリティが改良されています。

たとえば、Property Manager では、Column Chooser を使用した列の選択が最適化されています。

#### 関連ドキュメント

- 「[Show Column Chooser \(Property View\)](#)」(『TargetLink ツールおよびユーティリティリファレンス』[📄](#))
- 「[Column Chooser Dialog \(Property View\)](#)」(『TargetLink ツールおよびユーティリティリファレンス』[📄](#))

# 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 2022-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。

- 「ReduceLifetimeOfConstantVariables」(『TargetLink モデルエレメントリファレンス』[📖](#))  
初期値を保持して自動記憶域期間に修正されない構造体変数を減らします。
- StructAssignmentDecompositionAuxVarMemoryUsageThreshold  
構造体変数が削除されて必要なコンポーネントが補助変数に置き換えられた場合に、元の構造体変数のメモリ使用量のうち最大何パーセントを残すかを示す値を指定します。
- StructAssignmentDecompositionStatementMultiplicationThreshold  
実際に使用されている構造体のコンポーネントが少ない場合に、TargetLink が構造体の代入ではなく、コンポーネントの代入ステートメントを生成することを示す数値を定義します。
- 「UseBusObjectMappingInsteadOfInheritedProperties」(『TargetLink モデルエレメントリファレンス』[📖](#))  
先行ブロックのプロパティを継承せず、マッピングされた構造体タイプを使用して変数を作成します。

### 関連ドキュメント

- 「構造体配列のコード効率性の改善」(164 ページ)
- 「StructAssignmentDecompositionAuxVarMemoryUsageThreshold」(『TargetLink モデルエレメントリファレンス』[📖](#))
- 「StructAssignmentDecompositionStatementMultiplicationThreshold」(『TargetLink モデルエレメントリファレンス』[📖](#))
- 「ReduceLifetimeOfConstantVariables」(『TargetLink モデルエレメントリファレンス』[📖](#))
- 「UseBusObjectMappingInsteadOfInheritedProperties」(『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」(180 ページ)を参照してください。

## 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.....	168
新しいフックスクリプト.....	168

## 新しい API 関数

API 関数	用途
tlExportFixedPointLibrary	TargetLink の固定小数点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、SystemDesk で V-ECU コンテナや FMU を作成するためなどに使用します。

### 関連トピック

### 参考文献

tlExportFixedPointLibrary (『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_arrayofbusesadapterblocks_hook	バス配列アダプタブロックがモデルに追加された後に実行されるカスタムコマンドを入力します。これは、バス配列信号をルーティングする BusInport/BusOutport または DataStoreRead/DataStoreWrite ブロックごとに呼び出されます。 「Classic AUTOSAR におけるバス配列信号を使用したフレームモデルの生成と更新の詳細」(『TargetLink Classic AUTOSAR モデリングガイド』 <a href="#">📖</a> )を参照してください。



フックスクリプト	説明
tl_pre_add_arrayofbusesadapterblocks_hook	<p>バス配列アダプタブロックがモデルに追加される前に実行するカスタムコマンドを入力します。これはバス配列信号をルーティングする BusInport/BusOutport または DataStoreRead/DataStoreWrite ブロックごとに呼び出されます。</p> <p>「Classic AUTOSAR におけるバス配列信号を使用したフレームモデルの生成と更新の詳細」(『TargetLink Classic AUTOSAR モデリングガイド』<a href="#">📖</a>)を参照してください。</p>

## 関連トピック

### 基本概要

[Classic AUTOSAR におけるバス配列信号を使用したフレームモデルの生成と更新の詳細 \(『TargetLink Classic AUTOSAR モデリングガイド』\[📖\]\(#\)\)](#)

### 参考文献

[tl\\_post\\_add\\_arrayofbusesadapterblocks\\_hook \(『TargetLink ファイルリファレンス』\[📖\]\(#\)\)](#)

[tl\\_pre\\_add\\_arrayofbusesadapterblocks\\_hook \(『TargetLink ファイルリファレンス』\[📖\]\(#\)\)](#)

## その他

## TargetLink のデモと例

### 変更されたデモモデル

変更されたデモモデル	説明
AAR_COMMUNICATION	AAR_COMMUNICATION デモモデルは、Adaptive AUTOSAR で説明されているように、配列実装データタイプを表示するように調整されました。
BUS_CC	BUS_CC デモモデルは、バスオブジェクトとマッピングされた構造体タイプを使用するメリットを示すように調整されました。
BUS_STRUCT	BUS_STRUCT デモモデルは、以下のメリットを示すように調整されました。 <ul style="list-style-type: none"> <li>マッピングされた構造体タイプの使用。</li> <li>バス配列信号からの参照パラメータの生成。</li> <li>BusCreator ブロックを介したバス配列信号の組み立て。</li> </ul>
STRUCT_INTERFACES	STRUCT_INTERFACES デモモデルは、pointer-to-struct-transport 変数を使用してポインタ引数を関数呼び出し階層まで伝播するメリットを示すように調整されました。
AUTOSAR_FUELSYS	AUTOSAR_FUELSYS デモモデルは、コンテナ交換の廃止を反映するように調整されました。
AR_MEMORY_MAPPING	AR_MEMORY_MAPPING デモモデルは、コンテナ交換の廃止を反映するように調整されました。

#### 削除されたデモモデル

バス配列信号を使用して RTE API 関数のパラメータを生成できるようになったため、AR\_ARRAY\_OF\_STRUCT\_DATA デモモデルは製品から削除されました。

# TargetLink 2022-B および TargetLink Data Dictionary 2022-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.....	171
TargetLink 5.2 から 2022-B への移行.....	179
TargetLink 5.2 と TargetLink 2022-B 間でのコードの変更.....	196
TargetLink 2022-B での廃止項目.....	218

## 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.....	171
正確に表現可能な浮動小数点数の丸め動作の変更 (Microsoft).....	179

## 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.....	172
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.....	174
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.....	175
Data Dictionary の CodeDecorationSets への移行.....	176

## 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 2022-B では、TargetLink 3.1 以降で作成されたモデルと 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 バージョンで必要なすべてのステップを含んでいます。たとえば、TargetLink 4.0 から TargetLink 2022-B への自動アップグレードは、4.0 から 4.1、4.2、4.3、4.4、5.0、5.1、5.2、2022-B へのアップグレードのステップで構成されます。

#### 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。

### 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](#)」(174 ページ)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 2022-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ため、ライブラリは TargetLink 2022-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 2022-B のみ使用可能** ライブラリを TargetLink 準拠にするには、TargetLink 2022-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」(175 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 2022-B で保存されます。そのため、TargetLink 2022-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エレメント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 コマンドウィンドウの 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の確認 <sup>1)</sup>
<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 は、TargetLink Message Browser にエラーに関する情報を表示します。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』 )

### 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..... 175  
[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。

<b>方法</b>	<p><b>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</b></p> <ol style="list-style-type: none"> <li>[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。</li> <li>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/&lt;DDIncludeFile&gt; オブジェクトの AutoSave プロパティを On に設定します。</li> <li>[File] - [Save]を使用してメイン DD ファイルを保存します。</li> </ol>
-----------	---

<b>結果</b>	<p>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。</p>
-----------	---

<b>関連トピック</b>	<p><b>基本概要</b></p> <p><a href="#">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』)</a></p> <p><b>操作手順</b></p> <p><a href="#">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』)</a></p> <p><b>参考文献</b></p> <p><a href="#">Point of Inclusion (『TargetLink Data Dictionary Manager リファレンス』)</a></p>
---------------	--

## 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

<b>目的</b>	<p>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。</p>
<b>前提条件</b>	<p>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。</p> <p>必要な DD プロジェクトファイルは、<code>dsdd_manage_project('Open', '&lt;name&gt;.dd')</code> や <code>dsdd('Open', '&lt;name.dd&gt;')</code> 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。</p>

## 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す。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す。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、常に切り上げられます。	<a href="#">IEEE 754-2019</a> に準拠した最近接丸め。

## TargetLink 5.2 から 2022-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.....	180
AUTOSAR.....	182
API 関数とフックスクリプト.....	185
ルックアップテーブルブロックに関する移行上の注意点.....	188

最適化.....	189
移行に関するその他の注意点.....	190

## 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

#### ユーザインターフェース

OptimizedBoolType コードジェネレータオプションは、TargetLink Main Dialog ブロックの[Advanced]ページから削除されました。ただし、このページの[All options...]ボタンを使用して、このオプションを引き続き使用することができます。

#### 関連ドキュメント

- 「OptimizedBoolType」(『TargetLink モデルエレメントリファレンス』[📖](#))

#### 推奨される互換性設定

最適な下位互換性を維持するために、TargetLink 2022-B の新しいコードジェネレータオプションで次の設定を行ってください。

コードジェネレータオプション	設定
「ReduceLifetimeOfConstantVariables」(『TargetLink モデルエレメントリファレンス』 <a href="#">📖</a> )	off
「UseBusObjectMappingInsteadOfInheritedProperties」(『TargetLink モデルエレメントリファレンス』 <a href="#">📖</a> )	off

上記のコードジェネレータオプションの変更による影響は、多くの場合、バス構造体に関する他の変更と比べるとわずかです。「最適化に関する移行上の注意点」(189ページ)を参照してください。

#### 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)。また、DD CodegenOptionSet オブジェクトにユーザ定義の一連の Code Generator オプションを保存することができます(DD ベースのオプションの保存)。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、また、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

<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。

<sup>2)</sup> 手作業によるリセットが必要な場合があります。

<sup>3)</sup> 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](#)」(167 ページ)を参照してください。

## AUTOSAR

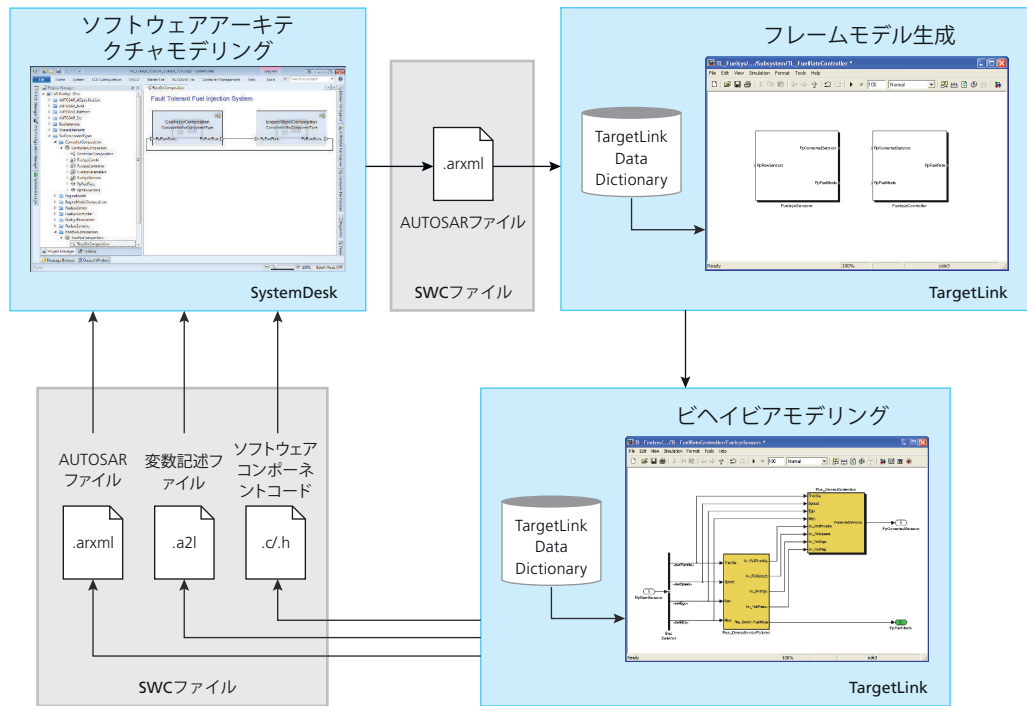
### AUTOSAR に関する移行上の注意点

ソフトウェアコンポーネントを交換するためのベストプラクティス

AUTOSAR に準拠した ECU ソフトウェア開発で、SystemDesk はシステムレベル設計ツールとして機能します。ドメインモデリング用に、dSPACE は TargetLink AUTOSAR Module を提供しています。この使用事例に従う場合、AUTOSAR モデルの関連部分を SystemDesk から TargetLink に転送する必要があります。

dSPACE は、V-ECU および環境モデルの閉ループシミュレーションのコンテキストで、ソフトウェアコンポーネントの SIL テストをサポートします。この使用事例に従う場合、さらに AUTOSAR モデルのコードファイルを参照し、ここで実装固有の側面を指定する必要があります。これにより、SystemDesk でモデルベースの V-ECU を生成することができます。

下の図は、SystemDesk と TargetLink の相互運用性を示しています。



TargetLink と SystemDesk の間でソフトウェアコンポーネントを交換するための dSPACE ツールである *Container Manager* の廃止以降は、ベストプラクティスに従うことをお勧めします。これには、パッケージおよび AUTOSAR ファイルへの AUTOSAR エレメントの割り当て、SystemDesk および TargetLink Data Dictionary での特定のインポート/エクスポート関連の設定、交換された AUTOSAR ファイルのバージョン管理が含まれます。また、AUTOSAR ファイルの比較およびマージツールの使用をお勧めします。

詳細については、お使いの dSPACE ツールのユーザマニュアルに記載されているベストプラクティスを参照してください。

dSPACE ツール	マニュアル
SystemDesk	「ソフトウェアコンポーネントをやり取りするためのベストプラクティス」(『SystemDesk マニュアル』📖)
TargetLink	「ソフトウェアコンポーネントをやり取りするためのベストプラクティス」(『TargetLink 相互運用およびデータ交換ガイド』📖)

### AUTOSAR 用ファイルのエクスポート

TargetLink 2022-B では、ARXML ファイルなどのソフトウェアコンポーネント関連のファイルを生成して、TargetLink File Export Utility でエクスポートすることができます。

#### 関連ドキュメント

- 「生成されたファイルを TargetLink 環境からエクスポートする方法」(『TargetLink 相互運用およびデータ交換ガイド』[📖](#))
- 「TargetLink File Export Utility」(『TargetLink ツールおよびユーティリティリファレンス』[📖](#))

#### AUTOSAR フレームモデル生成の変更

TargetLink 2022-B では、バス配列信号を渡さずに構造化された DD Typedef オブジェクトを参照する Bus Inport および Bus Outport ブロックに対して、Simulink.Bus オブジェクトが作成されます。この動作は、ForceBusObjectDatatypeUsage プロパティで無効にすることができます。

#### 関連ドキュメント

- ForceBusObjectDatatypeUsage  
( TargetLink Data Dictionary リファレンスを参照)
- 「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』[📖](#))

#### Adaptive AUTOSAR に関連するデータタイプのモデリングに関する変更

TargetLink 2022-B では、TargetLink Data Dictionary での Adaptive AUTOSAR に関連するデータタイプのモデリングが変更されました。Data Dictionary のアップグレード中に、既存の DD workspaces は変更されたモデリングに自動的にアップグレードされます。アップグレード処理後、以前 Adaptive AUTOSAR のタイプ定義に使用していた DD Module オブジェクトが未使用として残される場合があります。

#### 関連ドキュメント

- 「Adaptive AUTOSAR のデータタイプの概要」(『TargetLink Adaptive AUTOSAR モデリングガイド』[📖](#))
- 「Adaptive AUTOSAR におけるユーザ指定データタイプの定義に関する詳細」(『TargetLink Adaptive AUTOSAR モデリングガイド』[📖](#))

#### \$(Component)マクロの置換

TargetLink 2022-B では、以下の場合、\$(Component)マクロは\$(SwComponentSymbolicName)に置き換えられます。

- DD アップグレード  
AccessFunction.FunctionName、  
Variable.NameTemplate、Runnable.NameTemplate、  
CodeDecoration.Settings.DeclarationStatements の各プロパティに対して、マクロが置き換えられます。
- DD コマンド CreateActivationReasonCodeElements  
このコマンドは、作成した DD Variable オブジェクトの NameTemplate および ModulRef プロパティに対するマクロを置き換えます。
- DD コマンド CreateVariableObjects  
このコマンドは、NameTemplate プロパティに Rte\_\$(Component)\_\$(Group)\_\$D ではなく Rte\_\$(SwComponentSymbolicName)\_\$(Group)\_\$D を指定した DD Variable オブジェクトを作成します。



- 影響を受ける変数、マクロ、関数に由来する名前の変数、マクロ、関数に置き換わります。特に、ベクトルやマクロのローカルの access-function-buffer 変数が影響を受けます。

#### 関連ドキュメント

- CreateActivationReasonCodeElements (ランナブル)  
(TargetLink Data Dictionary リファレンスを参照)
- CreateVariableObjects (NvReceiverPort、NvSenderReceiverPort など)  
(TargetLink Data Dictionary リファレンスを参照)

## API 関数とフックスクリプト

### 次のステップ

### 本章の内容

TargetLink と TargetLink Data Dictionary API 関数の変更.....	185
フックスクリプトの変更.....	188

## TargetLink と TargetLink Data Dictionary API 関数の変更

### ds\_error\_display

#### プロパティ動作の変更

プロパティ	説明
ClearMessage	TargetLink のメッセージシステムから表示済みメッセージをクリアします。 このプロパティのデフォルトは、off になりました。

関連ドキュメント [「ds\\_error\\_display」](#) (『TargetLink API リファレンス』📖)

### dsdd\_validate

#### 新しい出力パラメータ

出力パラメータ	説明
bError	エラー時に true になるエラーフラグ。エラーでない場合は false。

関連ドキュメント [「dsdd\\_validate」](#) (『TargetLink API リファレンス』📖)

### tl\_compile\_host

削除されたプロパティ RebuildAll プロパティは削除されました。

関連ドキュメント [「tl\\_compile\\_host」](#) (『TargetLink API リファレンス』📖)

## tl\_compile\_target

削除されたプロパティ RebuildAll プロパティは削除されました。

関連ドキュメント [「tl\\_compile\\_target」](#) (『TargetLink API リファレンス』📖)

## tl\_export\_files

新しいプロパティ

プロパティ	説明
CopyAutosarFiles	AUTOSAR 関連のファイルを指定されたフォルダにコピーします。

関連ドキュメント [「tl\\_export\\_files」](#) (『TargetLink API リファレンス』📖)

## tl\_generate\_swc\_model

新しいプロパティ

プロパティ	説明
AddArrayOfBusesAdapterBlocks	バス配列アダプタブロックを追加し、バス配列信号を渡す BusInportDataStoreRead ブロックと BusOutputs または DataStoreWrite ブロックにそれらを接続するかどうかを指定します。 デフォルト: off
BusObjectsNamePrefix	マッピングされていない構造化された DD Typedef オブジェクトおよび DD Typedef でないオブジェクトに対して作成される Simulink.Bus オブジェクトの名前プレフィックスを指定します。 デフォルト: tlSwcGen_
ForceBusObjectDatatypeUsage	バス配列信号を渡さない BusInport または BusOutput ブロックに対して、Simulink.Bus オブジェクトを作成するかどうかを指定します。 デフォルト: on
UpdateBusObjectWithMappedDDTypedef	Simulink.Bus オブジェクトを、マッピングされた DD Typedef オブジェクトを使用して更新するかどうかを指定します (相互に互換性がない場合)。 デフォルト: on

関連ドキュメント [「tl\\_generate\\_swc\\_model」](#) (『TargetLink API リファレンス』📖)

**tl\_pref**

**環境設定の名前変更** 環境設定 AutoBackupDdFile は、SaveBackupOnDDUpgrade に名前が変更されました。

**関連ドキュメント** 「tl\_pref」(『TargetLink API リファレンス』[📖](#))

**tl\_sim**

MIL シミュレーション中の信号ロギングに影響する、TargetLink ブロックの Simulink<sup>®</sup>出力ポートパラメータの自動修正を行います。「MIL シミュレーションモードでの信号のロギングの基礎」(『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。

**関連ドキュメント** 「tl\_sim」(『TargetLink API リファレンス』[📖](#))

**tlSimulinkBusObject****プロパティ動作の変更**

プロパティ	説明
DDVariableName	DD Variable オブジェクトの名前またはパスを指定します。
CreateTypedef	DD Typedef オブジェクトを Simulink.Bus オブジェクトから作成して DD Variable オブジェクトで参照するかどうかを指定します。 このプロパティのデフォルトは、on になりました。

**関連ドキュメント** 「tlSimulinkBusObject('CreateDDVariable', busObject, propertyName, propertyValue, ...)」(『TargetLink API リファレンス』[📖](#))


**関連トピック****基本概要**

MIL シミュレーションモードでの信号のロギングの基礎 (『TargetLink 準備およびシミュレーションガイド』[📖](#))

**参考文献**

ds\_error\_display (『TargetLink API リファレンス』[📖](#))  
 dsdd\_validate (『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』[📖](#))  
 tl\_export\_files (『TargetLink API リファレンス』[📖](#))  
 tl\_generate\_sw\_model (『TargetLink API リファレンス』[📖](#))  
 tl\_pref (『TargetLink API リファレンス』[📖](#))  
 tl\_sim (『TargetLink API リファレンス』[📖](#))  
 tlSimulinkBusObject('CreateDDVariable', busObject, propertyName, propertyValue, ...) (『TargetLink API リファレンス』[📖](#))

## フックスクリプトの変更

tl\_post\_load\_hook 定義済み変数 system は mdlName に名前が変更されました。  
 関連ドキュメント [「tl\\_post\\_load\\_hook」](#) (『TargetLink ファイルリファレンス』)

### 関連トピック

#### 参考文献

[tl\\_post\\_load\\_hook](#) (『TargetLink ファイルリファレンス』)

## ルックアップテーブルブロックに関する移行上の注意 点

### Prelookup ブロックと Interpolation Using Prelookup ブロック

#### ブロックプロパティの変更



TargetLink 2022-B では、Prelookup および Interpolation Using Prelookup ブロックのブロックプロパティが変更されました。

ブロックプロパティ	TargetLink ≤ 2022-B	TargetLink 2022-B
Prelookup ブロックの[Use last breakpoint for input at or above upper limit]チェックボックス	TargetLink では評価されません。	Simulink ブロックプロパティと同期され、コード生成に使用されます。
Interpolation Using Prelookup ブロックの[Valid index input may reach last index]チェックボックス		

#### 潜在的な問題

- メッセージ E21581 が表示されます。
- メッセージ E21583 が表示されます。
- 予期しないコード変更については、「[1次元と2次元の Prelookup ブロックの実装の違い](#)」(217 ページ)を参照してください。

**ソリューション** 以下の手順を実行します。

- 「[MISRA-C:2012 準拠](#)」(『TargetLink モデルエレメントリファレンス』)の説明に従って Prelookup ブロックを設定します。
- 「[MISRA-C:2012 準拠](#)」(『TargetLink モデルエレメントリファレンス』)の説明に従って、Interpolation Using Prelookup ブロックを設定します。

**関連するコード変更** 「[1次元と2次元の Prelookup ブロックの実装の違い](#)」(217 ページ)を参照してください。

**DD ルックアップテーブルオブジェクトの変更**

Prelookup ブロックと Interpolation Using Prelookup ブロックの DD ルックアップテーブルオブジェクトは、対応するブロックプロパティの変更を反映するように変更されました。

DD ルックアップテーブルオブジェクトのプロパティ	TargetLink ≤ 2022-B	TargetLink 2022-B
IndexSearch Block オブジェクトの[Use last breakpoint for input at or above upper limit]	–	移行中に自動的に off に設定されます。
DD Interpolation Block オブジェクトの [Valid index input may reach last index]プロパティ	–	

**潜在的な問題**

- 予期しないコード変更については、「[1次元と2次元の Prelookup ブロックの実装の違い](#)」(217 ページ)を参照してください。
- メッセージ E21583 が表示されます。

**ソリューション** 要件に合わせて DD ルックアップテーブルオブジェクトを調整します。

**関連トピック****参考文献**

[IndexSearch ブロックオブジェクトダイアログ \(Prelookup\)](#) (『TargetLink Data Dictionary Manager リファレンス』[📖](#))  
[Interpolation Block Object ダイアログ](#) (『TargetLink Data Dictionary Manager リファレンス』[📖](#))  
[Interpolation Using Prelookup ブロック](#) (『TargetLink モデルエレメントリファレンス』[📖](#))  
[Prelookup ブロック](#) (『TargetLink モデルエレメントリファレンス』[📖](#))

## 最適化

### 最適化に関する移行上の注意点

**構造体の代入**

構造体配列変数のコード効率性を改善すると、状態更新ステートメントの再スケジューリングや変数のスコープの縮小でコード効率性が低下する可能性があります。「[構造体配列のコード効率性の改善](#)」(164 ページ)を参照してください。

特定の状況や使用事例によっては、一時的な回避策をとることで、以前の TargetLink バージョンでの量産コードの取得が可能になる場合があります。TargetLink については dSPACE 製品サポートにお問い合わせください。

## 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

#### Simulink.Bus オブジェクトと DD Typedef オブジェクトのマッピング

TargetLink 2022-B では、DD Typedef オブジェクトに SimulinkBusObject プロパティが導入されました。このプロパティは、DD Typedef オブジェクトと構造的に同一の Simulink.Bus オブジェクトの対応関係を示すために使用されます。Data Dictionary に関連付けられた TargetLink モデルを開く前に、Data Dictionary のマッピングをアップデートすることを強くお勧めします。

#### 関連ドキュメント

- 「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』📖)
- SimulinkBusObject  
( TargetLink Data Dictionary リファレンスを参照)

#### 関数インターフェースでの pointer-to-struct 変数

TargetLink 2022-B では、新しい DD PassStructToComponentReferences プロパティを使用して、関数インターフェースで pointer-to-struct タイプの仮パラメータを明示的に制御するオプションが追加されました。

以前のバージョンでは、以下の条件に該当する場合、関数インターフェースで pointer-to-struct-transport パラメータが自動的に生成されていました。

- 関数インターフェースに寄与するモデルエレメントが、構造化された DD Variable オブジェクトまたはそのコンポーネントの 1 つを参照している。
- 構造化された DD Variable オブジェクトが、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。

これらの仮パラメータは、実パラメータと複数の関数間でのそれらの使用に関して、特別なセマンティクスを持ちます。反対に、バスポートブロックでは、これらの仮パラメータは、構造化された DD Typedef オブジェクトの参照と、Scope プロパティが ref\_param に設定された DD VariableClass オブジェクトへの参照を介して指定されたブロック変数とは動作が異なります。

DD ファイルを TargetLink 2022-B にアップグレードすると、DD PassStructToComponentReferences プロパティがすべての DD Variable オブジェクトに追加されます。

次のすべての条件に該当する場合、PassStructToComponentReferences プロパティは on に設定されます。

- DD Variable オブジェクトは構造化されているが、構造体コンポーネントではない。
- DD Variable オブジェクトは構造体配列変数ではなく、構造体配列コンポーネントを含まない。
- 参照される DD VariableClass オブジェクトの Scope プロパティが ref\_param に設定されている。

結果の変数は、pointer-to-struct-transport 変数で呼び出されます。モデルエレメントでそのコンポーネントを参照することができます。

デフォルトでは、新しい DD Variable オブジェクトに PassStructToComponentReferences プロパティは設定されていません。これは off が設定されているのと同じです。Scope プロパティに ref\_param が設定された DD VariableClass オブジェクトを参照する場合、それらはバス構造体、すなわち構造体変数で収集された個々の値の集合ではなく、構造体タイプの複素数値を表すものとして解釈されます。この構造体変数のコンポーネントをモデルエレメントで参照することはできません。ただし、バスポートブロックで参照する場合、結果のコードは、バスポートまたはポートブロックの他のタイプの参照パラメータと構造的には同じです。

#### 関連ドキュメント

- 「関数インターフェースの定義」(『TargetLink カスタマイズおよび最適化ガイド』  
📖)
- 「構造体へのポインタの使用」(『TargetLink 準備およびシミュレーションガイド』  
📖)
- 「STRUCT\_INTERFACES」(『TargetLink デモモデル』📖)

#### TargetLink ログイングプロパティと Simulink データログイングの直接マッピング

TargetLink 2022-B では、[Data to log] (logdata.loggingmode) プロパティが Simulink ポートまたは Stateflow データオブジェクトの DataLogging パラメータにマッピングされます。

そのため、TargetLink ブロックのログイングが有効になると同時に、対応する Simulink 出力ポートまたは Stateflow オブジェクトのデータログイングも有効になります。その結果、各ブロックの出力にビューアアイコンが表示されます。



これらのビューアアイコンをモデル内で非表示にしたい場合は、MATLAB コマンドウィンドウで以下のいずれかのコマンドを入力します。

- `set_param(<model>, 'ShowViewerIcons', 'off')`
- `set_param(<model>, 'ShowTestPointIcons', 'off')`

**モデル比較への影響** このプロパティマッピングの変更は、Model Compare でのモデル比較など、下流の解析に影響を与える可能性があります。

たとえば、TargetLink 5.2 以前では、グローバルログイングオプションを変更しても、TargetLink モデルの 1 つのプロパティ(logopt.globalloggingmode)が変更されるだけでした。

TargetLink 2022-B では、グローバルログイングオプションを変更すると、非常に多くの TargetLink ブロックポートの Simulink パラメータも追加で変更されます。このため、たとえばグローバルログイングオプションが[Log according to block data]または[Log signal histories]に設定されている場合、Model Compare でそうでなければ同一のモデル間で相違が多数発生します。

#### 関連ドキュメント

- 「MIL シミュレーションモードでの信号のロギングの基礎」(『TargetLink 準備およびシミュレーションガイド』[📖](#))
- 「アドレスによるグローバル変数のロギングの基礎」(『TargetLink 準備およびシミュレーションガイド』[📖](#))

#### 新しいログマクロ

TargetLink 5.2 までは、TargetLink モデルで TargetLink ブロックの出力ポートに対して Simulink ロギングが有効になっている場合、この設定は無視されていました。

TargetLink 2022-B では、グローバルロギングオプションが [Log according to block data] に設定されている場合、これらの TargetLink ブロックの [Data to log] プロパティは [Signal history] に設定されます。その結果、コード生成中にログマクロが生成されます。

#### 関連ドキュメント

- 「生成されるコードにログに記録するマクロを挿入した場合の影響」(『TargetLink 準備およびシミュレーションガイド』[📖](#))

#### アドレスによるロギングの MIL/SIL 比較

TargetLink 5.2 までは、Data Dictionary でアドレスによるロギング (SIL/PIL モード) が指定されている変数では、それに対応するブロック信号も MIL モードで自動的にロギングされていました。

TargetLink 2022-B では、この機能はなくなりました。

アドレスによるロギングを使用するモデルで MIL/SIL 比較を行うには、以下の条件を満たす必要があります。

- TargetLink Main Dialog の [Code Generation] ページで、[Clean code] チェックボックスをオンにしている。  
したがって、コードインストールメンテーションは存在せず、そのためログマクロもありません。
- ロギングされる TargetLink ブロックが以下に該当する。
  - 信号ロギング (ローカルまたはグローバル) が有効である。  
[Clean code] オプションの設定により、これは MIL シミュレーションにのみ影響します。
  - コード生成後、Data Dictionary でロギングする変数にアドレスによるロギングを指定している。  
TargetLink の動作が変更されたため、これは SIL および PIL シミュレーションにのみ影響します。

#### 関連ドキュメント

- 「MIL シミュレーションモードでの信号のロギングの基礎」(『TargetLink 準備およびシミュレーションガイド』[📖](#))
- 「アドレスによるグローバル変数のロギングの基礎」(『TargetLink 準備およびシミュレーションガイド』[📖](#))

#### モデルパラメータ ConditionallyExecuteInputs の 設定でオーバーフロー検出なし

TargetLink 2022-B では、オーバーフロー検出と最小/最大ロギング (それぞれ信号ロギングなし) は、パラメータ ConditionallyExecuteInputs が設定されていない場合にのみ機能します。



このため、t1Upgrade では、ConditionallyExecuteInputs パラメータは off に設定され、モデルアップグレード中に対応するメッセージが表示されます。

ConditionallyExecuteInputs パラメータを無効にした結果として、これまで最適化から除外されていた計算が行われるようになり、MIL シミュレーション中は実行時間が長くなる場合があります。以前の TargetLink バージョンでは、影響を受ける TargetLink ブロックの条件付き分岐実行の最適化はローカルで防止されていたため、これはモデルの純粋な Simulink 部分にのみ影響します。

ConditionallyExecuteInputs モデルパラメータを on に設定している場合、オーバーフロー検出と最小/最大ロギングが機能しないことを示す警告が表示されます。この警告が表示されないようにするには、オーバーフロー検出を無効にし、グローバルロギングオプションを [Log min/max-values] に設定しないようにします。

**オーバーフロー検出か条件付き入力分岐の実行か - ユースケースに応じて設定**  
デフォルト設定は、オーバーフロー検出の使用と MIL シミュレーションモードでの作業に合わせて最適化されています。

コード生成に専念し、SIL または PIL シミュレーションモードで作業する場合は、オーバーフロー検出をオフにして、ConditionallyExecuteInputs モデルパラメータを on に設定することをお勧めします。こうすることで、MIL ロギングがなくなった場合に、パフォーマンスを向上させることができます。

#### 関連ドキュメント

- 「オーバーフロー検出 (制限事項)」(『TargetLink 制限事項リファレンス』[📖](#))
- 「オーバーフロー検出の基礎」(『TargetLink 準備およびシミュレーションガイド』[📖](#))

#### ローカルロギングがグローバルロギング設定に対応しない

TargetLink 2022-B では、グローバルロギングオプションを変更すると、Simulink ロギングパラメータが設定されます。

t1\_set コマンド、ブロックダイアログ、または Property Manager を使用してブロック固有の TargetLink ロギングプロパティを設定すると、TargetLink はローカルプロパティを定義済みのグローバルロギングオプションに対応させます。

ただし、以下の場合、この整合性は保証されません。

- 新しい TargetLink ブロックがモデルに追加されている。
- Simulink ブロックが拡張されている。
- Simulink ロギングパラメータが、Simulink 手段 (set\_param など) を使用して変更されている。

シミュレーション開始時のこのような不整合を避けるために、TargetLink はシミュレーションの開始時に、TargetLink Main Dialog またはプロットウインドウで t1\_sim コマンドまたは [Start simulation] ボタン (▶) を使用して、このような不適切な Simulink ロギングパラメータの修正を自動的に実行します。さらに、Merge ブロックのインポートにつながる信号のロギングも無効になります。

この自動修正は、t1\_set(<model>, 'logopt.applygloballogging', 'off') コマンドで無効にすることができます。

#### 関連ドキュメント

- 「ApplyGlobalLogging」(『TargetLink モデルエレメントリファレンス』[📖](#))
- 「MIL シミュレーションモードでの信号のロギングの基礎」(『TargetLink 準備およびシミュレーションガイド』[📖](#))

### MIL シミュレーションモードでのパフォーマンスの向上

MIL シミュレーションモードでのパフォーマンスを向上させるために、TargetLink 2022-B ではオーバーフロー検出を無効にすることができます。つまり、シミュレーションの開始時に、非常に時間のかかる TargetLink ブロックデータの読み込みは、信号のロギングを有効にした TargetLink ブロックに対してのみ行う必要があります。

特に、大規模なモデルで少数の TargetLink ブロックのみをログ記録したい場合は、オーバーフロー検出を無効にすると、パフォーマンスが向上する可能性があります。モデルのオーバーフロー検出を無効にするには、`t1_set(<model>, 'logopt.detectoverflows', 'off')` コマンドを使用します。

無効にすると、グローバルロギングモード[Log according to block data]と[Do not log anything]に影響します。パフォーマンス上の理由から、グローバルロギングモード[Log according to block data]では、ブロック固有のロギングが[Min/Max values]に設定されている場合は無視されます。

#### 関連ドキュメント

- 「オーバーフロー検出の基礎」(『TargetLink 準備およびシミュレーションガイド』[📖](#))

### Merge ブロックの入力ポートロギングでの Simulink エラー

ロギングを有効にした Merge ブロックの入力ポートに信号を入力すると、次のような場合に Simulink エラーでシミュレーションが中断することがあります。

- Simulink の `sim` コマンドまたはモデルツールバーの[Run]ボタンをクリックしてシミュレーションを開始した。
- ApplyGlobalLogging モデルプロパティが 'off' に設定されている。  
この場合、TargetLink Main Dialog またはプロットウィンドウで `t1_sim` コマンドまたは[Start simulation]ボタン([▶](#))によりシミュレーションを開始した場合でも、エラーが表示されることがあります。

このエラーを回避するには、関連するブロックで信号ロギングを無効にします。

#### 関連ドキュメント

- 「Merge ブロックの基礎」(『TargetLink 準備およびシミュレーションガイド』[📖](#))
- 「MIL シミュレーションモードでの信号のロギングの基礎」(『TargetLink 準備およびシミュレーションガイド』[📖](#))
- 「ログマクロを使用するロギングの基礎」(『TargetLink 準備およびシミュレーションガイド』[📖](#))
- 「ApplyGlobalLogging」(『TargetLink モデルエレメントリファレンス』[📖](#))

### リンクされたライブラリブロックを含むモデルのアップグレード時のエラー

TargetLink 2022-B のアップグレード処理中に、ロックされた(読み取り専用の)ライブラリの変更に際して、MATLAB®でエラーが報告される場合があります。これは次のような誤ったモデリングに関連したエラーである可能性があります。

- TargetLink ブロックがライブラリに直接リンクされている。
- ライブラリ内の TargetLink ブロックが、同じライブラリ内のブロックにリンクされている。

これらのエラーを解決するには、「`t1Upgrade`」(『TargetLink API リファレンス』[📖](#)) API コマンドを使用してライブラリを手動でアップグレードします。

```
t1Upgrade('Model', '<MyLibrary>', 'CheckModel', 'FixIssues');
```

**関連ドキュメント**

- 「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」(175 ページ)

**カスタムルックアップ関数でのメッセージ処理**

TargetLink 2022-B では、カスタムルックアップ関数 API(tlscript)でメッセージスタック内の全メッセージが収集されます。これにより、エラーなしは検出されなくなります。

メッセージを抑制するには、以下のように ds\_error\_none API 関数を使用する必要があります。

```
tlscript('Set', ...)

if ds_error_check
  ds_error_none
end
```

**関連ドキュメント**

- 「カスタムルックアップ関数の使用の基礎」(『TargetLink 準備およびシミュレーションガイド』📖)
- 「変数の初期化と属性の設定」(『TargetLink API リファレンス』📖)
- 「ds\_error\_none」(『TargetLink API リファレンス』📖)

**新しい PassStructToComponentReferences プロパティ**

TargetLink 2022-B では、DD Variable オブジェクトに PassStructToComponentReferences プロパティが追加されました。PassStructToComponentReferences プロパティを使用して、pointer-to-struct-transport 変数のリーフ構造体コンポーネントを、pointer-to-struct-transport メカニズムを介して参照によって渡すかどうかを指定することができます。

DD アップグレード中、DD VariableClass オブジェクトの Scope プロパティが ref\_param に設定された DD Variable オブジェクトでは、このプロパティは自動的に有効になります。これにより、TargetLink 5.2 の pointer-to-struct メカニズムが必ず使用されます。

Scope プロパティを ref\_param に指定したスクリプト、または Scope プロパティが ref\_param に設定された別の DD VariableClass オブジェクトを参照するスクリプトを使用している場合、PassStructToComponentReferences プロパティは変更されません。その場合、TargetLink 5.2 の pointer-to-struct メカニズムを使用するには、このプロパティを明示的に有効にする必要があります。

**関連ドキュメント**

- PassStructToComponentReferences (TargetLink Data Dictionary リファレンスを参照)
- 「Pointer-to-Struct-Transport メカニズムの基礎」(『TargetLink 準備およびシミュレーションガイド』📖)

**構造体コンポーネントでのアクセス関数テンプレート**

TargetLink 2022-B では、コードジェネレータは構造体のコピーよりも構造体の代入を優先します。構造体コンポーネントが DD AccessFunctionTemplate オブジェクトを参照する DD VariableClass オブジェクトを参照する場合、または親の構造体変数から AccessFunctionTemplate 参照を継承する場合、アクセス関数呼び出しによるカプセル化が行われないことがあります。

カプセル化が確実に行われるようにするには、コンポーネントアクセスを明示的にモデリングするか、AllowStructAssignments コードジェネレータオプションを無効にします。

#### 関連ドキュメント

- 「AllowStructAssignments」(『TargetLink モデルエレメントリファレンス』)

## TargetLink 5.2 と TargetLink 2022-B 間でのコードの変更

### 次のステップ

### 本章の内容

構造関連.....	196
バス構造体関連.....	199
コード順序とループ構造.....	203
スコープの縮小関連.....	204
コード効率性の向上.....	205
Adaptive AUTOSAR 関連.....	208
Classic AUTOSAR 関連.....	211
Simulink Function サブシステム関連.....	214
その他.....	215

## 構造関連

### 変数名の変更

構造に関しては、TargetLink 5.2 の変数とは異なる変数が最適化によって削除される場合があります。その結果、変数名が変更される場合があります。

また、const として修飾された構造化仮パラメータ用に生成される補助変数の名前が変更されています。以下に例を示します。

TargetLink 5.2 以前	TargetLink 2022-B
<code>DE_Record_Aux = *p_DE_Record; foo(DE_Record_Aux);</code>	<code>In = *p_DE_Record; foo(In);</code>

理由 整合性の向上

関連するコード変更 「補助変数の命名」(216 ページ)

## アクセス関数

構造体の代入による構造体コピーが優先されることで、アクセス関数を介してカプセル化された構造体コンポーネントでは、それぞれのコンポーネントが親構造体にアクセスして評価または修正されるため、実質的にこのカプセル化が失われる可能性があります。バス構造体のコンポーネントごとのアクセスをカプセル化するには、バスを逆アセンブルし、それぞれの値を明示的に読み書きするブロックを挿入してから、バスを再アセンブルします。構造体コンポーネントのアクセス関数は、コンポーネントのそれぞれの変数クラスで直接指定するか、またはアクセス関数テンプレートの伝播により、親構造体の AccessFunctionTemplate オブジェクトの PropagateStructToComponents 設定を介して指定することができます。

構造体に適用されるアクセス関数で、その CreateLocalValueCopy プロパティが avoid 以外に設定されている場合、以下のような状況でコードが変更されることがあります。

- 構造体がサブシステム内のブロックの出力である。
- 構造体が関数の中間変数である。

このような場合、アクセス関数の呼び出し回数を最小限に抑えるために、構造体タイプの補助変数がコードに含まれることがあります。

**理由** 整合性の向上

## 実パラメータ

以下のブロックのいずれかを使用して関数の実パラメータに名前を付けると、コードが変更される場合があります。

- Bus Inport ブロック
- Bus Outport ブロック
- Signal Conversion ブロック

最適化により、実パラメータの変数を使用している代入が、条件付きで実行される制御フローに移動する場合があります。関数呼び出しを条件付き実行制御フロー分岐に移動できず、中間変数としての削除が禁止されている場合は、変数の追加と構造体の代入が行われます。

TargetLink 5.2 以前	TargetLink 2022-B
<pre>struct Foo W_E;  ExternFunction(pInput1, &amp;W_E);  if (Input2 &gt; 0) {  ... Read from W_E ...</pre>	<pre>struct Foo Out;  ExternFunction(pInput1, &amp;Out);  if (Input2 &gt; 0) {     struct Foo W_E;      W_E = Out;     ... Read from W_E ...</pre>

**理由** 整合性の向上

**移行の問題** 構造体とそのコンポーネントに関連付けられている変数クラスの [Optimization] プロパティにより、完全に最適化するかまったく最適化しないかのどちらかに設定します。

- 後続の代入が条件付き実行制御フロー分岐に移動しないようにするには、[MOVABLE]と[ERASABLE]の両方のチェックボックスをオフにします。
- 実パラメータを削除できるようにするには、[MOVABLE]と[ERASABLE]の両方のチェックボックスをオンにします。

## 構造体の代入をコンポーネントの代入に置換

TargetLink では、構造体の代入が、構造体コンポーネントへの個別の代入に置き換わる場合があります。これは、十分な数の構造体コンポーネントが使用されておらず、結果として割り当て数が少ないままである場合に行われます。次に例を示します。

TargetLink 5.2 以前	TargetLink 2022-B
S = T ... = S.c	S.c = T.c ... = S.c

この動作は、[「StructAssignmentDecompositionStatementMultiplicationThreshold」](#) (『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を使用して変更することができます。

また、構造体の代入を構造体コンポーネントへの補助変数の代入に置き換えることも可能です。この場合、大多数のコンポーネントが使用されないため、メモリ使用量が大幅に改善されます。次に例を示します。

TargetLink 5.2 以前	TargetLink 2022-B
S = T ... = S.c	Aux_S_c = T.c ... = Aux_S_c

この動作は、[「StructAssignmentDecompositionAuxVarMemoryUsageThreshold」](#) (『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を使用して変更することができます。

**理由** コードの効率性

## 構造化された参照パラメータ

TargetLink 2022-B では、以下を一貫して区別するようになりました。

- 追加の構造化された参照パラメータを関数に渡す。
- pointer-to-struct-transport パラメータを介してコンポーネントを渡す。これには、ルート構造体変数全体の参照も含まれる場合があります(下記参照)。

TargetLink では、構造体配列変数を他の構造体と同様に扱うようになり、また、構造体配列変数を pointer-to-struct-transport 変数としてサポートしていないため、次のようなコード変更が発生する可能性があります。

TargetLink 5.2 以前	TargetLink 2022-B
追加された構造体配列の関数パラメータが、関数呼び出し階層まで誤って伝播されてしまう可能性があります。	追加された構造体配列の関数パラメータは、対応するモデルエレメントで選択された関数にのみ表示されます。これらは伝播されなくなりました。

### ヒント

STRUCT\_INTERFACES デモモデルでは、構造化された関数パラメータのさまざまな利用方法を紹介しています。

pointer-to-struct-transport 変数をパスポートブロックで参照できることに注意してください。これは ref\_param スコープを設定した通常の構造体変数の場合と同様で

す。前者は、すべての呼び出しに対して実変数を 1 つのみ使用してモデルを介して伝播し、後者では、結果の呼び出しごとに 1 つの実変数が使用されます。

理由 整合性の向上

## 関連トピック

### 参考文献

Bus Inport ブロック (『TargetLink モデルエレメントリファレンス』[📄](#))  
 Bus Outport ブロック (『TargetLink モデルエレメントリファレンス』[📄](#))  
 Signal Conversion ブロック (『TargetLink モデルエレメントリファレンス』[📄](#))

## バス構造体関連

### バスに起因する構造体の変更

DD Typedef オブジェクトを使用して構造体を明示的に指定している場合に、バスポートブロックとバス対応ブロックのブロック変数の最適化が改善されました。TargetLink は、構造体変数をコピーするための構造体の代入を高い優先度で作成するようになりました。また、これらの代入に基づいて、スカラー構造体変数をスカラー数値変数と同様の最適化順序で一貫して処理し、可能な限り最適化を実行して、非スカラー構造体変数を数値ベクトルや行列のように扱うようになりました。

TargetLink は、コンポーネントへの割り当てよりも構造体の代入を生成するほうが多くなり、不要な構造体タイプの変数変数をより確実に排除できるようになりました。

この改善の副作用として、次のような状況で生成されるコードがあまり最適化されない場合があります。

- 異なる構造体タイプの間での遷移、特に IMPLICIT\_STRUCT タイプまたは固定名を持つ空の構造体 typedef を使用する場合。
- 非構造体変数からの構造体変数のアセンブリ。  
まれに、他のモデルエレメントからのコードが散在している場合や、条件付き制御フローがある場合など、生成されたコードに余分な構造体変数が残っていることがあります。
- Constant ブロックのブロック変数や Stateflow の初期化変数など、構造体定数から構造体変数を作成する場合。  
静的記憶域期間または自動記憶域期間を持つ初期化された構造体変数が、最適化されたコードに残っており、より高速にコピーできる場合があります。初期値は伝播されません。
  - 以前のバージョンと同様のコードを取得するには、初期化変数を使用せずに値を使用するか、別の構造体タイプを使用してコンポーネント単位の構造体を強制的にコピーしてください。

理由 整合性の向上

### バスでのコンポーネントの代入を減らす

TargetLink 2022-B では、構造体の代入はコンポーネントの代入より優先されます。バスの場合、これにより通常、補助変数の数が減少します。モデリングでバス信号の一部のみが選択されている場合、最適化されたコードに補助変数が残っているこ

とがあります。これは個々の構造体変数が削除され、構造体タイプが削除されずに残されるためです。

#### 理由

- 整合性の向上
- 可読性

関連するコード変更 「構造体の代入をコンポーネントの代入に置換」(198 ページ)

### 信号線分割

構造体の代入の優先による副作用として、信号線分割によって作成されるコードが以前のバージョンのように最適化されない場合があります。

#### 理由 副作用

今後の TargetLink バージョンで修正される予定です。

関連するコード変更 「バスに起因する構造体の変更」(199 ページ)

### Stateflow チャートとバス信号

Stateflow チャートオブジェクトに入力されるバス信号で、SInput オブジェクトの `createinputvariable` が `true` に設定されている場合、コードが以前のバージョンのように最適化されない場合があります。

Stateflow チャートオブジェクトの構造化された入力変数は、構造体の代入で初期化できる場合は、最適化によって削除することができます。以下に例を示します。

```

Main() {
    S = T;
    Chart();
}
Chart()
{
    a = S.a;
    b = S.b;
}

```

これは次のように最適化することができます。

```

Main() {
    Chart();
}

Chart() {
    a = T.a;
    b = T.b;
}

```

構造体タイプの変更や通常の変数から構造体コンポーネントへの遷移により、S にコンポーネント単位でしか代入できない場合、最適化は行われません。以下に例を示します。

```

S.a = a1;
S.b = b1;

```



**移行の問題** 最適化を容易にするために、以下のいずれかの方法を使用することができます。

- S と T に同じ構造体タイプを使用する。
- 構造体タイプを使用しない。
- createinputvariable を false に設定する。

**理由** 副作用

### 構造体コンポーネントの変数クラスを継承しない

以前の TargetLink バージョンでは、バス対応ブロックのブロック変数に対する変数クラスの決定に関するバグがありました。このバグは、[Inherit properties]チェックボックスをオンにしているブロック B が、他のブロック A から構造体タイプを継承した場合に発生していました。TargetLink 2022-B では、次の表に示すように修正されています。

TargetLink 5.2 以前	TargetLink 2022-B
ブロック A からブロック B に構造体コンポーネントの変数クラスを継承する。	構造体コンポーネントの変数クラスはブロック A からは継承されず、ブロック B で TargetLink によって決定される。構造体コンポーネントの変数クラスの[Optimization]プロパティは、構造体のルートに指定された変数クラスの[Optimization]プロパティと同じ値に設定されるようになりました。

**理由**

- バグ修正
- 整合性の向上

**移行の問題** 次の 2 つのケースが確認される場合があります。

1. ブロック A の構造体コンポーネントが ERASABLE ではなく、かつブロック B の構造体のルートが ERASABLE である場合：  
TargetLink でブロック B の構造体変数が削除されるようになりました。
2. ブロック A の構造体コンポーネントが ERASABLE であり、かつブロック B の構造体のルートが ERASABLE ではない場合：  
TargetLink でブロック B の構造体変数は削除されなくなりました。

ブロック B のブロック変数のコンポーネントに対して変数クラスを明示的に指定するには、DD Variable オブジェクトまたは DD Typedef オブジェクトのいずれかと、Component.class プロパティを使用します。

### 関数インターフェースでの構造体配列の参照パラメータ

TargetLink では、参照パラメータとして構造体配列変数をサポートするようになりました。これに伴い、コードジェネレータは、内部のデフォルトテンプレート SLFcnInputStruct と SLFcnOutputStruct を回避せずに使用するようになりました。これらのテンプレートの基本の変数クラスは、デフォルトで ref\_param スコープを持ちます。

結果として、function-call-triggered でないアトミックサブシステムである場合に、以下のようなコード変更が発生します。

TargetLink 5.2 以前	TargetLink 2022-B
バス配列信号を global スコープの変数として渡す、サブシステムの拡張されていないポートを実装しました。	バス配列信号を ref_param スコープの変数として渡す、サブシステムの拡張されていないポートを実装しています。

**移行の問題** TargetLink 5.2 でコードを生成する場合は、以下の手順を実行します。

- Simulink ポートブロックを TargetLink バスポートブロックに拡張します。
- Simulink BusObject でバスを指定し、まだマッピングしていない場合は、`tlSimulinkBusObject('CreateDDTypedef', busObject, propertyName, propertyValue, ...)` API 関数で マッピング構造体タイプを作成し、それを バスポートブロックで参照します。
- バスポートブロックで Scope プロパティが `global` に設定されている DD `VariableClass` を参照します。

**理由** 整合性の向上

### Scaling-Invariant サブシステムの出力ポート

以下のすべての条件の該当する場合、Scaling-Invariant サブシステムのコードが変更されます。

- バス信号は、Scaling-Invariant Bus Outport ブロックを介して Scaling-Invariant サブシステムから転送されます。
- バス信号は、`global` スコープの構造体として実装されます。
- Scaling-Invariant 関数の呼び出し後に、最適化されていないコードで代入の左辺に中間変数が作成されます。

TargetLink 5.2 以前	TargetLink 2022-B
各リーフ構造体コンポーネントは非構造体中間変数に代入されていました。	中間変数はバス信号に生成されたグローバル構造体と同じタイプの構造体です。
<pre>/* call of function: TL_Root/Caller/ScalingInvariant */ Sa3_ScalingInvariant();  /* update(s) for outport TL_Root/Caller/ScalingInvariant/OutPort */ a = Sa3_OutPort.a; b = Sa3_OutPort.b;  /* BusOutport: TL_Root/Caller/OutPort */ Sa2_OutPort.a = a; Sa2_OutPort.b = b;</pre>	<pre>/* call of function: TL_Root/Caller/ScalingInvariant */ Sa3_ScalingInvariant();  /* update(s) for outport TL_Root/Caller/ScalingInvariant/OutPort */ Sa3_OutPort_a = Sa3_OutPort;  /* BusOutport: TL_Root/Caller/OutPort */ Sa2_OutPort = Sa3_OutPort_a;</pre>

**理由** 整合性の向上

### 関連トピック

#### 参考文献

[Bus Outport ブロック](#) (『TargetLink モデルエレメントリファレンス』)

## コード順序とループ構造

### 状態更新とデータストアメモリ更新の位置の変更

TargetLink 2022-B では、構造体の代入が他の最適化より優先されます。ExtendedLifeTimeOptimization コードジェネレータオプションの背後にあるアルゴリズムが、この変更に対応するために再構築されました。

- 中間変数の排除や条件付き実行制御フロー分岐へのコードの移動といった他の最適化アルゴリズムは、状態更新ステートメントやデータストアメモリ更新ステートメントの再スケジューリングよりも優先されます。
- 順序付けが当面の目的には関係ない場合は、更新ステートメントの最初の順序が保持されます。これにより、TargetLink バージョン間でのコードの安定性が向上します。
- Merge ブロックの出力変数や条件付きで実行されるサブシステムの出力変数への値の代入は、状態更新と見なされなくなりました。

このため、構造体には関連しないコードの相違が発生する可能性があります。更新ステートメントでは、以下が確認される場合があります。

- 低いインデックスから高いインデックスへと順に並べ替えられる。
- ペア単位で交換される。
- 状態またはデータストアメモリへの他のアクセスが下方に移動したか、この種の変数が関連していると見なされなくなったために、コード内の位置がさらに下へと変更される。
- ループのマージや制御フロー分岐の結合などの他の制御フローの最適化により、コード内の位置がさらに上へと変更されて、移動の自由度がさらに増す。
- 上記の結果：別のループに配置されます。

「[ループのマージ動作の変化](#)」(203 ページ)も参照してください。

#### 理由

- 整合性の向上
- 今後の TargetLink バージョンでのコードの安定性

移行の問題 「[最適化に関する移行上の注意点](#)」(189 ページ)を参照してください。

### ループのマージ動作の変化

TargetLink 2022-B では、構造体の代入は、他の最適化やアクセス関数を介した構造体コンポーネントのカプセル化よりも優先して行われます。ExtendedLifeTimeOptimization コードジェネレータオプションの背後にあるアルゴリズムの変更に伴い、次のようにループのマージが異なる動作をする場合があります。

- ループのマージが増加する
- ステートメントがループ間で移動する
- 読み取り／書き込みアクセス関数であるために、ループがマージされなくなる
  - ループはアクセス関数の呼び出しで区切られます。
- ループ反復に応じて 1 つの要素にアクセスするアクセス関数について、データフローの見積もりはより控えめになります。つまり、読み取りアクセス関数呼び出しを含むループと書き込みアクセス関数呼び出しを含むループはマージされなくなりました。

理由 整合性の向上

移行の問題 「最適化に関する移行上の注意点」(189 ページ)を参照してください。

#### トリガパターンのループコードの変更

トリガサブシステムでは、LoopUnrollThreshold が使用されるようになりました。これにより、以下の変化が生じる場合があります。

- ループが展開されない場合がある。
- 状態更新コードのループが、より頻繁にマージされる。
- ほとんどの場合、ループ変数のデータタイプがデフォルトの整数タイプ、S32 に変更される。

理由 整合性の向上

## スコープの縮小関連

#### 構造体変数のスコープの縮小

スコープの縮小については、構造体変数が数値変数と同じように扱われるようになりました。構造体変数のスコープがさらに縮小され、コードの変更が生じる場合があります。

理由 整合性の向上

#### 関連するコード変更

- 「初期化でのスコープの縮小」(204 ページ)
- 「変数定義の位置の変化」(204 ページ)

#### 初期化でのスコープの縮小

変数(V)のスコープは、他の変数(X)の初期化で使用される場合、変更されることがあります。

- TargetLink では、V のスコープを X のスコープに縮小できるようになりました。
- 最適化シーケンスの改善により、このようなスコープの縮小は以前のバージョンよりも頻繁に行われます。

理由 コードの効率性

#### 変数定義の位置の変化

TargetLink 2022-B では、構造体のスコープの縮小が改善されました。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。

1. 次の両方の条件を満たす場合、変数定義の位置が変わることがあります。
  - (構造体)変数またはそのコンポーネントの 1 つが、構造体変数 V の定義に使用されている。
  - 最適化中に V のスコープが縮小されている。
2. これにより、他の変数定義の位置もさらに変わる場合があります。

#### 理由

1. コードの効率性
2. コンパイルの容易化

**変更されていない構造体の記憶  
域期間の短縮**

TargetLink は、以下のすべての条件に該当する場合、構造体変数を自動記憶域期間に短縮するようになりました。

- 初期値を保持している。
- 修正されていない。

この動作は、ReduceLifetimeOfConstantVariables コードジェネレータオプションを使用して制御することができます。

**理由** コードの効率性

## コード効率性の向上

**同一オペランドでの比較**

TargetLink の同一オペランドでの比較の最適化が改善されました。以下の式が最適化されるようになりました。

式	代替
$a < a$	0
$a > a$	
$a != a$	
$a <= a$	1
$a >= a$	
$a == a$	

**理由** コードの効率性

**Restart 関数での可変ベクトル幅変数**

同じ初期値を持つ可変ベクトル幅変数の初期化が改善されました。初期化のための余分な補助変数は生成されなくなりました。

TargetLink 5.2 以前	TargetLink 2022-B
<pre>#if SR_CIRCULAR_BUFFER_SIZE == 200 int16_t Temp_a[SR_CIRCULAR_BUFFER_SIZE] = { /* [0..9] */ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, /* [10..19] */ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ... }; #elif SR_CIRCULAR_BUFFER_SIZE == 300 int16_t Temp_a[SR_CIRCULAR_BUFFER_SIZE] = { /* [0..9] */ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, /* [10..19] */ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ... };  ...  for (Aux_S32 = 0; Aux_S32 &lt; SR_CIRCULAR_BUFFER_SIZE; Aux_S32++)  { algSensorHistoryBuff.SensorHistory1_as16[Aux_S32] = Temp_a[Aux_S32]; } }</pre>	<pre>for (Aux_S32 = 0; Aux_S32 &lt; SR_CIRCULAR_BUFFER_SIZE; Aux_S32++)  { algSensorHistoryBuff.SensorHistory1_as16[Aux_S32] = 0; } }</pre>

理由 コードの効率性

**行列アクセスの初期値の伝播の改善**

以下の条件に該当する場合に、行列アクセスの初期値の伝播が改善されました。

- 行列次元のいずれかが展開されていない。
- 初期値がすべて同じである。

TargetLink 5.2 以前	TargetLink 2022-B
<pre>for (Aux_S32 = 0; Aux_S32 &lt; 5; Aux_S32++) { /* Sink: Subsystem/Enabled Subsystem/Sink2 */ Sa2_Sink2[Aux_S32][0] = Struct_OpArg.Matrix[Aux_S32][0]; Sa2_Sink2[Aux_S32][1] = Struct_OpArg.Matrix[Aux_S32][1]; Sa2_Sink2[Aux_S32][2] = Struct_OpArg.Matrix[Aux_S32][2]; Sa2_Sink2[Aux_S32][3] = Struct_OpArg.Matrix[Aux_S32][3]; } }</pre>	<pre>for (Aux_S32 = 0; Aux_S32 &lt; 5; Aux_S32++) { /* Sink: Subsystem/Enabled Subsystem/Sink2 */ Sa2_Sink2[Aux_S32][0] = 42; Sa2_Sink2[Aux_S32][1] = 42; Sa2_Sink2[Aux_S32][2] = 42; Sa2_Sink2[Aux_S32][3] = 42; } }</pre>

理由 コードの効率性

**関数呼び出しの余分なキャストの削除**

加算、減算、乗算の場合、TargetLink は関数呼び出しから不要なキャストを削除するようになりました。

TargetLink 5.2 以前	TargetLink 2022-B
<pre>U16Var = (UInt16)((UInt16)U16ReturningFcn() + 1);</pre>	<pre>U16Var = (UInt16)U16ReturningFcn() + 1);</pre>

理由 コードの効率性

### 構造体変数のカプセル化のための条件付きコンパイル

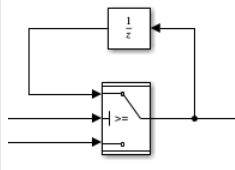
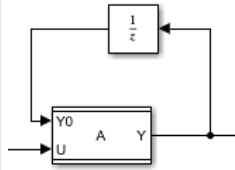
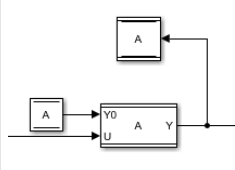
コンポーネントが変数を指す構造体変数に関するコードが変更されました。次のすべての条件に該当する場合、TargetLink では RAM の消費量を減らすために、`#if/#ifdef` 命令でこのような構造体をカプセル化するようになりました。

- 変数がすべて `#if` の範囲で使用されている。
- 可変コンポーネントがすべて `#if` の範囲にある。

理由 コードの効率性

### 信号線分割

以下のモデリングパターンの最適化が改善されました。

モデリングスタイル	モデリング例
Switch ブロックの後に信号線分割が続き、それが Delay ブロックまたは Unit Delay ブロックにフィードバックされます。	
Assignment ブロックの後に信号線分割が続き、それが Delay ブロックまたは Unit Delay ブロックにフィードバックされます。	
Assignment ブロックまたは Switch ブロックの後に信号線分割が続き、それが Data Store Write ブロックにフィードバックされ、対応する Data Store Read ブロックが Assignment ブロックまたは Switch ブロックを駆動します。	

Assignment ブロックについて、次に例を示します。

TargetLink 5.2 以前	TargetLink 2022-B
<pre>for (i = 0; i &lt; 10; i++) {     Assign[i] = X[i]; } Assign[index] = U; ... Use Assign[] for (k = 0; k &lt; 10; k++) {     X[k] = Assign[k];     ... }</pre>	<pre>X[index] = U; ... Use X[]</pre>

理由 コードの効率性

関連するコード変更 「既知のエイリアス関係の活用」(213 ページ)

## 関数のインライン化

以下の条件に該当する場合に、関数のインライン化が変更されました。

- local スコープと static 記憶域期間をもつユーザ固有の変数を使用している。
- 1 つのみの呼び出し元から複数回呼び出される。

TargetLink 5.2 以前	TargetLink 2022-B
インライン化されない	インライン化される

これにより、その後の最適化の順序が変わる可能性があり、通常はコードが改善されます。また、メッセージ W15504 の表示が多くなるまたは少なくなる場合があります。

**移行の問題** 一般的には、DD VariableClass オブジェクトを以下のように指定した変数を指定することをお勧めします。

- Scope プロパティを global に設定している。
- Optimization プロパティを SCOPE\_REDUCIBLE に設定している。
- ScopeReducedClass プロパティが、以下のいずれかを参照している。
  - Scope プロパティを local に設定し、Storage プロパティを static に設定している DD VariableClass オブジェクト。
  - Scope プロパティを local に設定し、Storage プロパティを static に設定している DD VariableClass オブジェクトを含むスコープの縮小チェーン。

静的ローカル変数が必要な場合は、親関数で適切な関数クラスを参照することで、インライン化を禁止することができます。

**理由** コードの効率性

## 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』)

Delay ブロック (『TargetLink モデルエレメントリファレンス』)

Switch ブロック (『TargetLink モデルエレメントリファレンス』)

Unit Delay ブロック (『TargetLink モデルエレメントリファレンス』)

## Adaptive AUTOSAR 関連

## シミュレーションコードの限定的な最適化

次のモデルエレメントから生成されるシミュレーションコードは、完全には最適化されません。

- メソッド呼び出しサブシステム

次の最適化のみが実行されます。

- 構造体の代入の生成
- 変数のスコープの縮小
- 共通の補助変数による複数の変数の置換



## ピヘイピアコードとシミュレーションコードの C++データタイプ

TargetLink が C++モジュール用のコードを生成する場合、タイプ定義の代わりにタイプエイリアスが使用されます。

TargetLink 5.2 以前	TargetLink 2022-B
<code>typedef int MyType[3];</code>	<code>using MyType = std::array&lt;int, 3&gt;;</code>

TargetLink 2022-B では、C++名前空間でのタイプ定義がサポートされます。

```
namespace my {
  namespace types {
    using MyInt16_WithNS = std::int16_t;
    using MyArrV3 = std::array<MyInt16_WithNS, 3>;
  }
}

struct MyStrType {
  my::types::MyInt16_WithNS a;
  std::int16_t b[2][3];
  my::types::MyArrV3 c;
};
```

**理由** ユーザの期待に応えるため

## C++配列データタイプ

ara::core::Array のネイティブサポートにより、Adaptive AUTOSAR 関数インターフェースでは、C 配列の代わりに ara::core::Array データタイプが使用されます。

関数パラメータとして使用される std::array および ara::core::Array データタイプについては、参照が生成されます。

```

/*****
*** FUNCTION:
***   Sa2_FcnVarsWithTypedef
***
*** DESCRIPTION:
***
*** PARAMETERS:
***   Type           Name           Description
***   ~~~~~
***   MyInt16        Sa2_InPort
***   MyArrayV_3 &  Sa2_InPort1
***   MyArrayM_2x3 & Sa2_InPort2
***   MyInt16 *      Sa2_OutPort
***   MyArrayV_3 &  Sa2_OutPort1
***   MyArrayM_2x3 & Sa2_OutPort2
***
*** RETURNS:
***   void
***
*** SETTINGS:
***
*****/
void Sa2_FcnVarsWithTypedef(MyInt16 Sa2_InPort, MyArrayV_3 & Sa2_InPort1,
MyArrayM_2x3 & Sa2_InPort2, MyInt16 * Sa2_OutPort, MyArrayV_3 & Sa2_OutPort1,
MyArrayM_2x3 & Sa2_OutPort2)
```

この関数シグネチャの変更により、以下のアダプターコード関数では、C から C++データタイプへのコピー操作が省略されます。

- イベント:Receive\_および Send\_を参照してください。
- フィールド:Get\_、Set\_、および Update\_。
- メソッド:Call\_、Method\_。

C++データタイプは名前空間に属する場合がありますため、対応する名前空間がデータタイプとともに生成されます。

コードジェネレータは、次のような場合に補助変数を導入します。

- 関数の仮パラメータは C 配列、実パラメータは C++配列。
- 関数の仮パラメータは C++配列、実パラメータは C 配列。

TargetLink 5.2 以前	TargetLink 2022-B
<pre>AdaptiveAutosarFunction_For_MethodBehavior_2(..., STopFunctionSystemID1_InPort1, ...);</pre>	<pre>ds::VectorIdt_W3 Aux_a; // ara::core::Array type Aux_a[0] = STopFunctionSystemID1_InPort1[0]; Aux_a[1] = STopFunctionSystemID1_InPort1[1]; Aux_a[2] = STopFunctionSystemID1_InPort1[2];  AdaptiveAutosarFunction_For_MethodBehavior_2(..., Aux_a, ...);</pre>

理由 コードの効率性

## 無名の構造体タイプ定義なし

C++ファイル内の構造体タイプでは、CreateTypedef プロパティが on または off に設定されているかどうかにかかわらず、TargetLink は固定フォーマットで構造体タイプを生成します。構造体タイプは必ず名前を持ちます。

TargetLink 5.2 以前	TargetLink 2022-B
<pre>struct MyNtdStrType_Typedefs_tag {     MyInt16 aa;     MyArrayV_3 ab; };  typedef struct MyStrType_Typedefs_tag {     MyInt16 aa;     MyArrayV_3 ab; } MyStrType_Typedefs;</pre>	<pre>struct MyNtdStrType_Typedefs {     MyInt16 aa;     MyArrayV_3 ab; };  struct MyStrType_Typedefs {     MyInt16 aa;     MyArrayV_3 ab; };</pre>

これは、C++ファイルで定義されている構造体タイプの使用にも影響します。CreateTypedef プロパティが off に設定されている構造体タイプでは、タイプ名が名前として使用されます。

TargetLink 5.2 以前	TargetLink 2022-B
<pre>extern struct MyNtdStrType_Typedefs_tag MyNtdStrVar_Init;</pre>	<pre>extern MyNtdStrType_Typedefs MyNtdStrVar_Init;</pre>

理由 ユーザの期待に応えるため

## 配列データタイプの定義

TargetLink 2022-B より前のバージョンでは、多次元配列タイプのタイプ定義で基本タイプが使用されていました。TargetLink 2022-B では、DD Typedef オブジェクトで指定されたタイプが使用されます。これはタイプ定義間の依存関係に影響を及ぼし、タイプ定義の並べ替えが変わる可能性があります。

TargetLink 5.2 以前	TargetLink 2022-B
<pre>typedef int16_t MyArrayM_2x3[2][3]; typedef int16_t MyArrayV_3[3]; typedef int16_t MyInt16;  typedef struct MyStrType_NoTypedefs_a_tag {     int16_t aa;     int16_t ab[3]; } MyStrType_NoTypedefs_a;  typedef struct MyStrType_Typedefs_a_tag {     MyInt16 aa;     MyArrayV_3 ab; } MyStrType_Typedefs_a;  typedef struct MyStrType_NoTypedefs_tag {     MyStrType_NoTypedefs_a a;     int16_t b[2][3]; } MyStrType_NoTypedefs;  typedef struct MyStrType_Typedefs_tag {     MyStrType_Typedefs_a a;     MyArrayM_2x3 b; } MyStrType_Typedefs;</pre>	<pre>using MyInt16 = std::int16_t;  struct MyStrType_NoTypedefs_a {     int16_t aa;     int16_t ab[3]; }; using MyArrayV_3 = ara::core::Array&lt;MyInt16, 3&gt;;  struct MyStrType_NoTypedefs {     MyStrType_NoTypedefs_a a;     int16_t b[2][3]; }; using MyArrayM_2x3 = ara::core::Array&lt;MyArrayV_3, 2&gt;;  struct MyStrType_Typedefs_a {     MyInt16 aa;     MyArrayV_3 ab; };  struct MyStrType_Typedefs {     MyStrType_Typedefs_a a;     MyArrayM_2x3 b; };</pre>

また、C 配列データタイプの定義では、基本タイプの代わりにユーザタイプが使用されます。つまり、行列 C 配列データタイプで CreateTypedef プロパティを on に設定したベクトルデータタイプを使用している場合、行列データタイプの定義には、スカラーチェーンメンバの基本タイプではなく、ベクトルデータタイプが使用されます。

TargetLink 5.2 以前	TargetLink 2022-B
<pre>typedef int16_t MyInt16; typedef int16_t MyArrayV_3[3]; typedef int16_t MyArrayM_2x3[2][3];</pre>	<pre>using MyInt16 = int16_t; using MyArrayV_3 = MyInt16[3]; using MyArrayM_2x3 = MyArrayV_3[2];</pre>

理由 ユーザの期待に応えるため

## Classic AUTOSAR 関連

### RTE API 関数での変更

TargetLink 2022-B では、構造体の代入が他の最適化より優先されます。このため、副作用として、構造体には関連しないコードの相違が発生する可能性があります。RTE API 関数に関して、以下のことが確認される場合があります。

- RTE API 関数は、その戻り値を必要とするコード位置により近い場所に配置されます。これは、特にポインタを返す RTE API 関数で確認されます。

#### 理由

- 整合性の向上
- 今後の TargetLink バージョンでのコードの安定性

移行の問題 「最適化に関する移行上の注意点」(189 ページ)を参照してください。

### シミュレーションコードの限定的な最適化

次のモデルエレメントから生成されるシミュレーションコードは、完全には最適化されません。

- 同期オペレーション呼び出しサブシステム
- 非同期オペレーション呼び出しサブシステム
- オペレーション結果プロバイダサブシステム

次の最適化のみが実行されます。

- 構造体の代入の生成
- 変数のスコープの縮小
- 共通の補助変数による複数の変数の置換

### スカラーの実パラメータの最適化の改善

以下のすべての条件に該当する状況で、最適化が改善されました。

- `v` がグローバル変数である。
- `v` のアドレスが、関数呼び出しで実パラメータとして使用されている。

以前のバージョンでは、このような実パラメータは他のスカラー変数のアドレスにのみ置き換えることができました。

TargetLink 2022-B では、このような実パラメータを、ベクトル要素や行列要素のポインタやアドレスに置き換えることができるようになりました。

次に例を示します。

#### TargetLink 5.2 以前

```
Int16 DE_Scalar_a;
Rte_Read_ReceiverPort_DE_Scalar(&DE_Scalar_a); // Write access
SSecond1_Second((sint32) (- (DE_Scalar_a * 2)));
Sa2_Assignment[2] = DE_Scalar_a;

Aux_S32 = *Arg_Scalar;
Rte_Call_new_ClientPort_Scalar_ARGINOUT(instance, &Aux_S32);
*Arg_Scalar = Aux_S32;
```

#### TargetLink 2022-B

```
Rte_Read_ReceiverPort_DE_Scalar(&(Sa2_Assignment[2]));
SSecond1_Second((sint32) (- (Sa2_Assignment[2] * 2)));?
Rte_Call_new_ClientPort_Scalar_ARGINOUT(instance, Arg_Scalar);
```

理由 コードの効率性

### サービランナブル

以前は、サービランナブルに生成されるコードに、行列への不要なポインタが含まれていたことがありました。これは現在修正されています。

#### TargetLink 5.2 以前

```
FUNC(void, Swc_ForOpCall_CODE) RunnableForOpCall(P2CONST(MyScalar, AUTOMATIC, RTE_APPL_DATA)
  Matrix_In)
{
  /* SLLocalConst: Default storage class for local const variables | Width: N.A. */
  const sint32 (*p_Matrix_In)[3];

  p_Matrix_In = (const sint32 (*)[3]) Matrix_In;
}
```

## TargetLink 2022-B

```
FUNC(void, Swc_ForOpCall_CODE) RunnableForOpCall(P2CONST(MyScalar, AUTOMATIC, RTE_APPL_DATA)
  Matrix_In)
{
}
```

理由 バグ修正

名前マクロが\$(Component)ではなく\$(SwComponentSymbolicName)に

TargetLink 2022-B では、RTE API 呼び出しを生成する際のデフォルトの名前マクロは\$(SwComponentSymbolicName)です。

「\$(Component)マクロの置換」(184 ページ)に記載する状況では、名前マクロ\$(Component)は\$(SwComponentSymbolicName)に置き換えられます。

AUTOSAR のバージョンに応じて、名前マクロ\$(SwComponentSymbolicName)は次のように展開されます。

AUTOSAR のバージョン	展開
< 21-11	DD SoftwareComponent オブジェクトの名前に展開されます (\$(Component)と同様)。
>= 21-11	<ul style="list-style-type: none"> <li>DD SoftwareComponent オブジェクトの DD SymbolicName プロパティの値に展開されます。</li> <li>DD SymbolicName プロパティが設定されていない場合、DD SoftwareComponent オブジェクトの名前 (\$(Component)と同様)に展開されます。</li> </ul>

例 SoftwareComponent オブジェクトの名前は Controller です。AUTOSAR バージョンが 21-11 より小さい場合、または AUTOSAR バージョンは 21-11 以上だが、SymbolicName プロパティが設定されていない場合。

```
Rte_Controller_ProvidedSignal_upi;
Rte_ActivationVector_Controller_PosController = activation;
```

AUTOSAR バージョンが 21-11 以上、かつ SymbolicName プロパティが ControllerSymbol に設定されている場合。

```
Rte_ControllerSymbol_ProvidedSignal_upi;
Rte_ActivationVector_ControllerSymbol_PosController = activation;
```

理由 AUTOSAR 準拠

## 既知のエイリアス関係の活用

Classic AUTOSAR に記載するように通信をモデリングする場合、同じバッファに異なるポインタが含まれることがあります。これにより、「信号線分割」(207 ページ)に記載するような状況が発生する可能性があります。

このような状況では、以下のすべての条件に該当する場合に、最適化が改善されます。

- 中間変数を削除すると、アクセスがループに伝播される。
- ポインタが潜在的に同じ変数を指しているエイリアスであるため、生成されるコードが実際には a=a 式となる。

次に例を示します。

TargetLink 5.2	TargetLink 2022-B
<pre>for (Aux_S32 = 0; Aux_S32 &lt; 5; Aux_S32++) {     Sa4_Assignment1[Aux_S32] = p_DataElement3_5[Aux_S32]; }  Sa4_Assignment1[3] = DataElement_In1_3;  for (Aux_S32 = 0; Aux_S32 &lt; 5; Aux_S32++) {     p_DataElement3_5_a[Aux_S32] = Sa4_Assignment1[Aux_S32]; }</pre>	<pre>for (Aux_S32 = 0; Aux_S32 &lt; 5; Aux_S32++) {     p_DataElement3_5_a[Aux_S32] = p_DataElement3_5[Aux_S32]; } p_DataElement3_5_a[3] = DataElement_In1_3;</pre>

ここで、p\_DataElement3\_5 と p\_DataElement3\_5\_a は相互にエイリアスであっても、Sa4\_Assignment1 は削除されます。

これにより、次のような代入になります。

```
p_DataElement3_5_a[Aux_S32] = p_DataElement3_5[Aux_S32];
```

ポインタ変数は同じではないため、この代入は削除されません。TargetLink は、AUTOSAR に準拠しないバッファコードなど、潜在的な問題を回避するために、それらが常に同じ変数を指しているとは仮定しません。

**理由** コードの効率性

**関連するコード変更** 「[信号線分割](#)」(207 ページ)

## Simulink Function サブシステム関連

**ArgIn ブロックが直前にある拡張されていない Inport ブロックの実装**

以下のすべての条件に該当する状況で、Simulink Inport ブロックの実装が変更されました。

- Simulink Inport ブロックが、Simulink Function サブシステムの直前にある ArgIn ブロックに接続されている。
- ArgIn ブロックの変数クラスが default. である。

TargetLink 5.2 以前	TargetLink 2022-B
Simulink Inport ブロックの変数は、ArgIn ブロックのブロック変数とは別に実装され、内部のデフォルトテンプレート S1FunctionInput または S1FunctionInputStruct から取得していました。	ArgIn ブロックのブロック変数をグローバル変数として実装すると、TargetLink は Simulink Inport ブロックのグローバルインターフェース変数として同じ変数を使用しようとします。

**理由** ユーザの期待に応えるため

## 未使用のローカル構造体の削除

次のすべての条件に該当する場合に、Simulink 関数サブシステムの ArgIn ブロックに関するコードが変更されました。

- ArgIn ブロックが pointer-to-struct-transport 変数を参照している。
- 実パラメータの NameTemplate プロパティで pointer-to-struct-transport 変数の名前が指定されていない。
- Simulink Function サブシステムが、Function Caller ブロックで 2 回以上呼び出されている。

TargetLink 5.2 以前	TargetLink 2022-B
TargetLink が誤ってローカルスコープをもつ構造体変数を作成し、それが Simulink Function サブシステムの関数呼び出しの前に更新されますが、この構造体変数が読み取られることはありません。	TargetLink は、ローカルスコープをもつ構造体変数を生成せず、pointer-to-struct-transport 変数を更新して、それを各関数呼び出しで実パラメータとして使用するようになりました。

理由 バグ修正

## 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』)

## その他

## ビット演算式

**Stateflow/MATLAB コード** 最適化中に式の変数が数値定数に置き換えられる場合、Stateflow または MATLAB コードからのビット演算式にキャストが追加されるようになりました。このような場合、数値定数は置換された変数のデータタイプにキャストされます。例として、Int16Var & 3 > 0 を考えてみましょう。

TargetLink 5.2 以前	TargetLink 2022-B
<code>5 &amp; 3 &gt; 0</code>	<code>((Int16) 5) &amp; 3 &gt; 0</code>

これにより、ビット演算の計算でデータタイプが変更されることがなくなります。このキャストは、その後の最適化によって削除される場合があり、その場合は副作用が発生する可能性があります。たとえば、畳み込み式の計算では符号が変わります。

**Stateflow/MATLAB 以外のコード** 次の変更が確認される場合があります。

- ビット演算のオペランドである演算は、Stateflow の CConvention モードで作成されている場合は、畳み込まれません。この場合、Simulink タイプと TargetLink タイプが異なるためです。
- キャストがビット演算のオペランドである場合、浮動小数点定数の整数キャストは、整数定数に置換されません。

理由 バグ修正

**#include 命令**

#include 命令に関して、以下の変更が生じる場合があります。

- 不要な#include 命令は削除されます。
- #include 命令はヘッダーファイル(H)から対応するソースファイル(C)に移動されます。

これは、特に以下のヘッダーファイルのインクルードで確認されます。tl\_basetypes、udt\_<SubsystemID>.h、およびtl\_defines\_<SubsystemID>.h。

利点として、この改善により、ユーザが誤って指定した循環インクルードが削除される可能性があります。

**理由** コンパイル依存の低減

**補助変数の命名**

コードジェネレータの改善により、最適化されていないコードに含まれる補助変数の数が減少する場合があります。その結果、最適化されていないコードに残された補助変数の名前が変更される可能性があります。次の例を考えます。

TargetLink 5.2 以前	TargetLink 2022-B
Aux_, Aux__b, Aux__e, Aux__f	Aux_, Aux__a, Aux__b, Aux__c

**理由** 副作用

**Pointer-to-struct-transport パラメータの追加**

Pointer-to-struct-transport パラメータに関して、まれに以下の変更が確認される場合があります。

TargetLink 5.2 以前	TargetLink 2022-B
実パラメータはグローバルスコープを持ちます	実パラメータはローカル関数です。そのため、より多くの関数呼び出しが pointer-to-struct-transport パラメータを持ちます。つまり、pointer-to-struct-transport 変数の伝播は、関数呼び出し階層のより上位にある関数から開始されます。

**理由** 整合性の向上

**関連するコード変更** 「[スコープの縮小関連](#)」(204 ページ)

**ログマクロの追加**

TargetLink 2022-B では、TargetLink ブロックプロパティ logdata.loggingmode は、Simulink port オブジェクトまたは Stateflow data オブジェクトの DataLogging パラメータにマッピングされます。

その結果、生成された量産コードにログマクロが追加される場合があります。「[TargetLink ログギングプロパティと Simulink データロギングの直接マッピング](#)」(191 ページ)を参照してください。



### 1次元と2次元の Prelookup ブロックの実装の違い

**Simulink と TargetLink のブロックプロパティが同期** 1次元と2次元の Prelookup ブロックに対して生成される量産コードは、以下のすべての条件に該当する場合、変更されることがあります。

プロパティ	値
ImplementLookUpTableFunctionsWithPointerArithmetics コードジェネレータオプション	on
Prelookup ブロックの[Use last breakpoint for input at or above upper limit]チェックボックス	off <sup>1)</sup>

<sup>1)</sup> Prelookup ブロックで参照される DD ルックアップテーブルオブジェクトの新しいデフォルト値による場合もあります。

ルックアップテーブル関数の実装は、次のように変更されました。

TargetLink ≤ 2022-B	TargetLink 2022-B
TabIdxS	TabIdx2S

**移行の問題** 以前のルックアップテーブル関数の実装に戻すには、Prelookup ブロックの[Use last breakpoint for input at or above upper limit]チェックボックスを on に設定します。「[Prelookup ブロックと Interpolation Using Prelookup ブロック](#)」(188 ページ)も参照してください。

#### 理由

- ユーザの期待に応えるため
- 整合性の向上

**致命的に誤ったコードを修正するための新しいテーブル関数の実装** 1次元と2次元の Prelookup ブロックに対して生成される量産コードは、次のすべての条件に該当する場合、変更されることがあります。

プロパティ	値
ImplementLookUpTableFunctionsWithPointerArithmetics コードジェネレータオプション	off
Prelookup ブロックの[Use last breakpoint for input at or above upper limit]チェックボックス	on

ルックアップテーブル関数の実装は、次のように変更されました。

TargetLink ≤ 2022-B	TargetLink 2022-B
TabIdx2S テーブルの最後のブレイクポイントに到達することがなかったため、致命的に誤ったコードになっていました。	TabIdx3S テーブルの最後のブレイクポイントに到達します。

理由 バグ修正

### FIR-filter 関数の呼び出しでの追加の補助変数

安定性の理由から、以下の条件に該当する場合、FIR フィルタ関数の呼び出しに追加の補助変数を実パラメータとして含めるようになりました。

- 係数を保持する配列に const 修飾子がある。
- 配列が FIR-filter 関数に渡される。

これは FIR-filter 関数の仮パラメータが現在、const として修飾されなくなったことに対応するためです。

TargetLink 5.2 以前	TargetLink 2022-B
<pre>Int32 Sa1_FIR_Filter1_coeff_Aux[5]; /* FIR Filter: Subsystem/FIR Filter1 */ *Sa1_FIR_Filter1 = (Int32) (AF_I32TCFIR32_SAT_I32I32(Sa1_FIR_Filter1_coeff_Aux , (UInt32 *) 3, Sa1_FIR_Filter1_coeff_Aux, &amp;p_DelayLine2) &gt;&gt; 1);</pre>	<pre>Int32 Sa1_FIR_Filter1_coeff_Aux[5]; for (Aux_ = 0; Aux_ &lt; 5; Aux_++) {     Sa1_FIR_Filter1_coeff_Aux[Aux_] =     Sa1_FIR_Filter1_coeff[Aux_]; } /* FIR Filter: Subsystem/FIR Filter1 */ *Sa1_FIR_Filter1 = (Int32) (AF_I32TCFIR32_SAT_I32I32(T1AuxVar, (UInt32 *) 3,     Sa1_FIR_Filter1_coeff_Aux, &amp;p_DelayLine2) &gt;&gt; 1);</pre>

**理由** コードの安定性  
今後の TargetLink バージョンで変更される予定です。

### RDI の自動宛先マッピング

RDI (Replaceable Data Items) の自動マッピングは、循環インクルードを防止するように変更されました。これにより、異なるヘッダーになる可能性があります。

**理由** コードの安定性

### 関連トピック

#### 参考文献

ImplementLookUpTableFunctionsWithPointerArithmetics (『TargetLink モデルエレメントリファレンス』 [📄](#))  
Prelookup ブロック (『TargetLink モデルエレメントリファレンス』 [📄](#))

## TargetLink 2022-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.....	219
廃止された制限事項.....	219
廃止された API 関数.....	221
廃止されたフックスクリプト.....	221

## 廃止された TargetLink の機能

### SWC コンテナのやり取りの廃止

TargetLink と SystemDesk 間での SWC コンテナのやり取りは、TargetLink 2022-B でサポートが廃止されました。ただし、ARXML ファイルを使用してソフトウェアコンポーネントを交換することは可能です。dSPACE AUTOSAR Compare を使用すると、これらのファイルを比較およびマージすることができます。「ソフトウェアコンポーネントをやり取りするためのベストプラクティス」(『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。

廃止されるソフトウェア部分は以下のとおりです。

- SystemDesk の Container File Explorer と SWC コンテナの準備および交換のためのコマンド。
- TargetLink の SWC コンテナをやり取りするための Data Dictionary コマンド。
- SWC コンテナの内容を管理するためのスタンドアロンツールである Container Manager。これは TargetLink や SystemDesk から利用できなくなりました。

## 廃止された制限事項

### インクリメンタルコード生成ユニットのインターフェースでの構造体配列変数のサポートなし

TargetLink では、インクリメンタルコード生成ユニットのインターフェースで構造体配列変数をサポートしていません。

### RTE-API 関数のパラメータとしての構造体配列データタイプ

クライアントサーバ通信に属する RTE-API 関数を除き、TargetLink は、RTE-API 関数のファンクションパラメータとしての構造体配列データタイプをサポートしていません。

### CombineControlFlowStatements が構造体の代入の形成の妨げになる

CombineControlFlowStatements コードジェネレータオプションは、リーフ構造体コンポーネントの代入から構造体の代入を形成する妨げになる可能性があります。これは、連続する構造体の代入に再配置できない書き込みアクセスや読み取りアクセスなど、コードジェネレータオプションによって特定の条件を含むループがマージされる場合に発生します。

このような場合、このコードジェネレータオプションがコードサイズや実行時間にもたらす影響は小さくなり、場合によっては逆効果になります。

### Stateflow での temporalCount の使用

temporalCount 関数(状態がアクティブになってから発生したイベント数または経過した秒数を返す)の使用はサポートされません。

### Library ブロックの参照先サブシステムでのデータロギング

参照先サブシステムのインスタンスが Library ブロックの一部として使用されている場合、TargetLink では、参照先サブシステムのインスタンスで信号ロギングをサポートしていません。

---

**ロギングに関する制限事項**

TargetLink の[Start]シミュレーションボタンまたは `t1_sim` コマンドではなく、Simulink の[Run]ボタンまたは Simulink の `sim` コマンドのいずれかを使用してシミュレーションを開始した場合、Stateflow の Local オブジェクトは MIL シミュレーション中にログ記録されません。

---

**Enabled、If Action、または Switch Action サブシステムでの信号のロギング**

信号が次の条件に一致する場合は、MIL シミュレーションモードで時系列データをログ記録することができません。

- 信号が Enabled、If Action、Switch Action のいずれかのサブシステムにある。
- プロット間隔が Inf と異なる。

代わりに、信号の最小値と最大値がログ記録されます。

---

**Fast Restart による参照先サブシステムでの信号のロギング**

モデルに参照先サブシステムのインスタンスが含まれていて、Fast Restart モードがアクティブである場合、TargetLink では、モデル内の信号で以下をサポートしません。

- 信号のロギング
- 最小値／最大値のロギング
- オーバーフロー検出

---

**SVG 画像の消失**

Internet Explorer を使用する場合、次の 2 つの条件が同時に満たされると、生成された HTML ドキュメントに SVG 形式の画像が表示されません。

- 生成されたドキュメントがネットワークドライブに保存されている
- Internet Explorer の互換表示が有効になっている Internet Explorer の [ツール] - [互換表示設定] - [イントラネット サイトを互換表示で表示する] オプションを参照してください。

---

**ブロックでの構造体配列タイプの参照のサポートなし**

ブロックで構造体配列変数を使用するには、DD Variable オブジェクトを参照する必要があります。DD Typedef オブジェクトの参照はサポートされません。

---

**DD Variable オブジェクトの Scope プロパティに関する制限**

TargetLink では、構造体配列変数で参照される VariableClass オブジェクトに対して、Scope プロパティの値 `ref_param` をサポートしていません。

---

**Simulink の[Run]ボタン**

Simulink のモデルツールバーにある[Run]ボタンを使用して MIL モードでシミュレーションを開始した場合、参照先モデル内の信号はログ記録されません。参照先モデル内の信号をログ記録できるようにするには、`t1_sim` API 関数、または以下の TargetLink ウィンドウのいずれかの[Start simulation]ボタンを使用します。

- TargetLink Main Dialog
- TargetLink Plot Overview Window
- TargetLink Detailed Plot Window
- TargetLink Signal Comparison Window

**Interpolation Using Prelookup ブロック**

TargetLink では、Simulink ブロックプロパティ[Valid index input may reach last index]を常に有効にする必要があります (Simulink ブロックダイアログの[Main]ページを参照)。そうしない場合、MIL シミュレーションと SIL/PIL シミュレーションに相違があります。このプロパティが無効な場合、エラーメッセージは表示されません。

**Prelookup ブロック**

TargetLink では、Simulink ブロックプロパティ[Use last breakpoint for input at or above upper limitValid index input may reach last index]を常に有効にする必要があります (Simulink ブロックダイアログの[Main]ページを参照)。そうしない場合、MIL シミュレーションと SIL/PIL シミュレーションに相違があります。このプロパティが無効な場合、エラーメッセージは表示されません。

**関連トピック**

参考文献

[TargetLink Main Dialog ブロック \(『TargetLink モデルエレメントリファレンス』\)](#)

## 廃止された API 関数

**廃止された TargetLink MATLAB API 関数**

関数	ステータス	代替
tl_export_container	エラー <sup>1)</sup>	-

<sup>1)</sup> この関数は TargetLink から削除されました。

## 廃止されたフックスクリプト

**廃止されたフックスクリプトのリスト**

フックスクリプト	説明
tl_pre_containerexport_hook	このフックスクリプトは TargetLink から削除されました。

## 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.....	222
廃止予定の API 関数.....	223
廃止されたコードジェネレータオプション.....	223

## 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

[InPort ブロック \(『TargetLink モデルエレメントリファレンス』\)](#)  
[OutPort ブロック \(『TargetLink モデルエレメントリファレンス』\)](#)  
[SWC ReceiverPort ブロック \(『TargetLink モデルエレメントリファレンス』\)](#)  
[SWC SenderPort ブロック \(『TargetLink モデルエレメントリファレンス』\)](#)

## 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』[📖](#))





# VEOS

## 次のステップ

## 本章の内容

VEOS 2022-B の新機能.....	225
VEOS 2022-B の新機能について概要を説明します。	
VEOS 2022-B の互換性.....	229
VEOS 2022-B の互換性に関する情報を提供します。	
VEOS 2022-B への移行.....	232
VEOS 5.4 から VEOS 2022-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 2023-A での廃止項目.....	233
VEOS 2023-A で廃止される機能について記載します。	

## VEOS 2022-B の新機能

### FMI 3.0 リリースのサポート

VEOS 2022-B では、FMI 3.0 リリースに準拠して実装された FMU もサポートされるようになりました。

FMI 3.0 に関しては、FMI 2.0 の場合と同じ機能がサポートされます。それに加えて、FMI 3.0 で導入された以下の概念と機能がサポートされます。


- 新たに導入されたすべてのデータタイプ (*clock* データタイプを除く) のサポート。これには、センサデータなどの非数値データ処理用の *binary* データタイプも含まれます。
- 配列変数のサポート。

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。

FMI 3.0 リリースの VEOS サポートの詳細については、「Functional Mock-Up Unit (FMU) 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。

### モデルコンテナユーティリティのサポート

VEOS は、dSPACE SIL テストツールチェーンを補完する新しいモデルコンテナユーティリティによってサポートされています。

モデルコンテナユーティリティの詳細については、「Basics on the Model Container Utility」(『Model Container Utility Manual』)を参照してください。

### VEOS を使用した協調シミュレーション

dSPACE は、VEOS を使用した協調シミュレーションを実装するための VeosCoSim インターフェースを提供しています。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マスターとなり、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。

VeosCoSim インターフェースは、VEOS インストレーションには含まれていません。インターフェースおよび関連するユーザドキュメントを入手するには、<http://www.dspace.jp/go/VEOS-Utilities> を参照してください。

### C コードやライブラリを V-ECU に統合する SDK (V-ECU SDK)

dSPACE では、一連の開発プロセスに SIL テストを導入する新しい V-ECU SDK を提供しています。この SDK が提供するコマンドラインツールを使用すれば、お気に入りの IDE で記述した C コードやライブラリを V-ECU に簡単に統合することができます。その後、VEOS を使用して V-ECU をシミュレーションし、デバッグすることができます。

これにより、環境モデルや他の V-ECU を使用したオープンループおよびクローズドループのシナリオで、ECU ソフトウェアの開発、プロトタイプ作成、テストを行うことができるようになります。

V-ECU SDK ユーティリティは、VEOS インストールには含まれていません。ユーティリティを入手するには、<http://www.dspace.jp/go/VEOS-Utilities> を参照してください。

### Linux での V-ECU および SIC のビルド (VEOS Build Console)

VEOS Build Console を Linux で使用して、VEOS 用のクラシックおよびアダプティブ V-ECU インプリメンテーションコンテナ (VECU) と Simulink インプリメンテーションコンテナ (SIC) ファイルをビルドできるようになりました。

詳細については、「VEOS Build Console コマンドリファレンス」(『VEOS/VEOS ユーザ インタフェース リファレンス』)を参照してください。

### SIMPHERA ワークフローのコンテキストでの ISO 26262 認証

SIMPHERA は、クラウドインフラでのコードベースのオフラインシミュレーションに VEOS を使用します。dSPACE は現在、ISO 26262 に準拠した SIMPHERA の認証を取得しています。

**ISO26262 と ASIL** ISO 26262 は、自動車の安全関連の電気/電子 (E/E) システムに関する国際規格です。これは、機能安全に影響を与える電子機能およびソフトウェアが車両で使用されるあらゆる場所に適用されます。

ASIL (Automotive Safety Integrity Level: 自動車安全水準) は、ハザードを防止するために必要なリスク低減のレベルを規定したものです。ASIL は、ISO 26262 に準拠したセーフティクリティカルな E/E システムの開発において、認定されたツールの使用など、必要な手順を決定します。

**SIMPHERA の認証** SIMPHERA は、ASIL D までの ASIL プロジェクトでの使用が認証されています。SIMPHERA は TÜV SÜD から認証を取得しています。その

結果、自動車メーカーやサプライヤーは、プロセス全体の ISO 26262 認証から SIMPHERA を除外できるため、自社のプロセスチェーンの機能安全の証明に専念することができます。

**SIMPHERA と VEOS** SIMPHERA は以下を使用します。

- バーチャル ECU や環境モデルを含むシミュレーションアプリケーションをコンパイルするための VEOS System Builder。
- SIMPHERA 実行フレームワークは、VEOS シミュレータを開始して、クラウドインフラでコードベースのオフラインシミュレーションを実行します。

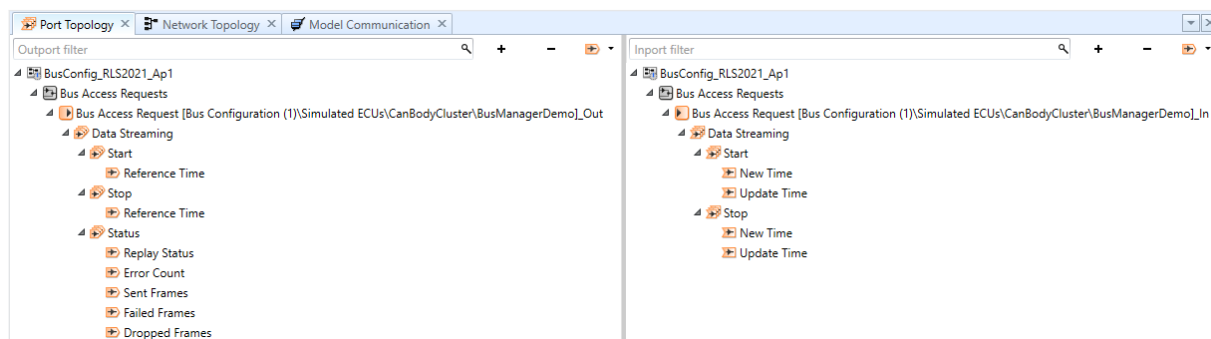
### CAN/Ethernet データリプレイ用ステータス信号

データリプレイが有効なバスアクセス要求ごとに、[Port Topology] ペインには、入出力信号の構造が表示されます。

VEOS 2022-B では、以下の出カステータス信号も使用することができます。

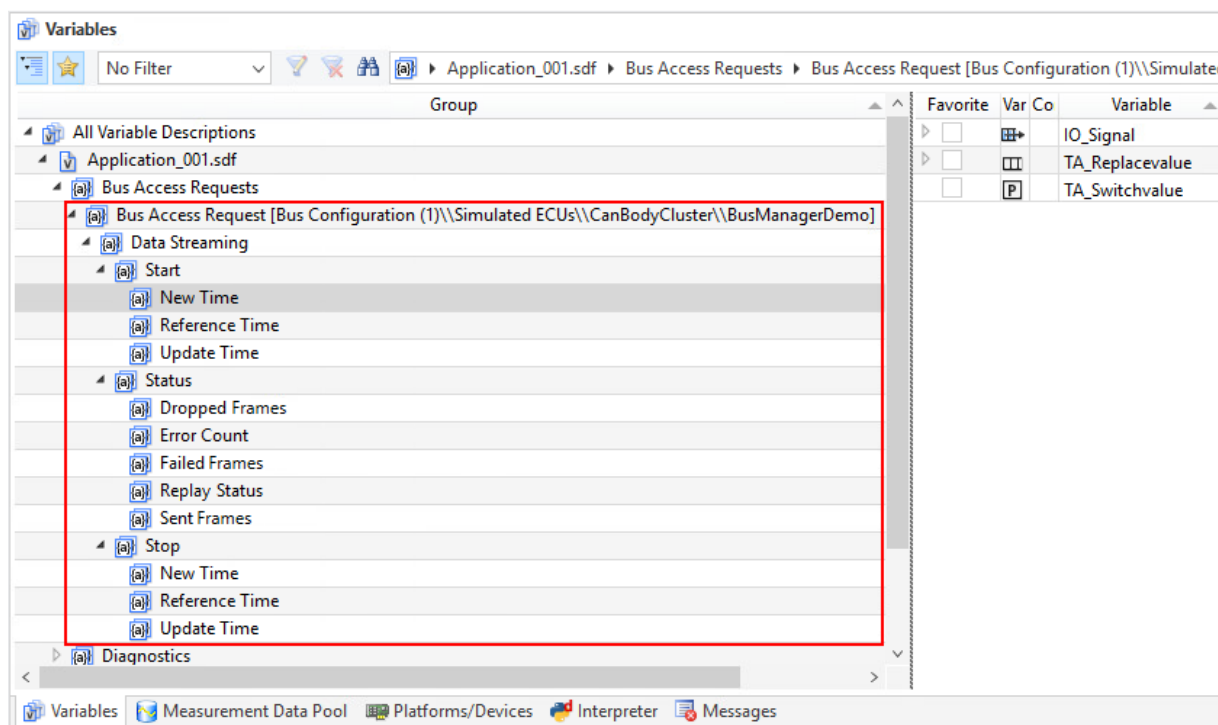
- Replay Status
- Error Count
- Sent Frames
- Failed Frames
- Dropped Frames

次の図に構造例を示します。



出カステータス信号に対応する変数は、ControlDesk で使用できる SDF 変数記述ファイルにも生成されます。

次の図は、関連する変数が表示された ControlDesk の[Variables]ペインを示します。



詳細については、「[VEOS でのバスデータの再生](#)」(『[VEOS マニュアル](#)』)を参照してください。

### MSVC 関連のインストールパスの指定

Simulation Target Manager では、MSVC コンパイラをベースにしたシミュレーションターゲットにも、コンパイラと Windows SDK のインストールパスを指定できるようになりました。

これは、MSVC コンパイラがネットワークドライブにインストールされている場合などに、VEOS がコンパイラのインストールを検出できない場合に便利です。

詳細については、「[Simulation Target Manager](#)」(『[VEOS ユーザ インタフェース リファレンス](#)』)を参照してください。

### MSVC コンパイラバージョン 14.2x のサポート

VEOS は、HostPC32 および HostPC64 シミュレーションターゲットに対して Microsoft Visual C/C++ Compiler バージョン 14.2x をサポートするようになりました。バージョン 14.2x は、Microsoft Visual Studio 2019 に付属しています。

Microsoft Visual Studio 2017 で提供される Version 14.1x のサポートは継続されません。

詳細については、「[Simulation Target Manager](#)」(『[VEOS ユーザ インタフェース リファレンス](#)』)を参照してください。

## VEOS 2022-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カ](#)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

**Windows** VEOS でサポートされている Windows オペレーティングシステムについては、「[オペレーティングシステム](#)」(236 ページ)を参照してください。

**Linux** Ubuntu Linux 20.04 LTS の使用をお勧めします。別の Linux ディストリビューションを使用する場合は、dSPACE サポートにお問い合わせください。

**VEOS の仮想化** クラウド/クラスター環境での 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

VEOS 2022-B は、dSPACE Release 2022-B の Bus Manager で作成されたバスシミュレーションコンテナ (BSC) ファイル (BSC バージョン 1.11) と互換性があります。

- 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 2022-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 ファイルの作成に使用した製品	OSA のバージョン
dSPACE Release 2022-B	22.2
dSPACE Release 2022-A	5.4 <sup>1)</sup>
dSPACE Release 2021-B	5.3 <sup>1)</sup>
dSPACE Release 2021-A	5.2 <sup>1)</sup>


<sup>1)</sup> 5.4 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実験サービスの有効化/無効化はできます。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(CTLGZ/VECU、SIC、BSC、FMU)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 2022-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 バージョン <sup>1)</sup>
dSPACE Release 2022-B: <ul style="list-style-type: none"> <li>▪ Model Interface Package for Simulink 2022-B</li> <li>▪ TargetLink 2022-B</li> </ul>	1.11
dSPACE Release 2022-A: <ul style="list-style-type: none"> <li>▪ Model Interface Package for Simulink 22.1</li> <li>▪ TargetLink 5.2</li> </ul>	1.10 <sup>2)</sup>
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>	1.10 <sup>2)</sup>
dSPACE Release 2021-A: <ul style="list-style-type: none"> <li>▪ Model Interface Package for Simulink 4.5</li> <li>▪ TargetLink 5.1</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できる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 2022-B (dSPACE Release 2022-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 2022-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
dSPACE Release 2022-B: ▪ VEOS 2022-B	1.2
dSPACE Release 2022-A: ▪ SYNECT 2.13 ▪ VEOS 5.4	1.2
dSPACE Release 2021-B: ▪ SYNECT 2.12 ▪ VEOS 5.3	1.2
dSPACE Release 2021-A: ▪ SYNECT 2.11 ▪ VEOS 5.2	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 2022-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 2022-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
dSPACE Release 2022-B: ▪ SystemDesk 2022-B ▪ TargetLink 2022-B	3.2
dSPACE Release 2022-A および 2021-B: ▪ SystemDesk 5.6 ▪ TargetLink 5.2	3.1
dSPACE Release 2021-A および 2020-B: ▪ SystemDesk 5.5 ▪ TargetLink 5.1	3.0

## Real-Time Testing 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 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 シミュレータ	RTT のバージョン
VEOS 5.1	Real-Time Testing Version 4.4
VEOS 5.0	Real-Time Testing Version 4.3

ControlDesk 2022-B は、自動的に VEOS 2022-B の VEOS シミュレータを使用しません。したがって、PC 上で Real-Time Testing Version 2022-B 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。

### AUTOSAR Adaptive Platform の互換性

AUTOSAR アダプティブ V-ECU のシミュレーションでは、VEOS 2022-B は AUTOSAR Adaptive Platform の次のリリースと互換性があります。

- R20-11

### 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 2022-B への移行

### 概要

VEOS 5.4 から VEOS 2022-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

5.4 より前のバージョンから VEOS 2022-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。


### ASM モデルの移行

VEOS インストールが属する 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 2023-A での廃止項目

#### VEOS 4.0~4.4 でのシミュレーション

VEOS 2023-A 以降がインストールされている場合、以下の VEOS バージョンでは、同一 PC でのオフラインシミュレーションができなくなります。

- dSPACE Release 2017-A の VEOS 4.0
- dSPACE Release 2017-B の VEOS 4.1
- dSPACE Release 2018-A の VEOS 4.2
- dSPACE Release 2018-B の VEOS 4.3
- dSPACE Release 2019-A の VEOS 4.4

ただし、これらのバージョンでオフラインのシミュレーションアプリケーションを構築、設定することには影響はなく、引き続き可能です。



# 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.....	235
オペレーティングシステム.....	236
仮想マシン (VM) での dSPACE ソフトウェアの使用.....	239
dSPACE ソフトウェアのランタイム互換性.....	243
Windows 機能の使用に関する制限事項.....	244
Linux 機能の使用に関する制限事項.....	246

## 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 2022-B の各コンポーネントによるサポート					
	RCP and HIL 2022-B <sup>1), 2)</sup>	AutomationDesk 2022-B <sup>3)</sup>	TargetLink 2022-B	Model Compare 2022-B	dSPACE Python Extensions 2022-B <sup>4)</sup>	XIL API .NET MAPort 2022-B
R2022b	✓ <sup>5)</sup>	✓	✓	✓	✓	✓
R2022a	✓ <sup>5)</sup>	✓	✓	✓	✓	✓
R2021b	✓	✓	✓	✓	✓	✓
R2021a	✓	✓	✓	✓	✓	✓

1) 'RCP and HIL Software'は、ASM、RTI、ConfigurationDesk、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。

2) MATLAB/Simulink Student Suite は、Automotive Simulation Models (ASM) ではサポートされません。

3) AutomationDesk の MATLAB Access ライブラリには MATLAB が必要です。

4) dSPACE Python Extensions の matlplotlib2 には MATLAB が必要です。

5) R2022a と R2022b は、FPGA Programming Blockset – FPGA Interface ではサポートされません。

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 2022-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
  - Long-Term Servicing Channel: 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

とになります。dSPACE は、General Availability Channel の 21H2 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(64 ビット版)
  - Windows 11 Home エディションはサポートされません。

#### 注記

FPGA Programming Blockset の場合 :FPGA Programming Blockset - Processor Interface のみ、Windows 11 をサポートしています。Windows 11 は、Xilinx のサポートがないため、FPGA アプリケーションのモデリングやハンドコーディングに使用することはできません。

- General Availability Channel: Microsoft の互換性に関する記述が適用されません。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 21H2 バージョンをテストに使用しました。
- 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。

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](#)」(244 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 2022-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](#)」(246 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。

## MicroAutoBox Embedded PC をホスト PC として使用

ControlDesk は、以下にインストールすることもできます。

- 以下で動作する MicroAutoBox III Embedded PC  
Microsoft® Windows® 10 IoT Enterprise LTSC 2019、64 ビット版

**SYNECT サーバ上のオペレーティングシステム**

SYNECT サーバでは、次のオペレーティングシステムをサポートしています。

- dSPACE Release 2022-B のすべての dSPACE 製品について、上記のオペレーティングシステムと同じオペレーティングシステム。
- Windows Server 2016

**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す。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 2022-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 を起動時すると、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
ASM	✓	—	—
AutomationDesk	—	✓ 既知の制限事項: ▪ 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。 ▪ 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。 ▪ DS1104 R&D Controller Board へのアクセスは行うことができません。	—
Bus Manager	✓	—	—
ConfigurationDesk for RapidPro	✓	—	—
ConfigurationDesk	✓	—	—
Container Manager	✓	—	—
ControlDesk	—	✓ 既知の制限事項: ▪ 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。 ▪ 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。 ▪ DS1104 R&D Controller Board へのアクセスは行うことができません。	—
Data Dictionary Manager	✓	—	—
DCI-GSI Configuration Package	✓	—	—



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 AUTOSAR Compare	✓	—	—
dSPACE Installation Manager	✓	—	—
ECU Flash Programming Tool	✓	—	—
ECU Interface Base Package	✓	—	—
ECU bypassing target compiler	✓	—	—
Firmware Archives	—	✓ 既知の制限事項: ▪ 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。 ▪ 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。 ▪ DS1104 R&D Controller Board へのアクセスは行うことができません。	—
Firmware Manager	—		—
FlexRay Configuration Tool	✓	—	—
Model Compare	✓	—	—
ModelDesk	—	✓ 既知の制限事項: ▪ 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。 ▪ 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: ▪ 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。 ▪ 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。 ▪ DS1104 R&D Controller Board へのアクセスは行うことができません。	—
Real-Time Testing	—	✓ 既知の制限事項: ▪ 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。 ▪ 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。 ▪ DS1104 R&D Controller Board へのアクセスは行うことができません。	—
RTI Blocksets (Real-Time Interface)	—	✓ 既知の制限事項: ▪ 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。 ▪ 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。 ▪ DS1104 R&D Controller Board へのアクセスは行うことができません。	—
Sensor Simulation	—	—	✓ <sup>1)</sup>
SYNECT	✓	—	—
SYNECT Server	✓	—	—

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
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』[11](#))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](#)』[11](#))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。

**dongleベースのデバイスの接続** dongle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](#)』[11](#))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 は無効になりません。
  - 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

### 定義

ランタイム互換性とは、以下のことを意味します。

- 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
- 相互作用なく個別に dSPACE 製品を使用可能

### dSPACE Release 2022-B の製品互換性

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。

以下の点を確認してください。

- 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。  
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。  
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、<https://www.dspace.jp/go/CompPatch> を参照してください。
- Release 2022-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。

#### SCALEXIO システムまたは MicroAutoBox III の使用に関する推奨事項

SCALEXIO システムまたは MicroAutoBox III で使用する製品には互換性が必要です。同一の dSPACE Release で提供される製品でのみ互換性が保証されます。詳細については、dSPACE までお問い合わせください。

**DS1006、DS1104、または MicroAutoBox II プラットフォームにロードされるリアルタイムアプリケーションの互換性** リアルタイムアプリケーションが dSPACE Release 2016-B 以降のソフトウェア製品を使用して関連するプラットフォームのいずれかにロードされた場合、dSPACE Release 2016-A(以前)のソフトウェア製品は、ロードされたリアルタイムアプリケーションがホスト PC に保存されたリアルタイムアプリケーションと同一であることを検出しません。この場合、関連するソフトウェア製品を制限なしで使用することができません。

これと同じことが、dSPACE Release 2016-A 以前のソフトウェア製品を使用してリアルタイムアプリケーションをロードし、dSPACE Release 2016-B 以降のソフトウェア製品を使用して試験などを実行する場合にも適用されます。

以前のリリースの dSPACE 製品との併用

以前のリリースの複数の製品を併用する場合の詳細と注意事項については、[https://www.dspace.jp/goto.cfm/ja\\_0501](https://www.dspace.jp/goto.cfm/ja_0501) を参照してください。

## Windows 機能の使用に関する制限事項

目的	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 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	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。
ユーザの簡易切り替えのサポートなし	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。

---

**PCをシャットダウンする前に dSPACE ソフトウェアを閉じる**

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**

以下の dSPACE ソフトウェア製品には、高 DPI モニターの使用に関する制約があります。

- SYNECT: 高 DPI モニターを使用する場合、SYNECT GUI は適切に表示されない場合がありますが、機能上の制約の原因とはなりません。
- Real-Time Testing: 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 35
- ASM Environment Blockset
  - 新機能 37
  - 移行 37
- ASM Fuel Cell Blockset
  - 新機能 38
  - 移行 39
- ASM Traffic Blockset
  - 移行 42
- ASM Utils
  - 新機能 40
- ASM Utils Blockset
  - 移行 40
- ASM Vehicle Dynamics Blockset
  - 新機能 44
- ASM ブロックセット
  - 新機能 34
- AutomationDesk
  - 新機能 29
  - 移行 29

**B**

- Bus Manager(スタンドアロン)
  - 新機能 45
  - 移行 47

**C**

- ConfigurationDesk
  - 新機能 51
  - 移行 60
- ConfigurationDesk の Bus Manager
  - 新機能 55
- ControlDesk
  - 新機能 64
  - 移行 71

**D**

- DCI Configuration Tool
  - 新機能 79
- dSPACE FlexRay Configuration Package
  - 新機能 81
- dSPACE Python Extensions
  - 新機能 85
- dSPACE XIL API
  - 新機能 87
  - 移行 88

**E**

- ECU Interface Manager
  - 新機能 91
  - 移行 93

**F**

- Firmware Manager
  - 新機能 95
- FPGA Programming Blockset
  - 新機能 97

**L**

- Linux
  - 制限事項 246
- Linux 機能の使用に関する制限事項 246

**M**

- MATLAB
  - サポートされるリリース 235
  - 要件 235
- MicroAutoBox III のファームウェア
  - 新機能 103
- Model and Sensor Interface
  - 移行 112
- Model and Sensor Interface Blockset
  - 新機能 111
- Model Compare
  - 新機能 105
  - 移行 107
- Model Interface Package for Simulink
  - 新機能 117
  - 移行 118
- ModelDesk
  - 新機能 115
  - 移行 115
- MotionDesk
  - 新機能 119
  - 移行 119

**R**

- Real-Time Testing
  - 移行 122
- RTI Bypass Blockset
  - 移行 125
- RTI CAN MultiMessage Blockset
  - 移行 127
- RTI FPGA Programming Blockset
  - 移行 100
- RTI LIN MultiMessage Blockset
  - 移行 129
- RTI Synchronized Time Base Manager Blockset
  - 新機能 131
- RTI/RTI-MP
  - 新機能 123
- RTLlib
  - 新機能 123

**S**

- SCALEXIO Firmware
  - 新機能 133
  - 移行 134
- Sensor Simulation
  - 新機能 135
  - 移行 135
- SystemDesk
  - 新機能 146

**T**

- TargetLink
  - AUTOSAR 機能、新規  
サポートされるリリース 161

TargetLink Data Dictionary

- 移行 172
- ライブラリとモデルの手動アップグレード 175
- 廃止されたドキュメント 172
- 既存のデータディクショナリのアップグレード 174

**V**

- VEOS
  - 新機能 225

**W**

- Windows
  - 制限事項 244
- Windows 機能の使用に関する制限事項 244

**サ**

- サポートしている MATLAB リリース 235

**シ**

- システム要件
  - Linux オペレーティングシステム 237
  - Windows オペレーティングシステム 236
- 新機能
  - ConfigurationDesk 51

**ト**

- ドキュメントフォルダ 12

**ハ**

- バージョン履歴 18

**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 237
  - MATLAB 235
  - Windows オペレーティングシステム 236

**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 21
- 共通プログラムデータフォルダ 12
- 廃止 16
  - ソフトウェアのサポート 17
  - ハードウェア 17
  - 予定(ソフトウェアサポート) 17
  - 予定(ハードウェア) 17
- 新機能
  - ASM Battery Blockset 35
  - ASM Environment Blockset 37
  - ASM Fuel Cell Blockset 38
  - ASM Utils 40
  - ASM Vehicle Dynamics Blockset 44
  - ASM ブロックセット 34
  - AutomationDesk 29
  - Bus Manager(スタンドアロン) 45

- ConfigurationDesk の Bus Manager 55
  - ControlDesk 64
  - DCI Configuration Tool 79
  - dSPACE FlexRay Configuration Package 81
  - dSPACE Python Extensions 85
  - dSPACE XIL API 87
  - ECU Interface Manager 91
  - Firmware Manager 95
  - FPGA Programming Blockset 97
  - MicroAutoBox III のファームウェア 103
  - Model and Sensor Interface Blockset 111
  - Model Compare 105
  - Model Interface Package for Simulink 117
  - ModelDesk 115
  - MotionDesk 119
  - RTI Synchronized Time Base Manager Blockset 131
  - RTI/RTI-MP 123
  - RTLib 123
  - SCALEXIO Firmware 133
  - Sensor Simulation 135
  - SystemDesk 146
  - VEOS 225
- 移行
- ASM Environment Blockset 37
  - ASM Fuel Cell Blockset 39
  - ASM Traffic Blockset 42
  - ASM Utils Blockset 40
  - AutomationDesk 29
  - Bus Manager (スタンドアロン) 47
  - ConfigurationDesk 60
  - ControlDesk 71
  - dSPACE XIL API 88
  - ECU Interface Manager 93
  - Model and Sensor Interface 112
  - Model Compare 107
  - Model Interface Package for Simulink 118
  - ModelDesk 115
  - MotionDesk 119
  - Real-Time Testing 122
  - RTI 123
  - RTI Bypass Blockset 125
  - RTI CAN MultiMessage Blockset 127
  - RTI FPGA Programming Blockset 100
  - RTI LIN MultiMessage Blockset 129
  - SCALEXIO Firmware 134
  - Sensor Simulation 135
  - XIL API ライブラリ 30
- 製品の概要 18
- 要件
- ホスト PC の Linux オペレーティングシステム 237
  - ホスト PC の Windows オペレーティングシステム 236
  - ホスト PC のソフトウェア
    - MATLAB 235