OMRON

マシンオートメーションコントローラ

NJ/NXシリーズ

データベース接続CPUユニット

ユーザーズマニュアル

形NX701-□□20
形NX502-1□00
形NX102-□□20
形NJ501-□□20
形NJ101-□□20

CPUユニット



SYSMAC
always in control

SBCA-411AC

おことわり

- 本マニュアルの内容の一部または全部を無断で複写、複製、転載することを禁じます。
- 本マニュアルの内容に関しては、改良のため予告なしに仕様などを変更することがあります。
  あらかじめご了承ください。
- 本マニュアルの内容に関しては、万全を期しておりますが、万一ご不審な点や誤りなど、お気づきのことがありましたら、お手数ですが、弊社支店または営業所までご連絡ください。
  その際、巻末記載の Man.No.（マニュアル No.）も併せてお知らせください。

商標

- Sysmac は、オムロン株式会社 FA 機器製品の日本およびその他の国における商標または登録商標です。
- Microsoft、Windows、Windows Vista、Excel、SQL Server、Microsoft Edge は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。
- EtherCAT®は、ドイツ Beckhoff Automation GmbH によりライセンスされた特許取得済み技術であり登録商標です。
- ODVA、CIP、CompoNet、DeviceNet、EtherNet/IP は ODVA の商標です。
- SD、SDHC ロゴは、SD-3C,LLC の商標です。
- Oracle、Java、MySQL は、Oracle Corporation およびその子会社、関連会社の米国およびその他の国における登録商標です。
- IBM、DB2 は、米国およびその他の国における International Business Machines Corporation の登録商標です。
- Firebird は、Firebird Foundation Incorporated.の登録商標です。
- PostgreSQL は、PostgreSQL Global Development Group の登録商標です。

その他、本文中に掲載している会社名および製品名は、それぞれ各社の商標または登録商標です。

著作権

- スクリーンショットはマイクロソフトの許諾を得て使用しています。
- 本製品にはサードパーティ製のソフトウェアが組み込まれています。このソフトウェアに関連するライセンスと著作権については、 http://www.fa.omron.co.jp/nj_info_j/ をご覧ください。

はじめに

このたびは、NJ/NX シリーズ CPU ユニットをお買い上げいただきまして、ありがとうございます。本マニュアルは、NJ/NX シリーズ CPU ユニットでデータベース（以下 DB と表記する場合があります。）接続サービスを使用する上で、必要な情報を記載しています。お使いになる前に本マニュアルをよく読んで、機能・性能などを十分ご理解いただき、システムの構築にご活用ください。
また、お読みになった後も本マニュアルは大切に保管して、いつも手元においてお使いください。

対象となる読者の方々

本マニュアルは、次の方を対象に記述しています。
電気の知識（電気工事士あるいは同等の知識）を有する方で
- FA 機器の導入を担当される方
- FA システムを設計される方
- FA 機器を設置、接続される方
- FA 現場を管理される方

また、プログラミング言語に関して、国際標準規格 IEC 61131-3 もしくは国内標準規格 JIS B 3503 に規定される内容をご理解いただいている方を対象に記述しています。

対象となる製品

本マニュアルは以下の製品を対象にしています。
- NX シリーズ データベース接続 CPU ユニット
  a）形 NX701-□□20
  b）形 NX502-1□00
  c）形 NX102-□□20
- NJ シリーズ データベース接続 CPU ユニット
  a）形 NJ501-□□20
  b）形 NJ101-□□20
- Sysmac Studio
  a）形 SYSMAC-SE2□□□
  形 NX701-□□20 の場合：Ver.1.21 以降
  形 NX502-1□00 の場合：Ver.1.54 以降
  形 NX102-□□20 の場合：Ver.1.24 以降
  形 NJ501-1□20、形 NJ501-4□20、形 NJ101-□□20 の場合：Ver.1.14 以降
  形 NJ501-R□20 の場合：Ver.1.44 以降

分冊構成

本製品のマニュアルは、次表のとおりに分冊されています。目的に応じてお読みいただき、本製品を活用してください。
本製品の操作には、おもにオートメーションソフトウェア Sysmac Studio を使用します。Sysmac Studio については、『Sysmac Studio Version 1 オペレーションマニュアル (SBCA-470)』 を参照してください。

使用目的	マニュアル													
	基本情報													
	NX シリーズ CPU ユニット ユーザーズマニュアル ハードウェア編	NX シリーズ 形 NX502 CPU ユニット ユーザーズマニュアル ハードウェア編	NX シリーズ 形 NX102 CPU ユニット ユーザーズマニュアル ハードウェア編	NJ シリーズ CPU ユニット ユーザーズマニュアル ハードウェア編	NJ/NX シリーズ CPU ユニット ユーザーズマニュアル ソフトウェア編	NJ/NX シリーズ コマンドリファレンスマニュアル 基本編	NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編	NJ/NX シリーズ コマンドリファレンスマニュアル モーション編	NJ/NX シリーズ CPU ユニット 内蔵 EtherCAT ポート ユーザーズマニュアル	NJ/NX シリーズ CPU ユニット 内蔵 EtherNet/IP ポート ユーザーズマニュアル	NJ/NX シリーズ CPU ユニット ユーザーズマニュアル FINS 機能編	NJ/NX シリーズ データベース接続 CPU ユニット ユーザーズマニュアル	NJ シリーズ ロボット統合 CPU ユニット ユーザーズマニュアル	NJ/NX シリーズ CPU ユニット トラブルシューティングマニュアル
形 NX701 の概要を知る	○													
形 NX502 の概要を知る		○												
形 NX102 の概要を知る			○											
NJ シリーズの概要を知る				○										
装着、設置、ハードウェア設定を行う	○	○	○	○										
モーション制御を行う場合							○							
EtherCAT を使用する場合									○					
EtherNet/IP を使用する場合										○				
データベース接続サービスを使用する場合												○		
ソフトウェア設定を行う					○									
モーション制御を行う場合							○							
EtherCAT を使用する場合									○					
EtherNet/IP を使用する場合										○				
FINS を使用する場合											○			
データベース接続サービスを使用する場合												○		
オムロン製ロボットの制御を行う場合													○	
ユーザプログラムを作成する					○	○								
モーション制御を行う場合							○	○						
EtherCAT を使用する場合									○					
EtherNet/IP を使用する場合										○				
FINS を使用する場合											○			
データベース接続サービスを使用する場合												○		
オムロン製ロボットの制御を行う場合													○	
異常処理を組み込む場合														○

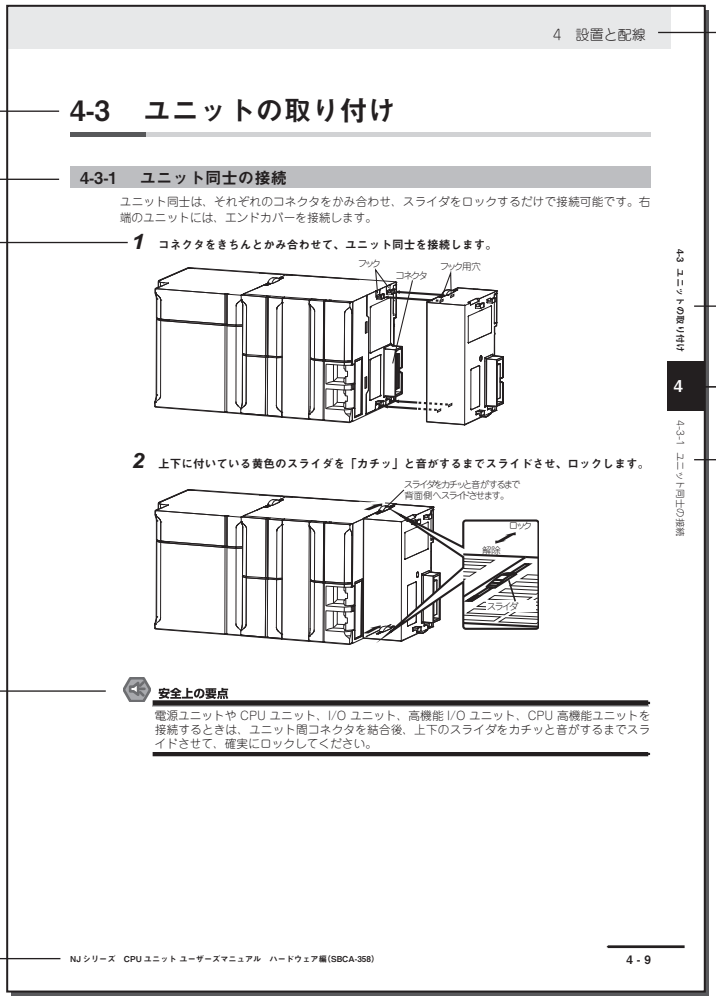
使用目的	マニュアル													
	基本情報													
	NX シリーズ CPU ユニット ユーザーズマニュアル ハードウェア編	NX シリーズ 形 NX502 CPU ユニット ユーザーズマニュアル ハードウェア編	NX シリーズ 形 NX102 CPU ユニット ユーザーズマニュアル ハードウェア編	NJ シリーズ CPU ユニット ユーザーズマニュアル ハードウェア編	NJ/NX シリーズ CPU ユニット ユーザーズマニュアル ソフトウェア編	NJ/NX シリーズ コマンドリファレンスマニュアル 基本編	NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編	NJ/NX シリーズ コマンドリファレンスマニュアル モーション編	NJ/NX シリーズ CPU ユニット 内蔵 EtherCAT ポート ユーザーズマニュアル	NJ/NX シリーズ CPU ユニット 内蔵 EtherNet/IP ポート ユーザーズマニュアル	NX シリーズ CPU ユニット ユーザーズマニュアル FINS 機能編	NJ/NX シリーズ データベース接続 CPU ユニット ユーザーズマニュアル	NJ シリーズ ロボット統合 CPU ユニット ユーザーズマニュアル	NJ/NX シリーズ CPU ユニット トラブルシューティングマニュアル
動作確認やデバッグを行う														
モーション制御を行う場合							○							
EtherCAT を使用する場合									○					
EtherNet/IP を使用する場合					○					○				
FINS を使用する場合											○			
データベース接続サービスを使用する場合												○		
オムロン製ロボットの制御を行う場合													○	
異常管理機能やトラブル発生時の対処方法を知る*1											△	△	△	○
保守作業について知る	○	○	○	○										
モーション制御を行う場合							○							
EtherCAT を使用する場合									○					
EtherNet/IP を使用する場合										○				

*1. 異常管理の考え方と異常項目については、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。ただし、△ マークがついているマニュアルの製品に特有の異常の詳細については、△ マークがついているマニュアルを参照してください。

マニュアルの読み方

ページ構成

このマニュアルの各ページの構成は次のとおりです。



節タイトル

項タイトル

操作手順

操作手順を示します。

注記、補足、参照先

注記、補足、参照先などの各項目を、アイコンによって示しています。

マニュアル名称

章タイトル

節タイトル

項タイトル

現在のページの章／節／項タイトルをそれぞれ示します。

章番号

現在のページの章番号を示します。

このページは、説明用の見本です。実際の内容とは異なります。

アイコン

このマニュアルで使われているアイコンには、次のような意味があります。

安全上の要点

製品を安全に使用するために実施または回避すべきことを示します。

使用上の注意

製品の動作不能、誤動作、または性能・機能への悪影響を予防するために実施または回避すべきことを示します。

参考

必要に応じて読んでいただきたい項目です。

知っておくと便利な情報や、使用する上で参考となる内容について説明しています。

バージョン関連情報

コントローラや Sysmac Studio の、バージョンによる性能や機能の違いについて説明しています。

表記についての留意事項

本マニュアルでは、Sysmac Studio から実機コントローラにデータを転送することを“ダウンロード”、実機コントローラからデータを Sysmac Studio に転送することを“アップロード”と記述しています。

Sysmac Studio では、“ダウンロード”および“アップロード”は、いずれも“同期”機能を使用します。Sysmac Studio の“同期”機能は、パソコン上のデータと実機コントローラのデータを自動照合し、どちらの方向に転送するかをユーザが選択する機能です。

目次構成

1 DB接続サービスの概要
2 DB接続設定
3 DB接続機能のプログラミング
4 DB接続サービスの開始／コネクションの接続と状態確認
5 各種の機能
6 稼働ログの使用手順
7 DB接続命令
8 異常の確認と処置
A 付録
I 索引

目次

はじめに **1**
対象となる読者の方々 1
対象となる製品 1

分冊構成 **2**

マニュアルの読み方 **4**
ページ構成 4
アイコン 4
表記についての留意事項 5

目次構成 **7**

ご承諾事項 **15**

安全上の注意 **17**

安全上の要点 **18**

使用上の注意 **19**

法規と規格 **20**

バージョン **21**
バージョンの種類 21
バージョンの確認方法 21
CPU ユニットのユニットバージョンと Sysmac Studio のバージョン 23

関連マニュアル **24**

用語説明 **26**

マニュアル改訂履歴 **28**

第 1 章　DB 接続サービスの概要

1-1　概要と特長 **1-2**
1-1-1　概要 1-2
1-1-2　特長 1-3
1-2　DB 接続サービスの仕様としくみ **1-4**
1-2-1　DB 接続サービスの仕様 1-4
1-2-2　DB 接続機能のしくみ 1-10
1-3　DB 接続サービスの使用手順 **1-13**

第 2 章　DB 接続設定

2-1　Sysmac Studio の起動と新規プロジェクト作成 **2-2**
2-1-1　Sysmac Studio の起動 2-2
2-1-2　新規プロジェクト作成 2-2
2-1-3　内蔵 EtherNet/IP ポート設定 2-3

2-1-4　コントローラ設定 2-3

2-2　DB 接続設定 2-5
2-2-1　DB 接続サービス設定 2-5
2-2-2　DB コネクション設定 2-7

第 3 章　DB 接続機能のプログラミング

3-1　DB にアクセスするまでの手順 3-2

3-2　構造体データ型の作成 3-3
3-2-1　概要 3-3
3-2-2　DB にアクセスするための構造体データ型の仕様 3-3
3-2-3　DB にアクセスするための構造体データ型の作成方法 3-11

3-3　DB マップ変数の作成 3-15
3-3-1　DB マップ変数、および DB へのマッピング 3-15
3-3-2　DB マップ変数の登録および属性 3-16
3-3-3　DB マップ変数の制約事項 3-17

3-4　DB マップ変数の DB へのマッピング 3-18
3-4-1　DB マップ生成命令による DB へのマッピング 3-18
3-4-2　DB マップ変数のマッピングの解除 3-18
3-4-3　DB マッピングの制約 3-18

3-5　DB 接続命令のプログラミング／転送 3-21
3-5-1　DB 接続サービスのプログラミングの概要 3-21
3-5-2　Sysmac Studio での DB 接続命令の表示 3-21
3-5-3　DB 接続命令一覧 3-22
3-5-4　システム定義変数 3-23
3-5-5　DB 接続命令のシミュレーションデバッグ 3-24
3-5-6　DB 接続設定とユーザプログラムの転送 3-24

3-6　設計、立ち上げ、運用時の各デバッグ 3-26
3-6-1　設計時 3-26
3-6-2　立ち上げ時 3-26
3-6-3　運用時 3-26

第 4 章　DB 接続サービスの開始／コネクションの接続と状態確認

4-1　DB 接続サービスの動作モードと開始／停止 4-2
4-1-1　DB 接続サービスの動作モード 4-2
4-1-2　DB 接続サービスを開始／停止する方法 4-2
4-1-3　DB 接続サービスの開始不可および停止の条件 4-4
4-1-4　DB 接続サービスの動作モードの変更 4-5

4-2　各 DB コネクションの接続／切断 4-6

4-3　DB 接続サービスおよび各コネクションの状態の確認 4-7
4-3-1　DB 接続サービスの稼働状態 4-7
4-3-2　DB 接続サービスの状態の確認 4-8
4-3-3　DB コネクションの接続状態 4-10
4-3-4　各 DB コネクションの状態の確認 4-11

第 5 章　各種の機能

5-1　各種機能の使用例 5-3

5-2　スプール機能 5-5
5-2-1　概要 5-5
5-2-2　スプール機能のしくみ 5-5
5-2-3　スプール機能の対象命令とスプール実行条件 5-5
5-2-4　スプール機能で使用するメモリ 5-6

5-2-5 スプール機能の設定 5-8
5-2-6 スプールに保留された SQL 文の再送方法 5-9
5-2-7 スプールに保留された SQL 文のクリア条件 5-10
5-2-8 DB 接続命令の動作との関係 5-11
5-2-9 スプールに保留可能な SQL 文数の目安の計算 5-13

5-3 ストアドプロシージャ呼出機能 5-15
5-3-1 概要 5-15
5-3-2 DB ごとのストアドプロシージャ呼出機能の仕様 5-16
5-3-3 ストアドプロシージャ呼出機能の実行方法 5-17
5-3-4 DB マップ変数の DB へのマッピング 5-18
5-3-5 ストアドプロシージャ呼出時の異常 5-20

5-4 バッチインサート機能 5-23
5-4-1 概要 5-23
5-4-2 バッチインサート機能の実行方法 5-23

5-5 DB 接続サービスシャットダウン機能 5-25
5-5-1 概要 5-25
5-5-2 シャットダウン機能のしくみ 5-25
5-5-3 シャットダウン機能の実行方法 5-26
5-5-4 DB 接続サービスのシャットダウンの確認方法 5-26

5-6 電源断発生時の SQL 文消失の回避策 5-28
5-6-1 概要 5-28
5-6-2 回避の方法 5-28

5-7 タイムアウト監視機能 5-32
5-7-1 タイムアウト監視機能の種類 5-32
5-7-2 ログインタイムアウト 5-33
5-7-3 クエリ実行タイムアウト 5-33
5-7-4 通信タイムアウト 5-33
5-7-5 命令実行タイムアウト 5-34
5-7-6 KeepAlive 監視時間 5-34

5-8 その他の機能 5-35
5-8-1 DB 接続サービスにおける、バックアップ／リストア機能 5-35
5-8-2 DB 接続サービスにおける、操作権限認証 5-36
5-8-3 暗号化通信 5-36

第 6 章 稼働ログの使用手順

6-1 稼働ログとは 6-2

6-2 実行ログ 6-3
6-2-1 概要 6-3
6-2-2 使用手順 6-3
6-2-3 実行ログの設定 6-3
6-2-4 実行ログの確認 6-4
6-2-5 実行ログファイルの仕様 6-4

6-3 デバッグログ 6-14
6-3-1 概要 6-14
6-3-2 使用手順 6-14
6-3-3 デバッグログの設定 6-14
6-3-4 デバッグログの開始 6-15
6-3-5 デバッグログの停止 6-16
6-3-6 デバッグログの確認 6-17
6-3-7 デバッグログファイルの仕様 6-17

6-4 SQL 実行失敗ログ 6-26
6-4-1 概要 6-26
6-4-2 使用手順 6-26
6-4-3 SQL 実行失敗ログの設定 6-26
6-4-4 SQL 実行失敗ログの確認 6-27
6-4-5 SQL 実行失敗ログファイルの仕様 6-27

6-5　SD メモリカードへの保存および SD メモリカードの交換 6-32
6-5-1　各稼働ログファイルの SD メモリカードへの保存条件 6-32
6-5-2　DB 接続サービスのディレクトリ構成 6-32
6-5-3　SD メモリカード交換時の各稼働ログ機能の動作 6-32
6-5-4　SD メモリカードの交換作業の目安時間 6-33
6-5-5　SD メモリカードの交換時期 6-34

6-6　稼働ログを確認する 6-35
6-6-1　稼働ログの確認方法 6-35
6-6-2　Sysmac Studio の稼働ログ表示による確認 6-35
6-6-3　SD メモリカード経由での確認 6-37
6-6-4　FTP クライアントソフトでの取り出し確認 6-37

第 7 章　DB 接続命令

DB 接続命令と変数 7-2
DB 接続命令の一覧 7-2
DB 接続命令の変数 7-2

DB_Connect（DB コネクション接続） 7-6
変数 7-6
関連するシステム定義変数 7-7
関連するデバイスパラメータ 7-7
関連するエラーコード一覧 7-7
機能説明 7-7
使用上の注意 7-8
サンプルプログラム 7-8

DB_Close（DB コネクション切断） 7-9
変数 7-9
関連するシステム定義変数 7-9
関連するデバイスパラメータ 7-10
関連するエラーコード一覧 7-10
機能説明 7-10
使用上の注意 7-10
サンプルプログラム 7-10

DB_CreateMapping（DB マップ生成） 7-11
変数 7-11
関連するシステム定義変数 7-12
関連するデバイスパラメータ 7-12
関連するエラーコード一覧 7-12
機能説明 7-13
使用上の注意 7-13
サンプルプログラム 7-14

DB_Insert（DB レコード挿入） 7-15
変数 7-15
関連するシステム定義変数 7-16
関連するデバイスパラメータ 7-16
関連するエラーコード一覧 7-16
機能説明 7-17
使用上の注意 7-17
サンプルプログラム 7-18

DB_Update（DB レコード更新） 7-19
変数 7-19
関連するシステム定義変数 7-20
関連するデバイスパラメータ 7-20
関連するエラーコード一覧 7-20
機能説明 7-21
使用上の注意 7-22
サンプルプログラム 7-23

DB_Select（DB レコード取得） 7-35
変数 7-35

関連するシステム定義変数 7-36
関連するデバイスパラメータ 7-36
関連するエラーコード一覧 7-36
機能説明 7-37
使用上の注意 7-38
サンプルプログラム 7-39

DB_Delete（DB レコード削除） 7-40
変数 7-40
関連するシステム定義変数 7-41
関連するデバイスパラメータ 7-41
関連するエラーコード一覧 7-41
機能説明 7-42
使用上の注意 7-42
サンプルプログラム 7-43

DB_ControlService（DB 接続サービスの制御） 7-54
変数 7-54
関連するシステム定義変数 7-55
関連するデバイスパラメータ 7-55
関連するエラーコード一覧 7-55
機能説明 7-55
使用上の注意 7-56
サンプルプログラム 7-56

DB_GetServiceStatus（DB 接続サービスの状態取得） 7-60
変数 7-60
関連するデバイスパラメータ 7-60
関連するエラーコード一覧 7-61
機能説明 7-61
使用上の注意 7-61
サンプルプログラム 7-61

DB_GetConnectionStatus（DB コネクションの状態取得） 7-65
変数 7-65
関連するシステム定義変数 7-66
関連するデバイスパラメータ 7-66
関連するエラーコード一覧 7-66
機能説明 7-66
使用上の注意 7-67
サンプルプログラム 7-67

DB_ControlSpool（スプールの再送・クリア） 7-71
変数 7-71
関連するシステム定義変数 7-72
関連するデバイスパラメータ 7-72
関連するエラーコード一覧 7-72
機能説明 7-72
使用上の注意 7-72
サンプルプログラム 7-73

DB_PutLog（稼働ログ記録） 7-78
変数 7-78
関連するエラーコード一覧 7-79
機能説明 7-79
使用上の注意 7-79
サンプルプログラム 7-80

DB_Shutdown（DB 接続サービスのシャットダウン） 7-83
変数 7-83
関連するシステム定義変数 7-83
関連するデバイスパラメータ 7-84
関連するエラーコード一覧 7-84
機能説明 7-84
使用上の注意 7-84
サンプルプログラム 7-84

DB_BatchInsert（DB レコード一括挿入） 7-87

変数 7-87
関連するシステム定義変数 7-88
関連するデバイスパラメータ 7-88
関連するエラーコード一覧 7-88
機能説明 7-89
使用上の注意 7-89
サンプルプログラム 7-90

DB_AttachProcedure（DB ストアドプロシージャ ハンドル生成） 7-99
変数 7-99
関連するシステム定義変数 7-100
関連するデバイスパラメータ 7-100
関連するエラーコード一覧 7-100
機能説明 7-101
使用上の注意 7-102
サンプルプログラム 7-102

DB_ExecuteProcedure（DB ストアドプロシージャ 実行） 7-103
変数 7-103
関連するシステム定義変数 7-104
関連するデバイスパラメータ 7-104
関連するエラーコード一覧 7-104
機能説明 7-105
使用上の注意 7-105
サンプルプログラム 7-106

DB_DetachProcedure（DB ストアドプロシージャ ハンドル解放） 7-115
変数 7-115
関連するシステム定義変数 7-115
関連するデバイスパラメータ 7-116
関連するエラーコード一覧 7-116
機能説明 7-116
使用上の注意 7-116
サンプルプログラム 7-117

第 8 章　異常の確認と処置

8-1　異常の概要 8-2
8-1-1　異常の確認方法 8-2
8-1-2　DB 接続サービスに関する異常 8-5

8-2　トラブルシューティング 8-7
8-2-1　異常一覧 8-7
8-2-2　異常詳細 8-14

付録

A-1　タスク設計方法 A-2
A-1-1　DB 接続サービスの起動時間 A-2
A-1-2　DB 接続命令の実行時間の参考値 A-4
A-1-3　DB 接続命令の実行時間の測定方法 A-12
A-1-4　システムサービス実行時間割合の目安 A-12
A-1-5　システムサービス実行時間割合の確認 A-15

A-2　DB 接続命令の実行時間 A-17
A-2-1　DB 接続命令の実行時間の制約 A-17
A-2-2　稼働ログ記録の DB 接続命令実行時間への影響 A-22
A-2-3　DB 応答時間の測定方法 A-23
A-2-4　命令実行タイムアウト監視による装置性能（タクト時間）の確保 A-24

A-3　仕様 A-25
A-3-1　一般仕様 A-25
A-3-2　性能仕様 A-25
A-3-3　機能仕様 A-25

A-4　バージョン関連情報 A-26
A-4-1　ユニットバージョンと DB 接続サービスのバージョン A-26
A-4-2　ユニットバージョンによる DB 接続機能の追加と変更 A-27
A-4-3　ユニットバージョン／DB 接続サービスのバージョンと Sysmac Studio の設定ユニットバージョン A-28
A-4-4　DB 接続サービスのバージョンとデバイス変更後の接続データベース種別 A-31
A-4-5　DB 接続サービスのバージョンと接続データベースの種別およびバージョン A-31

索引

ご承諾事項

「当社商品」について特別の合意がない場合には、お客様のご購入先にかかわらず、本ご承諾事項記載の条件を適用いたします。

定義

本ご承諾事項中の用語の定義は次のとおりです。

- 「当社商品」:「当社」の FA システム機器、汎用制御機器、センシング機器、電子・機構部品
- 「カタログ等」:「当社商品」に関する、ベスト制御機器オムロン、電子・機構部品総合カタログ、その他のカタログ、仕様書、取扱説明書、マニュアル等であって電磁的方法で提供されるものも含みます。
- 「利用条件等」:「カタログ等」に記載の、「当社商品」の利用条件、定格、性能、動作環境、取り扱い方法、利用上の注意、禁止事項その他
- 「お客様用途」:「当社商品」のお客様におけるご利用方法であって、お客様が製造する部品、電子基板、機器、設備またはシステム等への「当社商品」の組み込み、または利用を含みます。
- 「適合性等」:「お客様用途」での「当社商品」の(a)適合性、(b)動作、(c)第三者の知的財産の非侵害、(d)法令の遵守および(e)各種規格の遵守

記載事項のご注意

「カタログ等」の記載内容については次の点をご理解ください。

- 定格値および性能値は、単独試験における各条件のもとで得られた値であり、各定格値および性能値の複合条件のもとで得られる値を保証するものではありません。
- 参考データはご参考として提供するもので、その範囲で常に正常に動作することを保証するものではありません。
- 利用事例はご参考ですので、「当社」は「適合性等」について保証いたしかねます。
- 「当社」は、改善や当社都合等により、「当社商品」の生産を中止し、または「当社商品」の仕様を変更することがあります。

ご利用にあたってのご注意

ご採用およびご利用に際しては次の点をご理解ください。

- 定格・性能ほか「利用条件等」を遵守しご利用ください。
- お客様ご自身にて「適合性等」をご確認いただき、「当社商品」のご利用の可否をご判断ください。「当社」は「適合性等」を一切保証いたしかねます。
- 「当社商品」がお客様のシステム全体の中で意図した用途に対して、適切に配電・設置されていることをお客様ご自身で、必ず事前に確認してください。
- 「当社商品」をご使用の際には、(i)定格および性能に対し余裕のある「当社商品」のご利用、冗長設計などの安全設計、(ii)「当社商品」が故障しても、「お客様用途」の危険を最小にする安全設計、(iii)利用者に危険を知らせるための、安全対策のシステム全体としての構築、(iv)「当社商品」および「お客様用途」の定期的な保守、の各事項を実施してください。
- 「当社」は DDoS 攻撃（分散型 DoS 攻撃）、コンピュータウイルスその他の技術的な有害プログラム、不正アクセスにより、「当社商品」、インストールされたソフトウェア、またはすべてのコンピュータ機器、コンピュータプログラム、ネットワーク、データベースが感染したとしても、そのことにより直接または間接的に生じた損失、損害その他の費用について一切責任を負わないものとします。

お客様ご自身にて、①アンチウイルス保護、②データ入出力、③紛失データの復元、④「当社商品」またはインストールされたソフトウェアに対するコンピュータウイルス感染防止、⑤「当社商品」に対する不正アクセス防止についての十分な措置を講じてください。

- 「当社商品」は、一般工業製品向けの汎用品として設計製造されています。従いまして、次に掲げる用途での使用は意図しておらず、お客様が「当社商品」をこれらの用途に使用される際には、「当社」は「当社商品」に対して一切保証をいたしません。ただし、次に掲げる用途であっても「当社」の意図した特別な商品用途の場合や特別の合意がある場合は除きます。
  a）高い安全性が必要とされる用途（例：原子力制御設備、燃焼設備、航空・宇宙設備、鉄道設備、昇降設備、娯楽設備、医用機器、安全装置、その他生命・身体に危険が及びうる用途）
  b）高い信頼性が必要な用途（例：ガス・水道・電気等の供給システム、24 時間連続運転システム、決済システムほか権利・財産を取扱う用途など）
  c）厳しい条件または環境での用途（例：屋外に設置する設備、化学的汚染を被る設備、電磁的妨害を被る設備、振動・衝撃を受ける設備など）
  d）「カタログ等」に記載のない条件や環境での用途
- 上記の(a)から(d)に記載されている他、「本カタログ等記載の商品」は自動車（二輪車含む。以下同じ）向けではありません。自動車に搭載する用途には利用しないで下さい。自動車搭載用商品については当社営業担当者にご相談ください。

保証条件

「当社商品」の保証条件は次のとおりです。

- 保証期間　ご購入後 1 年間といたします。
  （ただし「カタログ等」に別途記載がある場合を除きます。）
- 保証内容　故障した「当社商品」について、以下のいずれかを「当社」の任意の判断で実施します。
  a）当社保守サービス拠点における故障した「当社商品」の無償修理
  （ただし、電子・機構部品については、修理対応は行いません。）
  b）故障した「当社商品」と同数の代替品の無償提供
- 保証対象外　故障の原因が次のいずれかに該当する場合は、保証いたしません。
  a）「当社商品」本来の使い方以外のご利用
  b）「利用条件等」から外れたご利用
  c）本ご承諾事項「ご利用にあたってのご注意」に反するご利用
  d）「当社」以外による改造、修理による場合
  e）「当社」以外の者によるソフトウェアプログラムによる場合
  f）「当社」からの出荷時の科学・技術の水準では予見できなかった原因
  g）上記のほか「当社」または「当社商品」以外の原因（天災等の不可抗力を含む）

責任の制限

本ご承諾事項に記載の保証が、「当社商品」に関する保証のすべてです。
「当社商品」に関連して生じた損害について、「当社」および「当社商品」の販売店は責任を負いません。

輸出管理

「当社商品」または技術資料を、輸出または非居住者に提供する場合は、安全保障貿易管理に関する日本および関係各国の法令・規制を遵守ください。お客様が法令・規則に違反する場合には、「当社商品」または技術資料をご提供できない場合があります。

安全上の注意

安全上のご注意の内容については、以下のマニュアルを参照してください。
- 『NX シリーズ CPU ユニット ユーザーズマニュアル ハードウェア編 （SBCA-418)』
- 『NX シリーズ 形 NX502 CPU ユニット ユーザーズマニュアル ハードウェア編 (SBCA-497)』
- 『NX シリーズ 形 NX102 CPU ユニット ユーザーズマニュアル ハードウェア編 (SBCA-462)』
- 『NJ シリーズ CPU ユニット ユーザーズマニュアル ハードウェア編 (SBCA-466)』
- 『NJ シリーズ ロボット統合 CPU ユニット ユーザーズマニュアル (SBCE-436)』

形 NJ501-4320 の安全上のご注意の内容については、当社営業担当者までご相談いただき、製品仕様書等による確認をお願いします。

安全上の要点

安全上の要点の内容については、以下のマニュアルを参照してください。
- 『NX シリーズ CPU ユニット ユーザーズマニュアル ハードウェア編 （SBCA-418)』
- 『NX シリーズ 形 NX502 CPU ユニット ユーザーズマニュアル ハードウェア編 (SBCA-497)』
- 『NX シリーズ 形 NX102 CPU ユニット ユーザーズマニュアル ハードウェア編 (SBCA-462)』
- 『NJ シリーズ CPU ユニット ユーザーズマニュアル ハードウェア編 (SBCA-466)』
- 『NJ シリーズ ロボット統合 CPU ユニット ユーザーズマニュアル (SBCE-436)』

形 NJ501-4320 の安全上の要点の内容については、当社営業担当者までご相談いただき、製品仕様書等による確認をお願いします。

使用上の注意

DB 接続サービスについての、使用上の注意を記載します。
下記以外の使用上の注意の内容については、以下のマニュアルを参照してください。

- 『NX シリーズ CPU ユニット ユーザーズマニュアル ハードウェア編 （SBCA-418)』
- 『NX シリーズ 形 NX502 CPU ユニット ユーザーズマニュアル ハードウェア編 (SBCA-497)』
- 『NX シリーズ 形 NX102 CPU ユニット ユーザーズマニュアル ハードウェア編 (SBCA-462)』
- 『NJ シリーズ CPU ユニット ユーザーズマニュアル ハードウェア編 (SBCA-466)』
- 『NJ シリーズ ロボット統合 CPU ユニット ユーザーズマニュアル (SBCE-436)』

形 NJ501-4320 の使用上の注意の内容については、当社営業担当者までご相談いただき、製品仕様書等による確認をお願いします。

- NJ シリーズの場合、DB 接続サービスでスプール機能を使用するときは、形式に応じて以下の EM バンクを使用します。DB 接続サービスが使用する EM バンクを DB 接続サービス以外の用途で使用すると、EM バンク内のスプールデータを書き換えます。DB 接続サービスが使用する EM バンクを DB 接続サービス以外の処理で使用しないでください。
  NJ501-□□20：EM バンク No.9～18（E9_00000～E18_32767）
  NJ101-□□20：EM バンク No.1～3（E1_00000～E3_32767）

- ストアドプロシージャ呼出機能の実行の際には、実行するストアドプロシージャの名称、処理内容および引数の値の確認を行った上で実行してください。

法規と規格

法規と規格の内容については、以下のマニュアルを参照してください。
- 『NX シリーズ CPU ユニット ユーザーズマニュアル ハードウェア編（SBCA-418)』
- 『NX シリーズ 形 NX502 CPU ユニット ユーザーズマニュアル ハードウェア編 (SBCA-497)』
- 『NX シリーズ 形 NX102 CPU ユニット ユーザーズマニュアル ハードウェア編（SBCA-462)』
- 『NJ シリーズ CPU ユニット ユーザーズマニュアル ハードウェア編（SBCA-466)』

バージョン

NJ/NX シリーズの各ユニットおよび各 EtherCAT スレーブは、ハードウェアやソフトウェアが、ハードウェアリビジョンやユニットバージョンという改変番号でバージョン管理されています。ハードウェアまたはソフトウェアに仕様変更があるごとに、ハードウェアリビジョンまたはユニットバージョンが更新されます。したがって、同じ形式のユニットや EtherCAT スレーブでも、ハードウェアリビジョンまたはユニットバージョンが異なると搭載機能や性能が異なります。

バージョンの種類

バージョンには、ユニットバージョンと、DB 接続サービスのバージョンの 2 種類があります。これらは独立に管理されているため、どちらか一方だけが更新される場合があります。

- **ユニットバージョン**

  ユニットや EtherCAT スレーブのハードウェアおよびソフトウェアの改変番号です。ハードウェアおよびソフトウェアに仕様変更があるごとに、バージョンが更新されます。したがって、同じ形式のユニットや EtherCAT スレーブでも、ユニットバージョンが異なると搭載機能や性能が異なります。

- **DB 接続サービスのバージョン**

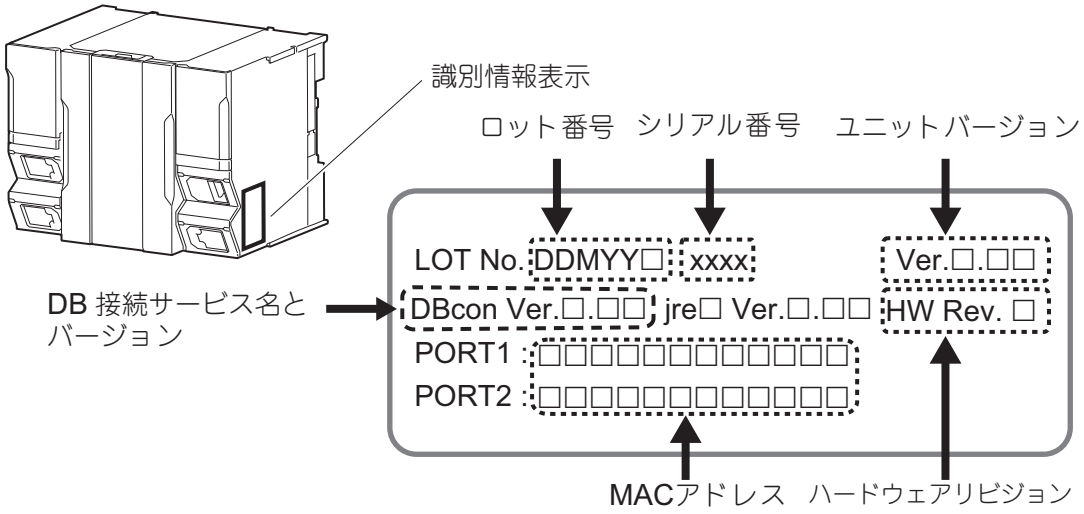
  DB 接続 CPU ユニットが搭載している、DB 接続サービスの改変番号です。DB 接続サービスに仕様変更があるごとに、バージョンが更新されます。

バージョンの確認方法

バージョンは、識別情報表示、または Sysmac Studio で確認できます。

識別情報表示による確認

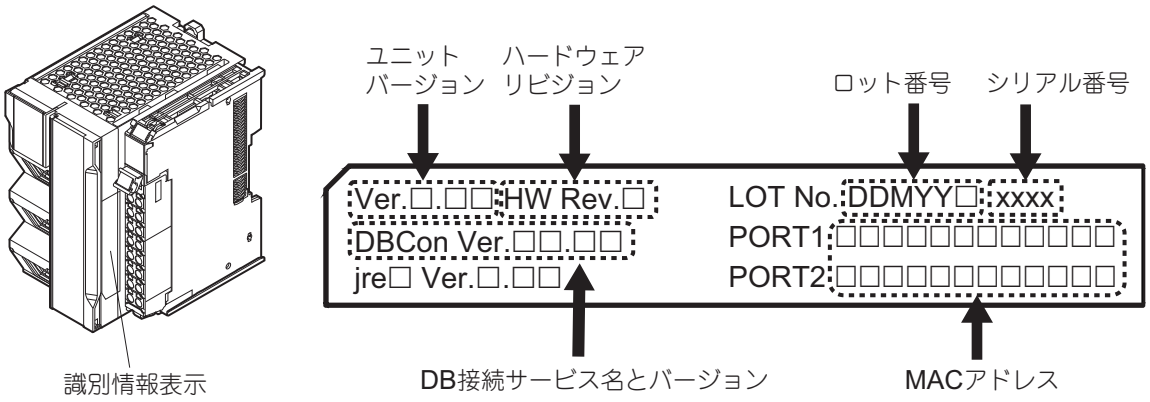
製品側面の識別情報表示で、バージョンが確認できます。
NX シリーズ CPU ユニット形 NX701-□□20 の識別情報表示は、下図のようになります。



（注）ハードウェアリビジョンが「なし」のユニットでは、ハードウェアリビジョンの表示はありません。

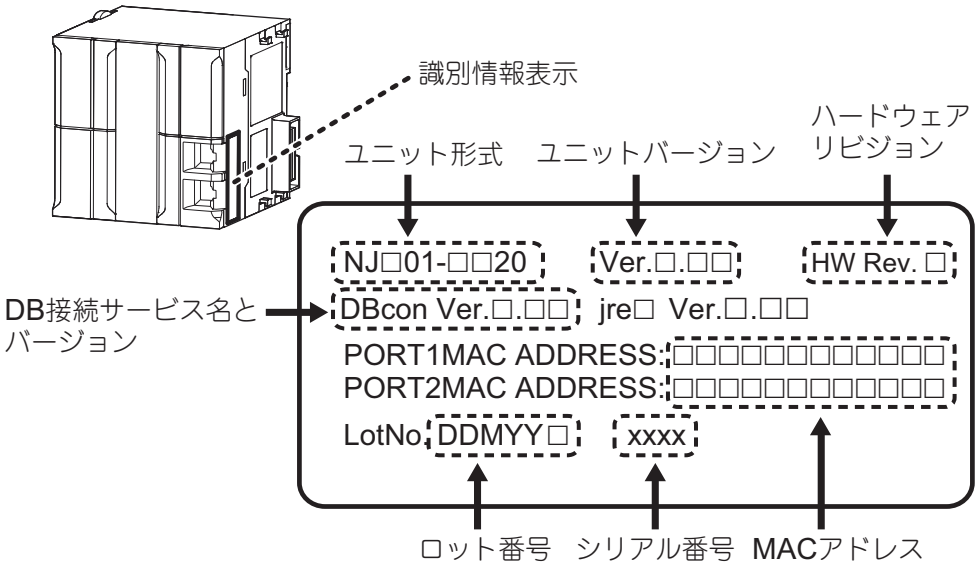
NX シリーズ CPU ユニット形 NX502-1□00 の識別情報表示は、『NX シリーズ 形 NX502 CPU ユニット ユーザーズマニュアル ハードウェア編 (SBCA-497)』を参照してください。

NX シリーズ CPU ユニット形 NX102-□□20 の識別情報表示は、下図のようになります。



（注）ハードウェアリビジョンが「なし」のユニットでは、ハードウェアリビジョンの表示はありません。

NJ シリーズ CPU ユニット形 NJ□01-□□20 の識別情報表示は、下図のようになります。



（注）ハードウェアリビジョンが「なし」のユニットでは、ハードウェアリビジョンの表示はありません。

Sysmac Studio による確認方法

Sysmac Studio で、バージョンが確認できます。ユニットと EtherCAT スレーブでは、確認方法が異なります。

● NX シリーズ CPU ユニットでのバージョン確認方法

ユニットのバージョンは、オンライン状態の［生産情報］で確認できます。バージョンが確認できるユニットは、以下のとおりです。

形式	バージョンを確認できるユニット
形 NX701-□□□□	CPU ユニット
形 NX502-□□□□	CPU ユニット、CPU ラック上の NX ユニット、CPU ラック上の X バスユニット
形 NX102-□□□□	CPU ユニット、CPU ラック上の NX ユニット

1 **マルチビューエクスプローラで［構成・設定］–［CPU・増設ラック］–［CPU ラック］を右クリックし、［生産情報表示］を選択します。**
［生産情報］ダイアログボックスが表示されます。

● NJ シリーズ CPU ユニットでのバージョン確認方法

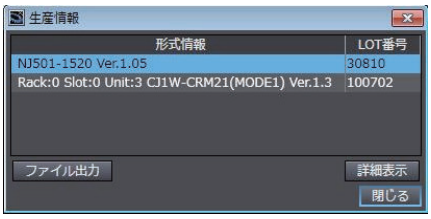
ユニットのバージョンは、オンライン状態の［生産情報］で確認できます。ただし、バージョンが確認できるユニットは、CPU ユニット、CJ シリーズの高機能 I/O ユニット、および CJ シリーズの CPU 高機能ユニットだけです。CJ ユニットの基本 I/O ユニットのバージョンは、Sysmac Studio では確認できません。

1 **マルチビューエクスプローラで［構成・設定］–［CPU・増設ラック］をダブルクリックします。または、［構成・設定］–［CPU・増設ラック］を右クリックして［編集］を選択します。**
ユニットエディタが表示されます。

2 **ユニットエディタの空欄上で右クリックして［生産情報表示］を選択します。**
［生産情報］ダイアログボックスが表示されます。

● 生産情報の表示内容の切り替え

1 **［生産情報］ダイアログボックスの右下の、［簡易表示］または［詳細表示］を選択します。**
［生産情報］の簡易表示と詳細表示が、切り替わります。



簡易表示



詳細表示

表示内容は、簡易表示と詳細表示とで異なります。詳細表示では、ユニットバージョンと DB 接続サービスのバージョンの両方が表示されます。簡易表示では、ユニットバージョンだけが表示されます。

（注）ハードウェアリビジョンは、ハードウェアバージョンの右端に「/」で区切られて表示されます。ハードウェアリビジョンが「なし」のユニットでは、ハードウェアリビジョンの表示はありません。

CPU ユニットのユニットバージョンと Sysmac Studio のバージョン

NJ/NX シリーズ CPU ユニットのユニットバージョンにより、搭載している機能が異なります。バージョンアップで追加された機能を使用するときは、対応しているバージョンの Sysmac Studio を使う必要があります。
NJ/NX シリーズ　DB 接続 CPU ユニットのユニットバージョンの種類と Sysmac Studio のバージョンとの関係、およびユニットバージョンによるサポート機能一覧については、「A-4 バージョン関連情報 (P.A-26)」を参照してください。

関連マニュアル

本マニュアルに関連するマニュアルは、下表のとおりです。併せてご覧ください。

マニュアル名称	Man.No.	形式	用途	内容
NX シリーズ CPU ユニット ユーザーズマニュアル ハードウェア編	SBCA-418	形 NX701-□□□□	形 NX701 CPU ユニットの概要／設計／取付／保守などの基本的な仕様について知りたいとき。おもにハードウェアに関する情報。	形 NX701 のシステム全体概要、および CPU ユニットに関して、以下の内容を説明します。 • 特長やシステム構成 • 概要 • 各部の名称と機能 • 一般仕様 • 設置と配線 • 保守点検
NX シリーズ 形 NX502 CPU ユニット ユーザーズマニュアル ハードウェア編	SBCA-497	形 NX502-□□□□	形 NX502 CPU ユニットの概要／設計／取付／保守などの基本的な仕様について知りたいとき。おもにハードウェアに関する情報。	形 NX502 のシステム全体概要、および CPU ユニットに関して、以下の内容を説明します。 • 特長やシステム構成 • 概要 • 各部の名称と機能 • 一般仕様 • 設置と配線 • 保守点検
NX シリーズ 形 NX102 CPU ユニット ユーザーズマニュアル ハードウェア編	SBCA-462	形 NX102-□□□□	形 NX102 CPU ユニットの概要／設計／取付／保守などの基本的な仕様について知りたいとき。おもにハードウェアに関する情報。	形 NX102 のシステム全体概要、および CPU ユニットに関して、以下の内容を説明します。 • 特長やシステム構成 • 概要 • 各部の名称と機能 • 一般仕様 • 設置と配線 • 保守点検
NJ シリーズ CPU ユニット ユーザーズマニュアル ハードウェア編	SBCA-466	形 NJ501-□□□□ 形 NJ301-□□□□ 形 NJ101-□□□□	NJ シリーズ CPU ユニットの概要／設計／取付／保守などの基本的な仕様について知りたいとき。おもにハードウェアに関する情報。	NJ シリーズのシステム全体概要、および CPU ユニットに関して、以下の内容を説明します。 • 特長やシステム構成 • 概要 • 各部の名称と機能 • 一般仕様 • 設置と配線 • 保守点検
NJ/NX シリーズ CPU ユニット ユーザーズマニュアル ソフトウェア編	SBCA-467	形 NX701-□□□□ 形 NX502-□□□□ 形 NX102-□□□□ 形 NX1P2-□□□□ 形 NJ501-□□□□ 形 NJ301-□□□□ 形 NJ101-□□□□	NJ/NX シリーズ CPU ユニットのプログラミング／システムの立ち上げについて知りたいとき。おもにソフトウェアに関する情報。	NJ/NX シリーズ CPU ユニットに関して、以下の内容を説明します。 • CPU ユニットの動作 • CPU ユニットの機能 • 初期設定 • IEC 61131-3 ベースの言語仕様とプログラミング
NJ/NX シリーズ コマンドリファレンス マニュアル 基本編	SBCA-468	形 NX701-□□□□ 形 NX502-□□□□ 形 NX102-□□□□ 形 NX1P2-□□□□ 形 NJ501-□□□□ 形 NJ301-□□□□ 形 NJ101-□□□□	NJ/NX シリーズの基本命令仕様の詳細について知りたいとき。	各命令（IEC 61131-3 仕様）の詳細を説明します。

マニュアル名称	Man.No.	形式	用途	内容
NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編	SBCE-433	形 NX701-□□□□ 形 NX502-□□□□ 形 NX102-□□□□ 形 NX1P2-□□□□ 形 NJ501-□□□□ 形 NJ301-□□□□ 形 NJ101-□□□□	モーション制御の設定やプログラミングの考え方について知りたいとき。	モーション制御のための CPU ユニットの設定や動作、プログラミングの考え方について説明します。
NJ/NX シリーズ コマンドリファレンス マニュアル モーション編	SBCE-434	形 NX701-□□□□ 形 NX502-□□□□ 形 NX102-□□□□ 形 NX1P2-□□□□ 形 NJ501-□□□□ 形 NJ301-□□□□ 形 NJ101-□□□□	モーション命令仕様の詳細について知りたいとき。	各モーション命令の詳細を説明します。
NJ/NX シリーズ CPU ユニット 内蔵 EtherCAT®ポート ユーザーズマニュアル	SBCD-376	形 NX701-□□□□ 形 NX502-□□□□ 形 NX102-□□□□ 形 NX1P2-□□□□ 形 NJ501-□□□□ 形 NJ301-□□□□ 形 NJ101-□□□□	NJ/NX シリーズ CPU ユニットの内蔵 EtherCAT ポートを使用するとき。	内蔵 EtherCAT ポートに関して説明します。 概要、構成、機能、セットアップについて記述しています。
NJ/NX シリーズ CPU ユニット 内蔵 EtherNet/IP™ポート ユーザーズマニュアル	SBCD-377	形 NX701-□□□□ 形 NX502-□□□□ 形 NX102-□□□□ 形 NX1P2-□□□□ 形 NJ501-□□□□ 形 NJ301-□□□□ 形 NJ101-□□□□	NJ/NX シリーズ CPU ユニットの内蔵 EtherNet/IP ポートを使用するとき。	内蔵 EtherNet/IP ポートに関して説明します。 基本設定、タグデータリンク、その他の機能について記述しています。
NJ/NX シリーズ データベース接続 CPU ユニット ユーザーズマニュアル	SBCA-411	形 NX701-□□20 形 NX502-□□□□ 形 NX102-□□20 形 NJ501-□□20 形 NJ101-□□20	NJ/NX シリーズでデータベース接続サービス機能を使用するとき。	データベース接続サービス機能について説明します。
NJ シリーズ ロボット統合 CPU ユニット ユーザーズマニュアル	SBCE-436	形 NJ501-R□□□	NJ シリーズ ロボット統合 CPU ユニットを使用するとき。	オムロン製ロボットを制御するための CPU ユニットの設定や動作、プログラミングの考え方について説明します。
Sysmac Studio ロボット統合システム構築機能 オペレーションマニュアル ロボット統合 CPU ユニット編	SBCA-464	形 SYSMAC-SE2□□□ 形 SYSMAC-SE200D-64	NJ シリーズ ロボット統合 CPU ユニットを使用してロボット統合システムを構築するとき。	Sysmac Studio で NJ シリーズ ロボット統合 CPU ユニットを使用するための機能の概要、操作方法について説明します。
NX シリーズ CPU ユニット ユーザーズマニュアル FINS 機能編	SBCD-375	形 NX701-□□20 形 NX502-□□□□ 形 NX102-□□□□	NX シリーズ CPU ユニットの FINS 機能を使用するとき。	NX シリーズ CPU ユニットの FINS 機能について説明します。
NJ/NX シリーズ トラブルシューティング マニュアル	SBCA-469	形 NX701-□□□□ 形 NX502-□□□□ 形 NX102-□□□□ 形 NX1P2-□□□□ 形 NJ501-□□□□ 形 NJ301-□□□□ 形 NJ101-□□□□	NJ/NX シリーズで検出する異常の詳細について知りたいとき。	NJ/NX シリーズ システムにて検出する異常管理の考え方と各異常項目について説明します。
Sysmac Studio Version 1 オペレーションマニュアル	SBCA-470	形 SYSMAC-SE2□□□	Sysmac Studio の操作方法、機能について知りたいとき。	Sysmac Studio の操作方法について説明します。

用語説明

用語	説明
CA	証明書を発行する機関。認証局。
DB	本書では、サーバ側のデータベースのことを指します。
DB コネクション	CPU ユニットと DB との間に確立された仮想的な通信路です。
DB 接続機能	CPU ユニット上で動作する、CPU ユニットと DB を接続する機能です。
DB 接続サービス	CPU ユニットから DB に接続するための DB 接続機能を提供するサービスです。 CPU ユニットの製品側面の識別情報表示および Sysmac Studio では DBCon と表記されます。
DB 接続サービスシャットダウン機能	稼働ログファイルを SD メモリカードに自動的に保存した上で、DB 接続サービスを終了する機能です。
DB 接続サービスの動作モード	DB 接続命令を実行したときに DB に対して実際にアクセスして動作するか、DB に対して実際にはアクセスせずに正常終了したものとして動作するかを切り替えるモードです。
DB 接続命令	DB 接続サービス用の命令の総称です。
DB にアクセスするための構造体データ型	DB のテーブルすべてのカラム、または一部のカラムを構造体メンバとして登録した構造体データ型です。
DB マップ変数	DB にアクセスするための構造体データ型をデータ型として使用した変数です。
DB マッピング	DB マップ変数の各メンバなどを、テーブルの各カラムやストアドプロシージャの引数、戻り値、結果セットに関連付けることです。
DB レコード一括挿入命令	DB_BatchInsert 命令を指します。
EM エリア	CJ ユニット用メモリの拡張データメモリを指します。電源断復帰（ON → OFF → ON）または動作モード変更（プログラムモード←→運転モード）時も、その電源断直前またはモード変更直前のデータが保持されます。
SQL	Structured Query Language の略です。データの書き込み、読み出しなどの DB の操作を行うための言語の一つです。
SQL 実行失敗ログ	稼働ログの一種で、DB 側での SQL 文の実行失敗に関するログです。
レコード処理	レコードの挿入や更新、取得、一括挿入などの、DB のレコードに対して操作を行う処理です。
SQL 種別	DB_CreateMapping 命令の入力変数の 1 つで、マッピングする変数のレコード処理の種別（レコードの挿入、更新、取得、一括挿入など）を指します。
SQL 文	データの書き込み、読み出しなどの DB の操作を行う、指示内容を示す文です。
暗号化通信	コントローラと DB の間の通信データを第三者が盗聴や改ざんすることを防止するために通信データを暗号化して送受信する手段です。
稼働ログ	DB 接続機能の CPU ユニット上の動作をトレースするためのログです。実行ログ、デバッグログ、SQL 実行失敗ログに大別されます。
カラム	DB の情報構造のひとつで、テーブルの列を表す語です。
サーバ証明書	DB を証明する X.509 電子証明書です。 証明書の秘密鍵と合わせて、DB が生成管理します。暗号化通信機能を使用する場合は、Sysmac Studio で CPU ユニットに登録しておくことが必要です。
実行ログ	稼働ログの一種で、DB 接続サービスの実行を記録するログです。
ストアドプロシージャ	DB に対する一連の処理を一つの手続きにまとめて、DB 管理システムに保存されたものです。 複雑な SQL 文の呼び出しを論理的に一つの処理単位にまとめて、簡単に名前で呼び出すことができます。
ストアドファンクション	一部の DB の種別では、戻り値を返すストアドプロシージャをストアドファンクションと呼んで、ストアドプロシージャと区別している場合があります。
ストアドプロシージャ呼出	ストアドプロシージャまたはストアドファンクションを呼び出すことや、呼び出す機能を指します。
スプール	スプール機能で使用する、DB への実行を蓄積するメモリです。
スプール機能	ネットワーク障害により実行できなかった DB へのレコード挿入、更新処理を、いくつか蓄積する機能です。
スプールデータ	スプールに保留された SQL 文を指します。

用語	説明
テーブル	DB の情報の構造のひとつです。データを格納するための入れ物です。
デバッグログ	稼働ログの一種で、どの SQL 文を実行したか、SQL 文ごとのパラメータ、実行結果を記録するログです。
バッチインサート	複数レコードを挿入する際、一括挿入する機能やコマンドを指します。

マニュアル改訂履歴

マニュアル改訂記号は、表表紙・裏表紙に記載されている Man.No.の後尾に付記されます。



改訂記号	改訂年月	改訂理由・改訂ページ
A	2013 年 4 月	初版
B	2013 年 8 月	• DB 接続命令のタイムアウト時間に関する記述の追加 • P.5-12、P.7-17、P.7-22、P.7-38、P.7-42 • 誤記修正
C	2014 年 2 月	DB 接続サービス Ver.1.01 以降のサポート機能に関する記述の追記
D	2014 年 7 月	• 形 NJ501-4320 の追加 • 誤記修正
E	2015 年 11 月	• 形 NJ101-□□20 の追加 • 誤記修正
F	2015 年 12 月	• DB 接続サービス Ver.1.02 以降のサポート機能に関する記述の追記 • 誤記修正
G	2016 年 6 月	EtherNet/IP ロゴの更新
H	2018 年 1 月	• 形 NX701-□□20 の追加 • DB 接続サービス Ver.1.03 以降のサポート機能に関する記述の追記
J	2018 年 6 月	• 形 NX102-□□20 の追加 • DB 接続サービス Ver.1.04 以降のサポート機能に関する記述の追記
K	2018 年 7 月	誤記修正
L	2019 年 7 月	• 形 NX701-□□20、および形 NX102-□□20 の DB 接続サービス Ver.2.00 以降のサポート機能に関する記述の追記 • 誤記修正
M	2019 年 11 月	誤記修正
N	2020 年 7 月	• 形 NJ501-□□20、および形 NJ101-□□20 の DB 接続サービス Ver.2.00 以降のサポート機能に関する記述の追記 • 誤記修正
P	2020 年 10 月	• 形 NX102-□□20 の Ver.1.35 以降のサポート機能に関する記述の追記 • 形 NX701-□□20、形 NJ501-1□20、形 NJ501-4320、および形 NJ101-□□20 の Ver.1.23 以降のサポート機能に関する記述の追記
R	2020 年 12 月	• 形 NJ501-R□20 の追加 • 誤記修正
S	2021 年 7 月	• 形 NX102-□□20 の Ver.1.37 以降のサポート機能に関する記述の追記 • 形 NJ501-□□20、形 NJ501-4320、形 NJ101-1□20、および形 NX701-□□20 の Ver.1.26 以降のサポート機能に関する記述の追記 • 形 NJ501-R□20 の Ver.1.46 以降のサポート機能に関する記述の追記 • SD メモリカードに関する記述の追記 • DB 接続サービス Ver.2.01 以降のサポート機能に関する記述の追記
T	2021 年 10 月	形 NX701-□□20 のハードウェアリビジョン A に関する記述の追記
U	2022 年 4 月	誤記修正
V	2022 年 6 月	イベントの追加に伴う改訂
W	2023 年 4 月	形 NX502-1□00 の追加に伴う改訂
X	2023 年 7 月	誤記修正

改訂記号	改訂年月	改訂理由・改訂ページ
Y	2024 年 1 月	• 形 NX502-1□00 の Ver.1.65 以降のサポート機能に関する記述の追記 • DB ベンダの延長サポート終了後の当社サポートについての記述を追記
Z	2024 年 10 月	誤記修正
AA	2025 年 2 月	誤記修正
AB	2025 年 4 月	形 NX701-□□20 および形 NX102-□□20 で、DB 接続サービス Ver.2.04 以降のサポート機能に関する記述の追記
AC	2025 年 7 月	形 NJ501-□□20 および形 NJ101-□□20 で、DB 接続サービス Ver.2.04 以降のサポート機能に関する記述の追記

1

DB 接続サービスの概要

この章では、DB 接続サービスの概要について説明しています。

1-1　概要と特長 1-2
1-1-1　概要 1-2
1-1-2　特長 1-3
1-2　DB 接続サービスの仕様としくみ 1-4
1-2-1　DB 接続サービスの仕様 1-4
1-2-2　DB 接続機能のしくみ 1-10
1-3　DB 接続サービスの使用手順 1-13

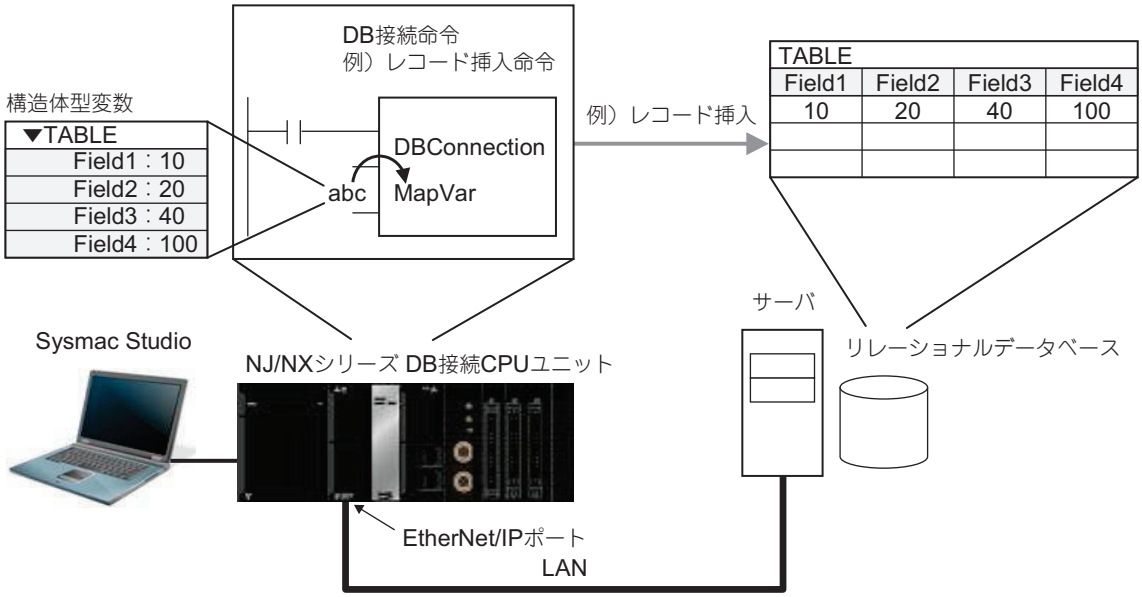
1-1 概要と特長

ここでは、DB 接続サービスの概要と特長について、説明しています。

1-1-1 概要

マシンオートメーションコントローラ NJ/NX シリーズは、機械制御に必要な機能・高速性能と、産業用コントローラとしての安全性・信頼性・保守性を両立した、次世代コントローラです。
従来のプログラマブルコントローラの機能を包含し、モーション制御に必要な各種機能を付加した統合型コントローラとして、セーフティ機器、視覚装置、モーション機器、I/O などの入出力機器を、高速な EtherCAT 上で同期して制御できます。
オムロンでは統一した通信仕様やユーザインターフェース仕様で設計されたオムロン制御機器を、Sysmac デバイスとしてご提供しています。マシンオートメーションコントローラ NJ/NX シリーズは、EtherCAT スレーブなどの Sysmac デバイス、およびオートメーションソフトウェア Sysmac Studio との組み合わせにおいて、最適な機能、操作性を実現できるように設計されています。
Sysmac デバイスにより構成されたシステムでは、機器どうしが容易につながり、統一した考え方とユーザビリティで操作できます。

DB 接続サービスは、NJ/NX シリーズ CPU ユニット上で専用命令（総称“DB 接続命令”）を実行することによって、CPU ユニットの内蔵 EtherNet/IP ポート、または CPU ユニットに接続された NX シリーズ EtherNet/IP ユニットの EtherNet/IP ポートに接続されているサーバ上のリレーショナルデータベース（以下 DB と表記する場合があります。）に対して、レコード挿入、更新、読み出し、削除を行う機能です。



- DB の種別は、Oracle 社製 Oracle Data base、Microsoft 社製 SQL Server、IBM 社製 DB2 for Linux, UNIX and Windows、Oracle 社製 MySQL、Firebird Foundation 製 Firebird、および PostgreSQL Global Development Group 製 PostgreSQL に対応しています。*1
- １つもしくは複数のサーバの中の、複数の DB*2 にアクセスできます。データや SQL 操作（INSERT/SELECT など）によってアクセスする DB を切り換えたり、サーバ側のパソコンのダウンなどにより、DB に接続できない場合に別サーバの DB に接続するなどの処理が実現できます。

*1. CPU ユニットの形式ごとに接続可能な DB が異なります。接続可能な DB については、「1-2-1 DB 接続サービスの仕様 (P.1-4)」を参照してください。
*2. 同時接続可能 DB 数については、「1-2-1 DB 接続サービスの仕様 (P.1-4)」を参照してください。

1-1-2　特長

専用のユニット、ツール、ミドルウェアが不要

- DB 接続機能のための専用のユニットは必要ありません。NJ/NX シリーズ CPU ユニットで対応します。
- DB 接続機能のための専用のツールは必要ありません。Sysmac Studio で対応します。
- サーバ側に NJ/NX シリーズへの接続専用のミドルウェアは不要です。

簡単な方法で DB にアクセス可能

- DB に対する INSERT、SELECT などの SQL 操作を簡単に実行できます。
- DB に対するストアドプロシージャ機能やバッチインサート機能などの高度な DB 機能も簡単に実行でき、より高速にデータ処理できます。*1
- SQL 文の知識が特になくても実行できます。
- DB 側のカラムに対応する構造体を作成することで、DB アクセス用の変数定義が可能です。
- 専用命令で SQL 操作を実行できるため、実行タイミングの制御、書き込む値の準備などが、容易です。
- 暗号化通信機能の使用により、よりセキュアな通信ができます。*1

*1. DB 接続サービス Ver.2.00 以降が対象です。

稼働ログが可能

- 専用命令の実行結果のログ、および処理（内部 SQL 文）の実行結果のログを、CPU ユニット装着の SD メモリカードにログファイルとして保存できます。それを Sysmac Studio または FTP クライアントから確認できます。*1*2

*1. ログファイル保存用に、形 NX502 CPU ユニットを除く DB 接続 CPU ユニットには SD メモリカードが同梱されています。SD メモリカード内のファイルを命令で読み書きするなど、DB 接続機能以外の用途にも使用できます。

*2. CPU ユニットに SD メモリカードを装着していない場合、稼働ログ（実行ログ、デバッグログ、SQL 実行失敗ログ）は使用できません。しかし、稼働ログを使用しない設定をしていれば、CPU ユニットに SD メモリカードを装着していなくても SD メモリカードに関するエラーは発生せず、DB 接続サービスを使用できます。

異常や電源断時でも安心の対応

- DB との情報交換異常が発生中に、送信できなかった処理（内部 SQL 文）がスプールに保留され、回復時に実行できます。
- CPU ユニットの電源 OFF 時に、DB 接続サービスシャットダウン機能を実行することで、稼働ログの内容が消失しないように DB 接続サービスを終了することができます。

DB アクセス機能のライブラリ化が可能

- 専用命令をユーザ定義のファンクションブロックに記述することで、ライブラリファイルとして提供および再利用ができます。

1-2 DB 接続サービスの仕様としくみ

ここでは、DB 接続サービスの仕様としくみについて示します。

1-2-1 DB 接続サービスの仕様

ここでは、DB 接続サービスの仕様について説明しています。データベース接続 CPU ユニットの一般仕様、性能仕様、機能仕様は、「A-3 仕様 (P.A-25)」を参照してください。
DB 接続サービスのバージョンアップに関する情報は、「A-4 バージョン関連情報 (P.A-26)」を参照してください。

NX シリーズ CPU ユニット

NX シリーズ CPU ユニットの、DB 接続サービスの仕様を示します。

仕様項目		CPU ユニット形式		
		形 NX701-1□20*1	形 NX502-1□00	形 NX102-□□20
接続可能な DB のバージョン*2	Microsoft 社製 SQL Server	2012, 2014, 2016, 2017, 2019, 2022	2014, 2016, 2017, 2019, 2022	2012, 2014, 2016, 2017, 2019, 2022
	Oracle 社製 Oracle Database	11g, 12c, 18c, 19c, 21c, 23ai(23c)	19c, 21c, 23ai(23c)	11g, 12c, 18c, 19c, 21c, 23ai(23c)
	IBM 社製 DB2 for Linux, UNIX and Windows	9.7, 10.1, 10.5, 11.1	非対応	9.7, 10.1, 10.5, 11.1
	Oracle 社製 MySQL Community Edition*3	5.6, 5.7, 8.0	8.0	5.6, 5.7, 8.0
	Firebird Foundation 製 Firebird	2.5	非対応	2.5
	PostgreSQL Global Development Group 製 PostgreSQL*4	9.4, 9.5, 9.6, 10, 11, 12, 13, 14, 15, 16	11, 12, 13, 14, 15, 16	9.4, 9.5, 9.6, 10, 11, 12, 13, 14, 15, 16
DB コネクション数（同時接続可能 DB 数）		3*5		2*5*6
命令	対応処理	NJ/NX シリーズ CPU ユニット内の DB 接続命令により、以下が可能。 レコード挿入（INSERT）、レコード更新（UPDATE）、レコード取得（SELECT）、レコード削除（DELETE）、ストアドプロシージャ実行*7、バッチインサート実行*7		
	命令同時起動最大数	32		
	INSERT 操作可能な最大カラム数	SQL Server と接続する場合：1024 Oracle と接続する場合：1000 DB2 と接続する場合：1000 MySQL と接続する場合：1000 Firebird と接続する場合：1000 PostgreSQL と接続する場合：1000	SQL Server と接続する場合：1024 Oracle と接続する場合：1000 DB2 と接続する場合：非対応 MySQL と接続する場合：1000 Firebird と接続する場合：非対応 PostgreSQL と接続する場合：1000	SQL Server と接続する場合：1024 Oracle と接続する場合：1000 DB2 と接続する場合：1000 MySQL と接続する場合：1000 Firebird と接続する場合：1000 PostgreSQL と接続する場合：1000

仕様項目		CPU ユニット形式		
		形 NX701-1□20*1	形 NX502-1□00	形 NX102-□□20
	UPDATE 操作可能な最大カラム数	SQL Server と接続する場合：1024 Oracle と接続する場合：1000 DB2 と接続する場合：1000 MySQL と接続する場合：1000 Firebird と接続する場合：1000 PostgreSQL と接続する場合：1000	SQL Server と接続する場合：1024 Oracle と接続する場合：1000 DB2 と接続する場合：非対応 MySQL と接続する場合：1000 Firebird と接続する場合：非対応 PostgreSQL と接続する場合：1000	SQL Server と接続する場合：1024 Oracle と接続する場合：1000 DB2 と接続する場合：1000 MySQL と接続する場合：1000 Firebird と接続する場合：1000 PostgreSQL と接続する場合：1000
	SELECT 操作可能な最大カラム数	SQL Server と接続する場合：1024 Oracle と接続する場合：1000 DB2 と接続する場合：1000 MySQL と接続する場合：1000 Firebird と接続する場合：1000 PostgreSQL と接続する場合：1000	SQL Server と接続する場合：1024 Oracle と接続する場合：1000 DB2 と接続する場合：非対応 MySQL と接続する場合：1000 Firebird と接続する場合：非対応 PostgreSQL と接続する場合：1000	SQL Server と接続する場合：1024 Oracle と接続する場合：1000 DB2 と接続する場合：1000 MySQL と接続する場合：1000 Firebird と接続する場合：1000 PostgreSQL と接続する場合：1000
	SELECT 操作した結果の最大レコード	65535 要素、4M バイト		
	ストアドプロシージャ呼出	引数：最大 256 変数 戻り値：1 変数 結果セット：対応 対応 DB：SQL Server, Oracle, MySQL, PostgreSQL スプール機能：未対応		
	バッチインサート実行	可能なデータサイズ：カラム 1000 未満かつ構造体の変数サイズの仕様上限以下 対応 DB：SQL Server, Oracle, MySQL, PostgreSQL スプール機能：未対応		
	マッピング可能な DB マップ変数の最大数	SQL Server と接続する場合：60 Oracle と接続する場合：30 DB2 と接続する場合：30 MySQL と接続する場合：30 Firebird と接続する場合：15 PostgreSQL と接続する場合：30 *8	SQL Server と接続する場合：60 Oracle と接続する場合：30 DB2 と接続する場合：非対応 MySQL と接続する場合：30 Firebird と接続する場合：非対応 PostgreSQL と接続する場合：30 *8	SQL Server と接続する場合：30 Oracle と接続する場合：20 DB2 と接続する場合：20 MySQL と接続する場合：20 Firebird と接続する場合：15 PostgreSQL と接続する場合：20 *6、*8
DB 接続サービスの動作モード		稼働モードまたはテストモード • 稼働モード： DB 接続命令を実行時、DB に対して実際にアクセスし動作する。 • テストモード：DB 接続命令を実行時、DB に対して実際にアクセスせず、正常終了したものとして動作する。		
スプール機能		障害発生時に SQL 文を保留し、復旧時に SQL 文を再送する機能。		
	スプール容量*9	2M バイト		192K バイト

仕様項目		CPU ユニット形式		
		形 NX701-1□20*1	形 NX502-1□00	形 NX102-□□20
稼働ログ機能		以下の 3 種類が可能。 • 実行ログ：DB 接続サービスの実行に関するログ。 • デバッグログ：DB 接続サービスの SQL 文実行に関する詳細ログ。 • SQL 実行失敗ログ：DB 側での SQL 文の実行失敗に関するログ。		
DB 接続サービスシャットダウン機能		稼働ログファイルを SD メモリカードに自動的に保存した上で、DB 接続サービスを終了する機能。		
通信ポート	2 ポート対応	• 内蔵 EtherNet/IP ポートを、2 ポートとも利用可能。 • コネクションごとにどちらのポートを使用するかは、IP アドレスの設定による。 • 二つのコネクションのそれぞれで、1 ポートずつ同時に使用することも可能。	• 内蔵 EtherNet/IP ポートを、2 ポートとも利用可能。 • コネクションごとにどちらのポートを使用するかは、IP アドレスの設定による。 • 二つのコネクションのそれぞれで、1 ポートずつ同時に使用することも可能。 • 内蔵 EtherNet/IP ポートの 2 ポートと、CPU ユニットに接続された NX シリーズ EtherNet/IP ユニットの EtherNet/IP ポート 1 ポートを合わせた 3 ポートで、1 ポートずつ同時に使用することも可能。	• 内蔵 EtherNet/IP ポートを、2 ポートとも利用可能。 • コネクションごとにどちらのポートを使用するかは、IP アドレスの設定による。 • 二つのコネクションのそれぞれで、1 ポートずつ同時に使用することも可能。*6
暗号化通信	対応 DB	SQL Server, Oracle, MySQL, PostgreSQL		
	TLS Ver.	TLS 1.2		

*1. 内蔵 EtherNet/IP ポートを使用した、CIP（Common Industrial Protocol）による通信は、以下の形式と同じ機能をサポートします。そのため、EtherNet/IP タグデータリンク機能を実行する場合、NetworkConfigurator 上で選択する形式は、以下となります。また、Sysmac Gateway または CX-Compolet で表示する形式は以下となります。

使用する CPU ユニット形式	対応する CPU ユニット形式
形 NX701-1720	形 NX701-1700
形 NX701-1620	形 NX701-1600

*2. DB を動作させる OS は、Windows Server OS および Windows クライアント OS を前提としています。動作要件の詳細は、DB ごとの動作要件を確認してください。また、クラウド上の DB への接続は、サポートしていません。なお、過去の DB 接続サービスのバージョンでサポートしていた接続可能な DB のバージョンについては、「A-4-5 DB 接続サービスのバージョンと接続データベースの種別およびバージョン (P.A-31)」を参照してください。

*3. 接続対象の DB のストレージエンジンは、InnoDB/MyISAM です。

*4. PostgreSQL と接続する場合、以下の設定を行い、サーバ側 DB のロケールを C に設定してください。設定を行わないとエラーメッセージが正しく表示されません。

<PostgreSQL インストールフォルダ>\data\postgresql.conf の lc_messages の値を C に変更し、PostgreSQL を再起動

lc_messages = 'C'

*5. 複数の DB コネクションに接続する場合、異なるデータベース種別に接続する設定にしたときの動作は保証できません。

*6. DB 接続サービス Ver.2.00 以降が対象です。DB 接続サービスのバージョンの詳細については、「DB 接続サービスのバージョンと Sysmac Studio での設定ユニットバージョンの関係 (P.A-29)」を参照してください。

*7. DB 接続サービス Ver.2.00 以降が対象です。DB 接続サービスのバージョンと命令の詳細については、「DB 接続命令の一覧 (P.7-2)」を参照してください。
*8. マッピング可能な DB マップ変数の最大数は、INSERT/UPDATE、ストアドプロシージャ、バッチインサートで使用される DB マップ変数の合計数です。また、最大個数に満たない場合でも、DB マップ変数のデータ型として使用する構造体定義のメンバ総数は最大 10000 個です。
*9. スプールに保留できる SQL 文数については、「5-2-9 スプールに保留可能な SQL 文数の目安の計算 (P.5-13)」を参照してください。

NJ シリーズ CPU ユニット

NJ シリーズ CPU ユニットの、DB 接続サービスの仕様を示します。

仕様項目		CPU ユニット形式		
		形 NJ501-□□20*1	形 NJ501-4320*1	形 NJ101-□020*1
接続可能な DB のバージョン *2	Microsoft 社製 SQL Server	2012, 2014, 2016, 2017, 2019, 2022		
	Oracle 社製 Oracle Database	11g, 12c, 18c, 19c, 21c, 23ai(23c)		
	IBM 社製 DB2 for Linux, UNIX and Windows	9.7, 10.1, 10.5, 11.1	非対応	9.7, 10.1, 10.5, 11.1
	Oracle 社製 MySQL Community Edition*3	5.6, 5.7, 8.0		
	Firebird Foundation 製 Firebird	2.5	非対応	2.5
	PostgreSQL Global Development Group 製 PostgreSQL*4	9.4, 9.5, 9.6, 10, 11, 12, 13, 14, 15, 16	非対応	9.4, 9.5, 9.6, 10, 11, 12, 13, 14, 15, 16
DB コネクション数（同時接続可能 DB 数）		3*5		1

仕様項目		CPU ユニット形式		
		形 NJ501-□□20[*1]	形 NJ501-4320[*1]	形 NJ101-□020[*1]
命令	対応処理	NJ/NX シリーズ CPU ユニット内の DB 接続命令により、以下が可能。 レコード挿入（INSERT）、レコード更新（UPDATE）、レコード取得（SELECT）、レコード削除（DELETE）、ストアドプロシージャ実行[*6]、バッチインサート実行[*6]		
	命令同時起動最大数	32		
	INSERT 操作可能な最大カラム数	SQL Server と接続する場合：1024 Oracle と接続する場合：1000 DB2 と接続する場合：1000 MySQL と接続する場合：1000 Firebird と接続する場合：1000 PostgreSQL と接続する場合：1000	SQL Server と接続する場合：1024 Oracle と接続する場合：1000 DB2 と接続する場合：非対応 MySQL と接続する場合：1000 Firebird と接続する場合：非対応 PostgreSQL と接続する場合：非対応	SQL Server と接続する場合：1024 Oracle と接続する場合：1000 DB2 と接続する場合：1000 MySQL と接続する場合：1000 Firebird と接続する場合：1000 PostgreSQL と接続する場合：1000
	UPDATE 操作可能な最大カラム数	SQL Server と接続する場合：1024 Oracle と接続する場合：1000 DB2 と接続する場合：1000 MySQL と接続する場合：1000 Firebird と接続する場合：1000 PostgreSQL と接続する場合：1000	SQL Server と接続する場合：1024 Oracle と接続する場合：1000 DB2 と接続する場合：非対応 MySQL と接続する場合：1000 Firebird と接続する場合：非対応 PostgreSQL と接続する場合：非対応	SQL Server と接続する場合：1024 Oracle と接続する場合：1000 DB2 と接続する場合：1000 MySQL と接続する場合：1000 Firebird と接続する場合：1000 PostgreSQL と接続する場合：1000
	SELECT 操作可能な最大カラム数	SQL Server と接続する場合：1024 Oracle と接続する場合：1000 DB2 と接続する場合：1000 MySQL と接続する場合：1000 Firebird と接続する場合：1000 PostgreSQL と接続する場合：1000	SQL Server と接続する場合：1024 Oracle と接続する場合：1000 DB2 と接続する場合：非対応 MySQL と接続する場合：1000 Firebird と接続する場合：非対応 PostgreSQL と接続する場合：非対応	SQL Server と接続する場合：1024 Oracle と接続する場合：1000 DB2 と接続する場合：1000 MySQL と接続する場合：1000 Firebird と接続する場合：1000 PostgreSQL と接続する場合：1000
	SELECT 操作した結果の最大レコード	65535 要素、4M バイト		65535 要素、2M バイト
	ストアドプロシージャ呼出	引数：最大 256 変数 戻り値：1 変数 結果セット：対応 対応 DB：SQL Server, Oracle, MySQL, PostgreSQL[*7] スプール機能：未対応		
	バッチインサート実行	可能なデータサイズ：カラム 1000 未満かつ構造体の変数サイズの仕様上限以下 対応 DB：SQL Server, Oracle, MySQL, PostgreSQL[*7] スプール機能：未対応		

仕様項目		CPU ユニット形式		
		形 NJ501-□□20*1	形 NJ501-4320*1	形 NJ101-□020*1
	マッピング可能な DB マップ変数の最大数 *8	SQL Server と接続する場合：60 Oracle と接続する場合：30 DB2 と接続する場合：30 MySQL と接続する場合：30 Firebird と接続する場合：15 PostgreSQL と接続する場合：30	SQL Server と接続する場合：60 Oracle と接続する場合：30 DB2 と接続する場合：非対応 MySQL と接続する場合：30 Firebird と接続する場合：非対応 PostgreSQL と接続する場合：非対応	SQL Server と接続する場合：15 Oracle と接続する場合：15 DB2 と接続する場合：15 MySQL と接続する場合：15 Firebird と接続する場合：15 PostgreSQL と接続する場合：15
DB 接続サービスの動作モード		稼働モードまたはテストモード • 稼働モード：DB 接続命令を実行時、DB に対して実際にアクセスし動作する。 • テストモード：DB 接続命令を実行時、DB に対して実際にアクセスせず、正常終了したものとして動作する。		
スプール機能		障害発生時に SQL 文を保留し、復旧時に SQL 文を再送する機能。		
	スプール容量*9	1M バイト		192K バイト
稼働ログ機能		以下の 3 種類が可能。 • 実行ログ：DB 接続サービスの実行に関するログ。 • デバッグログ：DB 接続サービスの SQL 文実行に関する詳細ログ。 • SQL 実行失敗ログ：DB 側での SQL 文の実行失敗に関するログ。		
DB 接続サービスシャットダウン機能		稼働ログファイルを SD メモリカードに自動的に保存した上で、DB 接続サービスを終了する機能。		
暗号化通信	対応 DB	SQL Server, Oracle, MySQL, PostgreSQL*7		
	TLS Ver.	TLS 1.2		

*1. 内蔵 EtherNet/IP ポートを使用した、CIP（Common Industrial Protocol）による通信は、以下の形式と同じ機能をサポートします。そのため、EtherNet/IP タグデータリンク機能を実行する場合、NetworkConfigurator 上で選択する形式は、以下となります。また、Sysmac Gateway または CX-Compolet で表示する形式は以下となります。

使用する CPU ユニット形式	対応する CPU ユニット形式
形 NJ501-1520	形 NJ501-1500
形 NJ501-1420	形 NJ501-1400
形 NJ501-1320	形 NJ501-1300
形 NJ501-R520	形 NJ501-R500
形 NJ501-R420	形 NJ501-R400
形 NJ501-R320	形 NJ501-R300
形 NJ501-4320	形 NJ501-4300
形 NJ101-□□20	形 NJ101

*2. DB を動作させる OS は、Windows Server OS および Windows クライアント OS を前提としています。動作要件の詳細は、各 DB の動作要件を確認してください。また、クラウド上の DB への接続は、サポートしていません。なお、過去の DB 接続サービスのバージョンでサポートしていた接続可能な DB のバージョンについては、「A-4-5 DB 接続サービスのバージョンと接続データベースの種別およびバージョン (P.A-31)」を参照してください。

*3. 接続対象の DB のストレージエンジンは、InnoDB/MyISAM です。

*4. PostgreSQL と接続する場合、以下の設定を行い、サーバ側 DB のロケールを C に設定してください。設定を行わないとエラーメッセージが正しく表示されません。
<PostgreSQL インストールフォルダ>\data\postgresql.conf の lc_messages の値を C に変更し、PostgreSQL を再起動
lc_messages = 'C'

*5. 複数の DB コネクションに接続する場合、異なるデータベース種別に接続する設定にしたときの動作は保証できません。

*6. DB 接続サービス Ver.2.00 以降が対象です。DB 接続サービスのバージョンと命令の詳細については、「DB 接続命令の一覧 (P.7-2)」を参照してください。
*7. 形 NJ501-4320 は、PostgreSQL をサポートしていません。
*8. マッピング可能な DB マップ変数の最大数は、INSERT/UPDATE、ストアドプロシージャ、バッチインサートで使用される DB マップ変数の合計数です。また、最大個数に満たない場合でも、DB マップ変数のデータ型として使用する構造体定義のメンバ総数は最大 10000 個です。
*9. スプールに保留できる SQL 文数については、「5-2-9 スプールに保留可能な SQL 文数の目安の計算 (P.5-13)」を参照してください。

延長サポートが終了している DB のバージョン

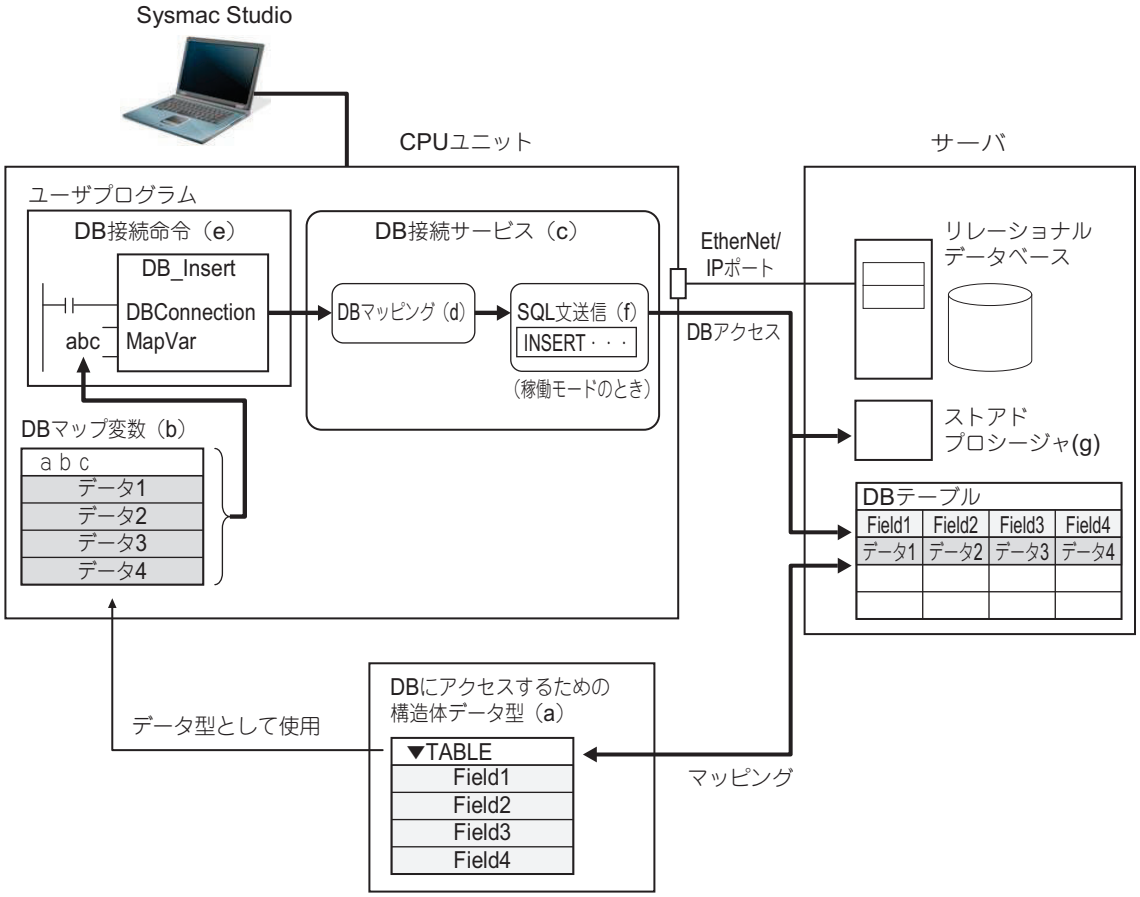
DB ベンダの延長サポートが終了した DB のバージョンについては、新しいバージョンへの置き換えをご検討ください。延長サポートが終了した DB のバージョンに対して、当社の DB 接続機能のサポートが継続できずサポート対象外となる可能性があります。

1-2-2 DB 接続機能のしくみ

以下に、DB 接続機能のしくみについて、基本的なしくみとその他のしくみに分けて、示します。
使用手順については、「1-3 DB 接続サービスの使用手順 (P.1-13)」を参照してください。

基本的なしくみ

DB 接続機能の基本的なしくみを、下図を使用して説明します。

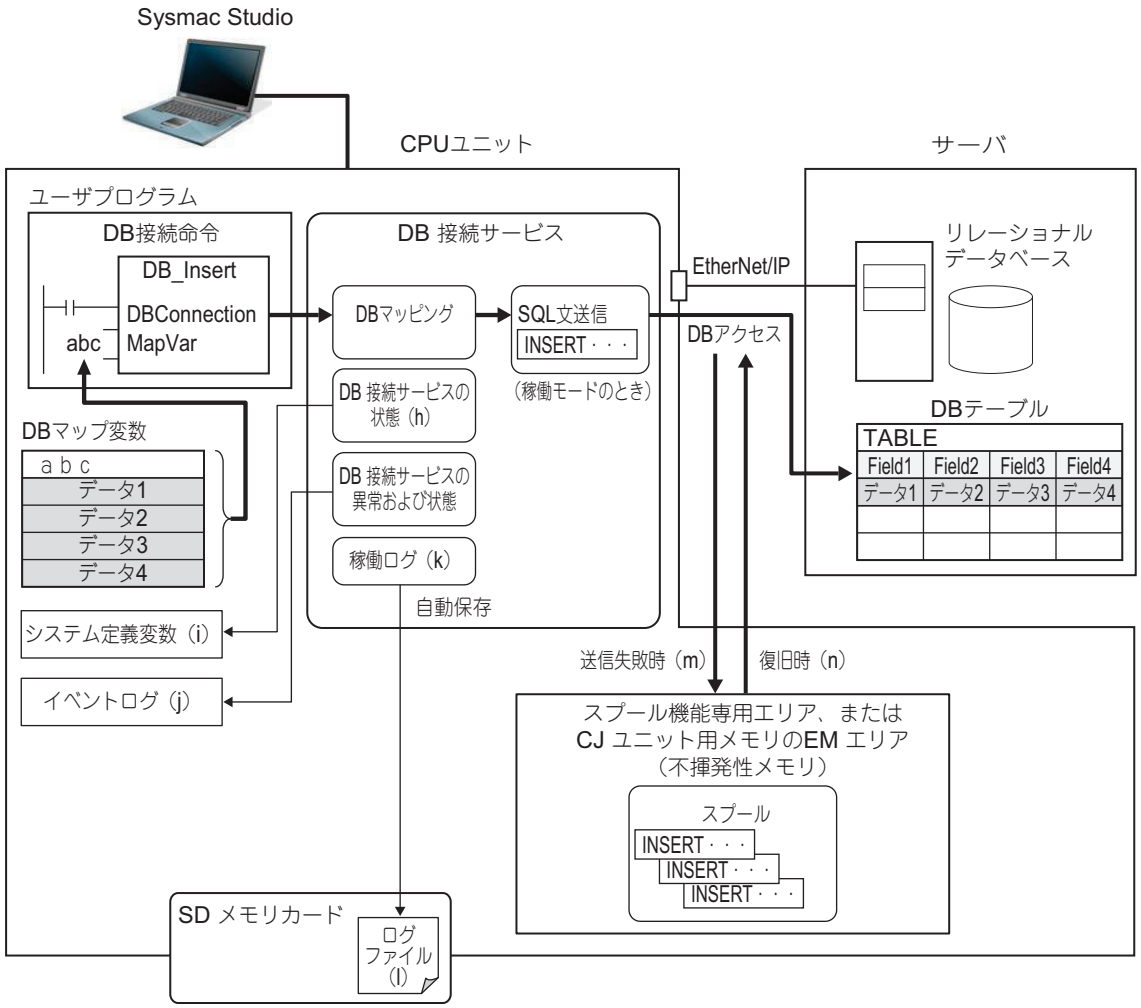


基本的なしくみ（番号は処理の順を示します）	参照
1．あらかじめ、DB のテーブルに合わせた構造体データ型を作成しておきます（図 a）。	「3-2 構造体データ型の作成 (P.3-3)」

基本的なしくみ（番号は処理の順を示します）	参照
2．上記構造体データ型を使用した変数（“DB マップ変数”と呼ぶ）を作成します（図 b）。	「3-3 DB マップ変数の作成 (P.3-15)」
3．DB 接続サービスを開始します（図 c）。 以下の条件に応じて、DB 接続サービスの動作モードを選択します。 • DB が存在し接続しているとき：稼働モードを選択 • DB が存在しない、または接続していないとき：テストモードを選択	「4-1 DB 接続サービスの動作モードと開始／停止 (P.4-2)」
4．DB コネクション接続命令にて、接続対象の DB コネクション名を指定して接続します。	「4-2 各 DB コネクションの接続／切断 (P.4-6)」
5．DB マップ生成命令にて、DB マップ変数を指定して、構造体データ型を DB のテーブルにマッピングします（“DB マッピング”と呼ぶ）（図 d）。	「3-4 DB マップ変数の DB へのマッピング (P.3-18)」
6．DB マップ変数を指定し、次の DB 接続命令を実行します（図 e）。 • レコード挿入 • レコード一括挿入 • レコード更新 • レコード取得 • ストアドプロシージャ 動作モードが稼働モードのとき、対応した SQL 文が送信されます（図 f）。	「3-5 DB 接続命令のプログラミング／転送 (P.3-21)」
7．ストアドプロシージャ機能を使用した場合、DB でストアドプロシージャを実行します。（図 g）	「5-3 ストアドプロシージャ呼出機能 (P.5-15)」

その他のしくみ

DB 接続機能のその他のしくみを、下図を使用して説明します。



その他のしくみ	参照
• DB 接続サービスの状態および各 DB コネクションの状態（図中の h）を、Sysmac Studio、DB 接続サービスの状態取得命令、またはシステム定義変数（図中の i）で確認できます。	「第 4 章 DB 接続サービスの開始／コネクションの接続と状態確認 (P.4-1)」
• DB 接続サービスの異常および状態が、イベントログ（図中の j）として格納されます。	「第 8 章 異常の確認と処置 (P.8-1)」
• DB 接続サービスの CPU ユニット上の動作をトレースしたログ（“稼働ログ”と呼ぶ）（図中の k）が、CPU ユニット装着の SD メモリカードに、ログファイル（図中の l）として保存されます。	「第 6 章 稼働ログの使用手順 (P.6-1)」
• SQL 文の送信に失敗したとき、送信できなかった SQL 文が、NX シリーズの場合スプール機能専用エリア、NJ シリーズの場合 CJ ユニット用メモリの EM エリアに自動的に保留（図中の m）されます。 復旧すると、保留された SQL 文が自動的または命令により再送されます（図中の n）。	「5-2 スプール機能 (P.5-5)」

1-3 DB 接続サービスの使用手順

以下に、基本的なフローを示します。
DB 接続サービスは、基本的に、以下のフローで使用します。

STEP1 Sysmac Studio の起動	「2-1 Sysmac Studio の起動と新規プロジェクト作成 (P.2-2)」

STEP2 新規プロジェクトの作成	「2-1 Sysmac Studio の起動と新規プロジェクト作成 (P.2-2)」

STEP3 DB 接続設定	「2-2 DB 接続設定 (P.2-5)」

DB 接続サービス全体の設定、および各 DB コネクションの設定をします。また、必要に応じ、Sysmac Studio から DB への通信テストをします。

(1)DB サービス全体の設定：
［構成・設定］－［上位接続設定］－［DB 接続］－［DB 接続サービス設定］の［サービス設定］で、以下を設定します。
サービス開始、実行ログ設定、デバッグログ設定、SQL 実行失敗ログ設定

(2)各 DB コネクションの設定：
［構成・設定］－［上位接続設定］－［DB 接続］－［DB コネクション設定］で、各 DB コネクション[*1] を追加の上、以下を設定します。
- DB 種別
- IP アドレス（サーバ側の IP アドレス）
- DB 名（サーバ側のデータベース名）
- ユーザ名、パスワードなど
- スプール設定

*1. 形式ごとの DB コネクション数については、「1-2-1 DB 接続サービスの仕様 (P.1-4)」を参照してください。

(3)Sysmac Studio からの DB との通信テスト（必要時のみ）：
［構成・設定］－［上位接続設定］－［DB 接続］－［DB コネクション設定］の［DB 通信テスト］にて、［通信テスト］ボタンをクリックします。

STEP4 DB にアクセスするための構造体データ型の作成	「3-2 構造体データ型の作成 (P.3-3)」

以下のメンバから成る、DB にアクセスするための構造体データ型を作成します。
- メンバ名：サーバ側 DB のテーブルのカラム名と一致させる
- メンバのデータ型：サーバ側 DB のテーブルのカラムの対応データ型に合わせる

STEP5　DB マップ変数の作成	「3-3 DB マップ変数の作成 (P.3-15)」

STEP4 で作成した構造体データ型を使用した変数（“DB マップ変数”と呼ぶ）を作成します。

STEP6　DB 接続命令によるプログラミング	「3-4 DB マップ変数の DB へのマッピング (P.3-18)」 「3-5 DB 接続命令のプログラミング／転送 (P.3-21)」

(1)初期処理

a）DB_ControlService（DB 接続サービスの制御）命令を記述します。
（命令によって DB 接続サービス開始時）。

b）DB_Connect（DB コネクション接続）命令もしくは DB_AttachProcedure（DB ストアドプロシージャ ハンドル生成）命令を記述します。

c）DB_CreateMapping（DB マップ生成）命令を記述します。
これにより、DB マップ変数が、サーバ側 DB のカラムに関連付けられ、レコード処理対象の変数として登録されます。

(2)運転中処理*1

a）レコード処理命令、ストアドプロシージャ命令などを記述します。

(3)終了処理

a）DB_Close（DB コネクション切断）命令を記述します。

(4)電源断処理*2

a）DB_Shutdown（DB 接続サービスのシャットダウン）命令を記述します。

*1. レコード処理命令、ストアドプロシージャ命令などを継続して実行する場合は、(2) 運転中処理のみを繰り返します。

*2. システムの電源を OFF にする場合は、DB_Shutdown 命令を実行してください。
DB_Shutdown 命令を実行せずに電源を OFF にした場合は、稼働ログファイルが壊れる、または内容が消失する場合があります。

STEP7　プロジェクトの CPU ユニットへの転送	「3-5 DB 接続命令のプログラミング／転送 (P.3-21)」

STEP8　DB 接続サービスの開始	「第 4 章 DB 接続サービスの開始／コネクションの接続と状態確認 (P.4-1)」

以下のいずれかの方法により、DB 接続サービスを開始します。

- CPU ユニットの動作モードをプログラムモードから運転モードに切り替えた時にサービスを開始する
- ［構成・設定］－［上位接続設定］－［DB 接続］－［DB 接続サービス設定］の［オンライン設定］で、［テストモード開始］または［稼働モード開始］ボタンをクリックする
- DB_ControlService（DB 接続サービスの制御）命令を実行する

DB 接続サービスの開始時に、同時に以下の動作モードを指定します。
- サーバ側 DB が存在しない環境または存在するが接続しない場合：“テストモード”とします。
- サーバ側 DB に接続する場合：“稼働モード”とします。

STEP9　DB 接続命令の実行	「3-5-3 DB 接続命令一覧 (P.3-22)」 「第 7 章 DB 接続命令 (P.7-1)」

システム定義変数「_DBC_Status.Run」（DB 接続サービスステータスの動作中フラグ）で DB 接続サービスの稼働状態が“動作中”であることを確認した上で、DB 接続命令を実行します。

STEP10　DB 接続命令のデバッグ	「3-6 設計、立ち上げ、運用時の各デバッグ (P.3-26)」

STEP11　Sysmac Studio による、ステータスの確認	「第 7 章 DB 接続命令 (P.7-1)」

DB サービス全体の状態、および各 DB コネクションの接続状態を確認できます。
- DB サービス全体の状態：
  ［構成・設定］－［上位接続設定］－［DB 接続］－［DB 接続サービス設定］を右クリックし［DB 接続サービスモニタ］で、DB 接続サービス全体の状態を確認できます。
- 各 DB コネクションの接続状態：
  ［構成・設定］－［上位接続設定］－［DB 接続］－［DB コネクション設定］を右クリックし［コネクションモニタテーブル］にて、各 DB コネクションの接続状態を確認できます。

STEP12　稼働ログの確認	「第 6 章 稼働ログの使用手順 (P.6-1)」

DB 接続サービスの CPU ユニット上の動作をトレースする、以下の稼働ログを確認できます。
- 実行ログ
  DB 接続サービスの実行をトレースするログです。DB 接続サービス動作中、常にロギングします。
  (1)［構成・設定］－［上位接続設定］－［DB 接続］を右クリックし［稼働ログ表示］にて［実行ログ］を表示します。
- デバッグログ
  DB 接続サービスが内部的にどの SQL 文を実行したか、SQL 文ごとのパラメータ、実行結果をトレースするログです。
  (1)［構成・設定］－［上位接続設定］－［DB 接続］を右クリックし［稼働ログ表示］にて［デバッグログ］を表示します。
- SQL 実行失敗ログ
  DB 側での SQL 文の実行失敗ログです。
  (1)［構成・設定］－［上位接続設定］－［DB 接続］を右クリックし［稼働ログ表示］にて［SQL 実行失敗ログ］を表示します。

STEP13　イベントログの確認	「第 8 章 異常の確認と処置 (P.8-1)」

2

DB 接続設定

この章では、DB 接続サービスを使用するための初期設定としての DB 接続設定について説明しています。

2-1 Sysmac Studio の起動と新規プロジェクト作成 2-2
2-1-1 Sysmac Studio の起動 2-2
2-1-2 新規プロジェクト作成 2-2
2-1-3 内蔵 EtherNet/IP ポート設定 2-3
2-1-4 コントローラ設定 2-3

2-2 DB 接続設定 2-5
2-2-1 DB 接続サービス設定 2-5
2-2-2 DB コネクション設定 2-7

2-1 Sysmac Studio の起動と新規プロジェクト作成

ここでは、DB 接続機能使用時の、Sysmac Studio の起動と新規プロジェクトの作成方法を説明します。
操作の詳細は、『Sysmac Studio Version 1 オペレーションマニュアル（SBCA-470）』を参照してください。
CPU ユニットと DB 接続サービスのバージョンの対応関係、および CPU ユニットと Sysmac Studio とのバージョンの対応関係についての詳細は、「A-4 バージョン関連情報 (P.A-26)」を参照してください。

2-1-1 Sysmac Studio の起動

1 **以下の Sysmac Studio をインストールします。**

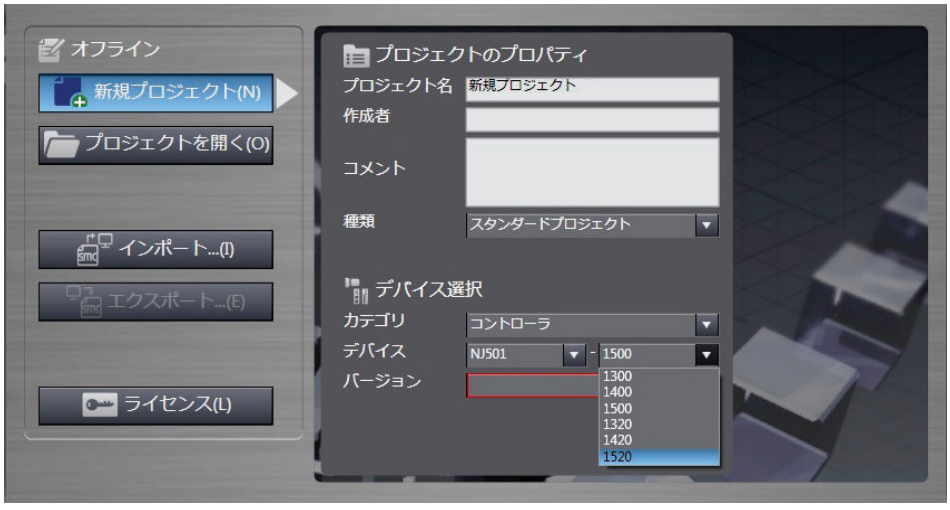
- 形 NX701-□□20 の場合：Ver.1.21 以降
- 形 NX502-1□00 の場合：Ver.1.54 以降
- 形 NX102-□□20 の場合：Ver.1.24 以降
- 形 NJ501-□□20 または形 NJ101-□□20 の場合：Ver.1.14 以降

2 **Sysmac Studio を起動します。**

2-1-2 新規プロジェクト作成

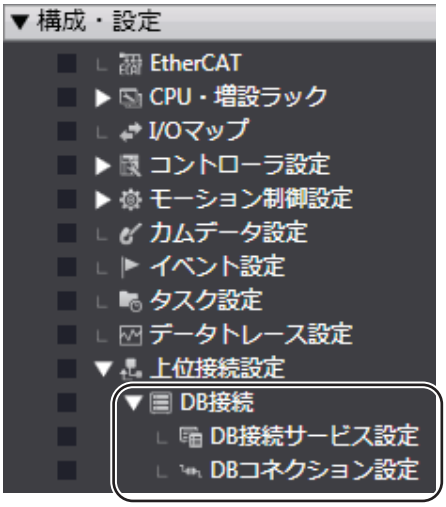
1 **［デバイス選択］の［デバイス］で、以下を選択します。**

［NX701］：［1720］、［1620］、のいずれか
［NX502］：［1500］、［1400］、［1300］のいずれか
［NX102］：［1220］、［1120］、［1020］、［9020］のいずれか
［NJ501］：［1520］、［1420］、［1320］、［4320］のいずれか
［NJ101］：［1020］、［9020］のいずれか



2 **［作成］ボタンをクリックします。**

マルチビューエクスプローラ上の［上位接続設定］の下に［DB 接続］が表示されます。



2-1-3 内蔵 EtherNet/IP ポート設定

1 **Sysmac Studio のマルチビューエクスプローラ上の［構成・設定］－［コントローラ設定］－［内蔵 EtherNet/IP ポート設定］を右クリックして［編集］を選択します。**

2 **内蔵 EtherNet/IP ポート設定画面で TCP/IP 設定、LINK 設定、FTP 設定、NTP 設定、SNMP 設定、SNMP トラップ設定、FINS 設定項目を設定します。**

CPU ユニットの内蔵 EtherNet/IP ポートを使用する場合の設定の詳細は、『NJ/NX シリーズ CPU ユニット内蔵 EtherNet/IP ポート ユーザーズマニュアル (SBCD-377)』を参照してください。
CPU ユニットに接続した NX シリーズ EtherNet/IP ユニットの EtherNet/IP ポートを使用する場合の設定の詳細は、『NX シリーズ EtherNet/IP ユニット ユーザーズマニュアル (SBCD-382)』を参照してください。

DB 接続サービスを使用する場合、EtherNet/IP ポートでは、以下のポート No.が使用されますので、重複して設定しないようにしてください。
形 NX701-□□□□、形 NX502-□□□□、形 NX102-□□□□、形 NJ501-□□□□および形 NJ101-□□□□の CPU ユニットの内蔵 EtherNet/IP ポート共通で使用されているポート No.については、『NJ/NX シリーズ CPU ユニット　内蔵 EtherNet/IP ポート　ユーザーズマニュアル（SBCD-377)』を参照してください。
CPU ユニットに接続した NX シリーズ EtherNet/IP ユニットの EtherNet/IP ポートで使用されているポート No.については、『NX シリーズ EtherNet/IP ユニット ユーザーズマニュアル (SBCD-382)』を参照してください。

アプリケーション	UDP	TCP
システムで使用	－	9800～9819

2-1-4 コントローラ設定

コントローラ設定とはコントローラの動作に関する設定をすることです。
下記以外の設定の詳細は、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル ソフトウェア編 (SBCA-467)』を参照してください。

動作設定

1 Sysmac Studio のマルチビューエクスプローラ上の［構成・設定］－［コントローラ設定］－［動作設定］を右クリックして［編集］を選択します。

● 全般設定

電源投入時や動作モード切替時などにおける動作定義など、CPU ユニットがサポートする機能に関する設定です。

設定項目	設定中項目	機能説明	設定値	デフォルト	反映タイミング	運転モード中書替
動作設定	電源投入時の起動待ち時間	電源投入後の起動中に、システムサービスの処理を優先的に行うための時間を設定します。*1	0 ～ 10s*2	0s	設定転送時	不可

*1. 本設定により、DB 接続サービスの起動時間を短縮できる場合があります。システムサービスの処理を優先する場合は“10”を、優先しない場合は“0”を設定してください。
“10”を設定した場合、CPU ユニットは、電源投入後の“起動中”に、約 10 秒間システムサービスの処理を優先してから“正常運転中”に移行します。“起動中”にシステムサービスの処理の一部を優先することで、DB 接続サービスが利用できる状態（システム定義変数「_DBC_Status.Run」が True）になるまでの時間を短縮できます。
“1～10”を設定した場合、設定した時間だけシステムサービスの処理を優先するため、CPU ユニットが“正常運転中”に移行するまでの時間が長くなります。

*2. 以下の CPU ユニットと Sysmac Studio の組み合わせでは、「0～30s」を設定できます。本設定は、CPU ユニットの電源投入から「正常運転中」に移行するまでの時間に影響するため、最大 20 秒を目安に設定してください。

CPU ユニット	Sysmac Studio
• 形 NX102-□□20 の Ver.1.35 以降 • 形 NX701-□□20、形 NJ501-1□20、形 NJ501-4320、形 NJ101-□□20 の Ver.1.23 以降	1.41 以降
形 NX502-1□00 の Ver.1.60 以降	1.54 以降

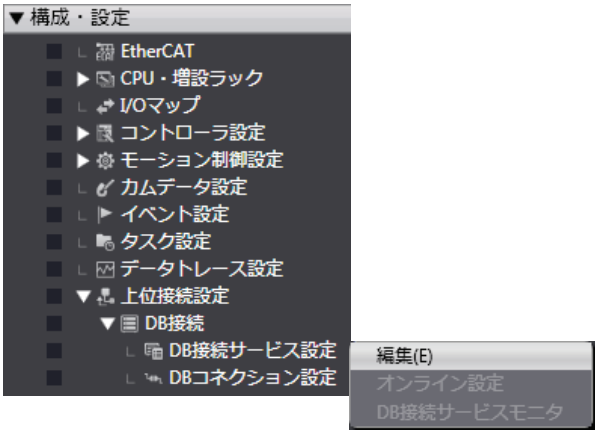
ただし、10 秒を超える値を設定すると、軽度フォールトレベルのコントローラ異常が発生する場合があります。軽度フォールトレベルのコントローラ異常が発生した場合は、異常解除を実行してください。

2-2 DB 接続設定

DB 接続サービスの実行の前に、初期設定として、DB 接続設定を行うことが必要です。
DB 接続設定には、DB 接続サービス全体の設定と DB コネクションごとの設定があります。
以下に、DB 接続サービス設定と DB コネクション設定について、説明します。

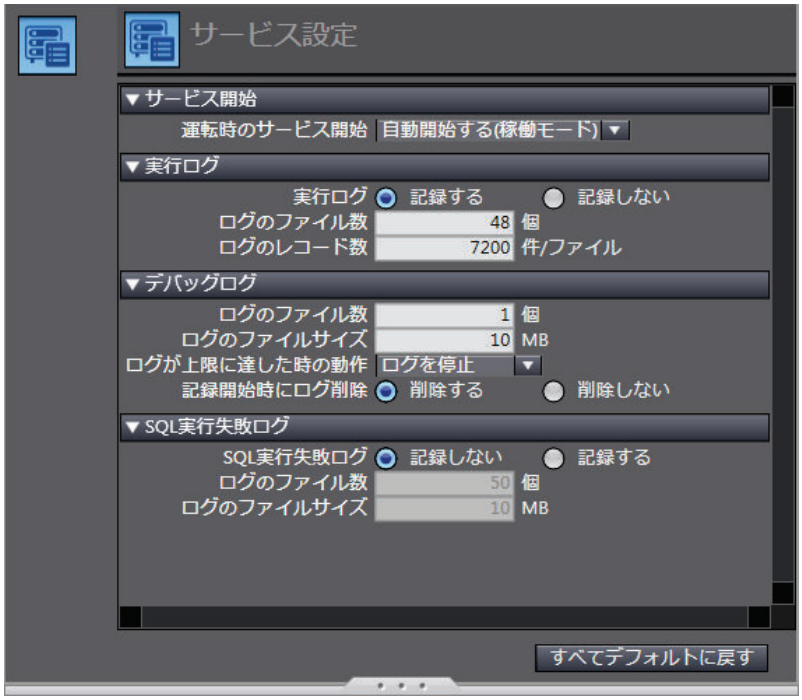
2-2-1 DB 接続サービス設定

Sysmac Studio のマルチビューエクスプローラ上の［構成・設定］－［上位接続設定］－［DB 接続］の下の［DB 接続サービス設定］を右クリックして［編集］を選択します。



サービス設定

［サービス設定］にて、サービス開始、実行ログ、デバッグログ、SQL 実行失敗ログの各設定をします。
DB 接続サービス開始についての詳細は、「4-1 DB 接続サービスの動作モードと開始／停止 (P.4-2)」を参照してください。
稼働ログについての詳細は、「第 6 章 稼働ログの使用手順 (P.6-1)」を参照してください。



以下の項目を設定します。

分類	項目	内容	値
サービス開始	運転時のサービス開始	CPU ユニットの動作モードが運転モードになったときに、DB 接続サービスを自動で開始するかを設定します。	• 使用しない（デフォルト）*1 • 自動開始する（稼働モード）*2 • 自動開始する（テストモード）*3 • 自動開始しない
実行ログ	実行ログ	実行ログを記録するかを設定します。	• 記録する • 記録しない（デフォルト）*4
	ログのファイル数	実行ログの最大ファイル数を設定します。 最大ファイル数に達した場合は、一番古いファイルが削除され、新しいファイルが作成されます。	2～100 個 （デフォルト：48 個）
	ログのレコード数	実行ログファイルの 1 ファイルに保存可能なレコード数を設定します。 設定したレコード数に達すると、新しいファイルが作成されます。	100～65536 件 （デフォルト：7200 件）
デバッグログ	ログのファイル数	デバッグログの最大ファイル数を設定します。	1～100 個 （デフォルト：1 個）
	ログのファイルサイズ	1 ファイルの最大サイズを設定します。 1 ファイルの最大サイズを超えるか、1 ファイルのレコード数が 65536 件を超えたとき、新しいファイルが作成されます。	1～100MB （デフォルト：10MB）
	ログが上限に達した時の動作	最大ファイル数に達した場合の動作を設定します。	• ログを継続（最も古いファイル削除） • ログを停止（デフォルト）
	記録開始時にログ削除	記録開始時に SD メモリカード内のデバッグログファイルを削除するかを設定します。	• 削除する（デフォルト） • 削除しない

分類	項目	内容	値
SQL 実行失敗ログ	SQL 実行失敗ログ	SQL 実行失敗ログを記録するかを設定します。	• 記録する（デフォルト） • 記録しない
	ログのファイル数	SQL 実行失敗ログの最大ファイル数を設定します。 最大ファイル数に達した場合は、一番古いファイルが削除され、新しいファイルが作成されます。	2～100 個 （デフォルト：50 個）
	ログのファイルサイズ	1 ファイルの最大サイズを設定します。 1 ファイルの最大サイズを超えるか、1 ファイルのレコード数が 65536 件を超えたとき、新しいファイルが作成されます。	1～100MB （デフォルト：10MB）

*1. 形 NX502-1□00 以外の場合、デフォルトは［自動開始する（稼働モード）］になります。また、形 NX502-1□00 以外の場合、［使用しない］は選択できません。
*2. DB 接続命令の実行により、実際の DB にアクセスします。
*3. DB 接続命令の実行によって DB にアクセスはせず、実行したと想定して正常終了します。
*4. 形 NX502-1□00 の場合、デフォルトは［記録する］になります。

参考

SD メモリカードに保存される稼働ログのファイル容量は、以下のように計算できます。
SD メモリカードの容量不足が頻繁に発生する場合は、各設定を見直してください。

- 実行ログ
  1 レコードのサイズ（注）×“ログのレコード数”×“ログのファイル数”
- デバッグログ
  “ログのファイルサイズ”×“ログのファイル数”
- SQL 実行失敗ログ
  “ログのファイルサイズ”×“ログのファイル数”

（注）DB 接続サービスのバージョンによって最大値が異なります。
Ver.1.04 以前：最大 256 バイト
Ver.2.00 以降：最大 58KB

2-2-2 DB コネクション設定

以下に、DB コネクションの追加、DB コネクション名の変更、コネクションの設定方法とその内容を示します。

DB コネクションの追加

1 **Sysmac Studio の［構成・設定］－［上位接続設定］－［DB 接続］の下の［DB コネクション設定］を右クリックし、［追加］－［DB コネクション設定］を選択します。［挿入］メニューの［DB コネクション設定］からも追加できます。**



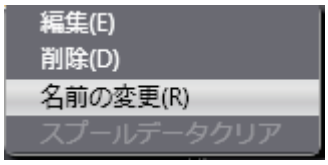
DB コネクションが追加されます。*1



*1. 形式ごとの DB コネクション数については、「1-2-1 DB 接続サービスの仕様 (P.1-4)」を参照してください。

DB コネクション名の変更

デフォルトの DB コネクション名は、つぎのように自動生成されます。＊＊は 01 からの追番です。
“DBConnection＊＊”
変更する場合、マルチビューエクスプローラ上の追加した DB コネクションを右クリックし［名前の変更］を選択します。



- 入力可能文字は、半角英数字とアンダーバー（_）です。
- DB コネクション名は、最大 16 バイトです。

● DB コネクションの編集または削除

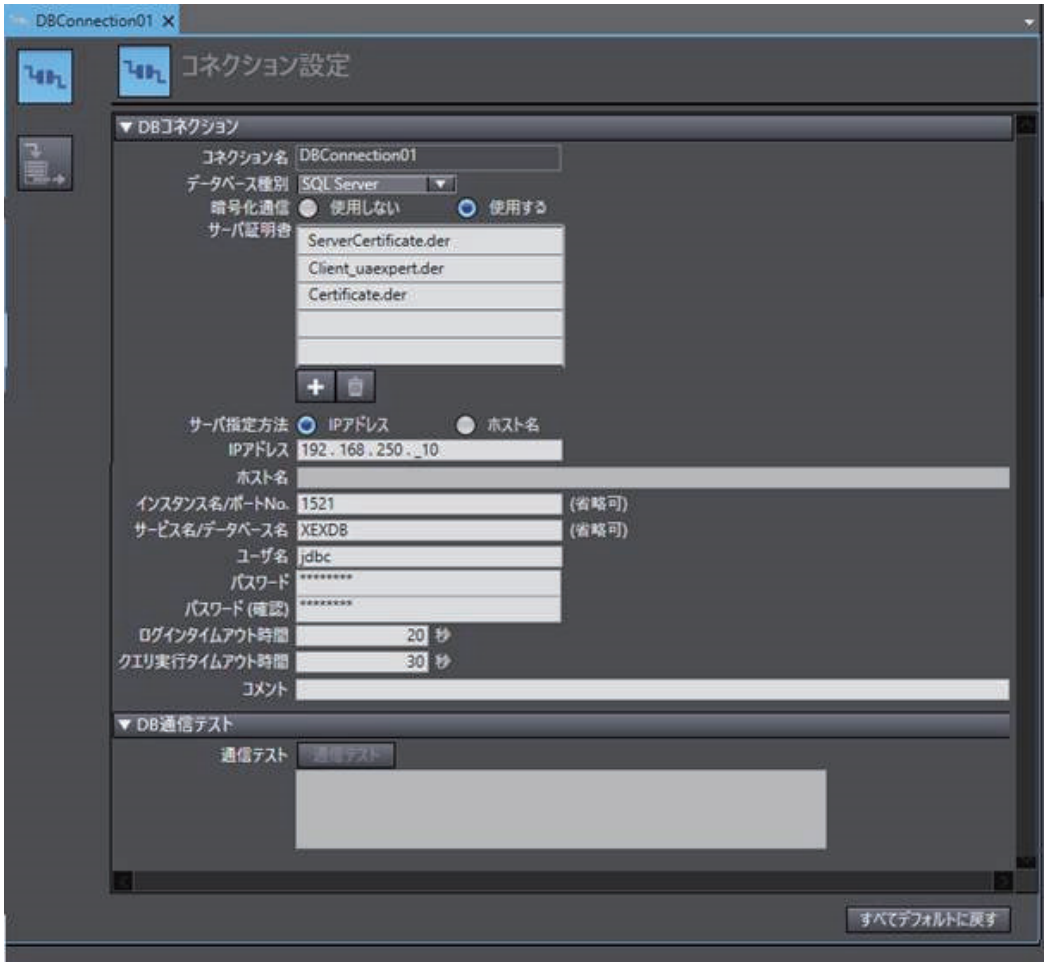
マルチビューエクスプローラ上の追加した DB コネクションを右クリックし［編集］または［削除］を選択します。

コネクション設定

DB コネクションごとの設定、および通信テストの方法を以下に示します。

● DB コネクションの設定

追加した各 DB コネクションをダブルクリックし、以下の［コネクション設定］で設定します。



以下の項目を設定します。

分類	項目	内容	値
DBコネクション	コネクション名	DB コネクション名を表示します。	DB コネクション名の変更は、マルチビューエクスプローラで変更する対象の DB コネクションを右クリックし、メニューから［名前の変更］を選択します。
	データベース種別	DB の種別を設定します。	• 形 NX701-□□20、形 NX502-1□00、形 NX102-□□20、形 NJ501-1□20 または形 NJ101-□□20 Oracle SQL Server（デフォルト） DB2 MySQL Firebird PostgreSQL • 形 NJ501-4320 Oracle SQL Server（デフォルト） MySQL
	暗号化通信 *1	コネクションごとの暗号化通信の有無を選択します。	使用しない、使用する デフォルト：使用しない
	サーバ証明書*1	サーバ証明書（CA/ルート証明書含む）を選択します。（複数選択可。1 コネクションあたり最大 5。）	選択したサーバ証明書のファイル名 デフォルト：空白 使用しない場合：選択不可 使用する場合：サーバ証明書ファイル名 Oracle の場合は省略不可
	サーバ指定方法	サーバの指定方法を選択します。 IP アドレス、または、ホスト名で選択します。	• IP アドレス（デフォルト） • ホスト名
	IP アドレス	サーバ側の IP アドレスを設定します。	デフォルト：空白 サーバ指定方法が IP アドレスの場合は省略不可。
	ホスト名	サーバのホスト名を設定します。*2	デフォルト：空白 サーバ指定方法がホスト名の場合は省略不可。

分類	項目	内容	値
	インスタンス名/ポート No.	サーバ側のインスタンス名またはポート No.を設定します。	• Oracle の場合： ポート No.（省略可）*3 例：1521 • SQL Server の場合： インスタンス名またはポート No.（省略可） 例：INSTANCE1 または 1433 • DB2 の場合 ポート No.（省略不可） 例：50000 • MySQL の場合 ポート No.（省略可） 例：3306 • Firebird の場合 ポート No.（省略可） 例：3050 • PostgreSQL の場合 ポート No.（省略可） 例：5432 インスタンス名の最大文字数：64 文字 ポート No.：1～65535 デフォルト：空白 省略した場合は、デフォルトのポート No.を使用します。 • Oracle の場合： 1521 • SQL Server の場合：1433 • MySQL の場合：3306 • Firebird の場合：3050 • PostgreSQL の場合：5432
	サービス名/データベース名	サーバ側のサービス名またはデータベース名を設定します。	• Oracle の場合：サービス名（省略可）*3 • SQL Server の場合：データベース名（省略可） • DB2 の場合：データベース名（省略不可） • MySQL の場合：データベース名（省略不可） • Firebird の場合：データベースパス（省略不可） 例：C:/Firebird/OMRON.FDB または C:\Firebird\OMRON.FDB • PostgreSQL の場合：データベース名（省略不可） 最大バイト数：127 バイト 省略した場合は、 • Oracle の場合：デフォルトのサービス • SQL Server の場合：デフォルトのデータベース

分類	項目	内容	値
	ユーザ名	サーバ側のユーザ名を設定しま す。	• DB2 の場合：サーバ側 Windows のユーザ名 • DB2 以外の場合：サーバ側 DB のユーザ名 最大文字数：127 文字 デフォルト：空白
	パスワード	サーバ側のパスワードを設定します。	• DB2 の場合：サーバ側 Windows のパスワード • DB2 以外の場合：サーバ側 DB のパスワード 最大文字数：127 文字 デフォルト：空白
	ログインタイムアウト時間	サーバ側 DB に接続する時のタイムアウト時間を設定します。	1～60 秒 デフォルト：10 秒
	クエリ実行タイムアウト時間	SQL 実行時のタイムアウト時間を設定します。	1～600 秒 デフォルト：30 秒
	コメント	コメントを設定します。	最大バイト数：1024 バイト デフォルト：空白 省略可。

*1. DB 接続サービス Ver.2.00 以降で表示されます。
*2. サーバをホスト名で指定する場合、内蔵 EtherNet/IP ポート設定で“DNS を使用する”または、“hosts 設定”での設定が必要です。CPU ユニットの内蔵 EtherNet/IP ポートを使用する場合の設定の詳細は、『NJ/NX シリーズ CPU ユニット内蔵 EtherNet/IP ポート ユーザーズマニュアル (SBCD-377)』を参照してください。CPU ユニットに接続した NX シリーズ EtherNet/IP ユニットの EtherNet/IP ポートを使用する場合の設定の詳細は、『NX シリーズ EtherNet/IP ユニット ユーザーズマニュアル (SBCD-382)』を参照してください。
*3. DB 接続サービス Ver.2.00 以降で、暗号化通信が「使用する」の場合は省略不可です。

バージョン関連情報

接続可能なデータベース種別は、CPU ユニットの DB 接続サービスのバージョンと Sysmac Studio プロジェクトの DB 接続サービスのバージョンの組み合わせにより、形式ごとに異なります。
ユニットバージョンと Sysmac Studio での設定ユニットバージョンの関係は、「A-4-3 ユニットバージョン／DB 接続サービスのバージョンと Sysmac Studio の設定ユニットバージョン (P.A-28)」を参照してください。

● 通信テスト

Sysmac Studio の DB コネクション設定画面で設定した内容*1 で、DB との接続を試みる機能です。
*1. コントローラに転送された DB コネクション設定ではありません。
Sysmac Studio が、コントローラとオンライン接続状態のときに実行可能です。

1 **同期機能を使用し、パソコンからコントローラに DB 接続設定を転送します。**

2 **[DB 通信テスト] の下の [通信テスト] ボタンをクリックします。**

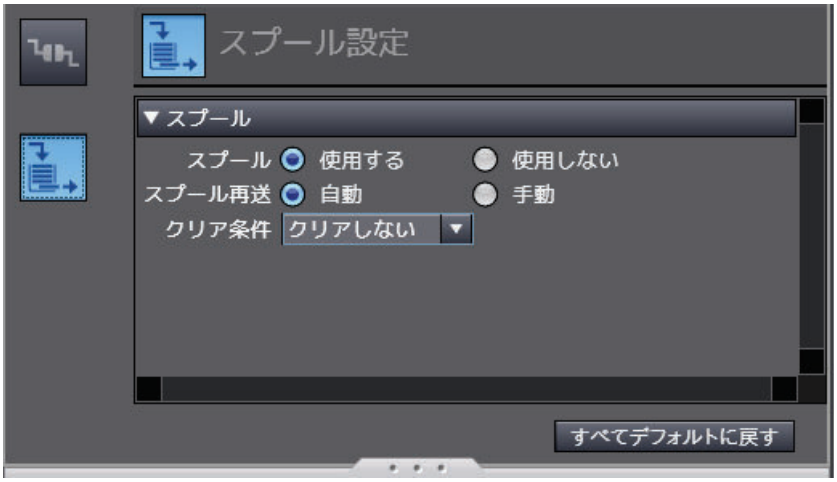
3 **通信テストの結果が、[通信テスト] ボタンの下のテキストボックスに表示されます。**
サーバとの接続に何らかの原因で失敗した場合、SQL ステータス、エラーコードおよび詳細エラーメッセージが表示されます。

SQL ステータス：　SQL 標準規格（ISO/IEC9075、JISX3005-2）で定義されたエラーコード
エラーコード：　接続対象の DB ベンダ固有のエラーコード
ネットワーク障害発生時は、エラーコードが 0 と表示される場合があります。0 と表示されている場合は、SQL ステータスを確認してください。

詳細エラーメッセージ：　　接続対象の DB ベンダ固有のエラーメッセージ

スプール設定

［スプール設定］にて、スプール機能関連の設定をします。



設定の詳細は、「5-2 スプール機能 (P.5-5)」を参照してください。

3

DB 接続機能のプログラミング

この章では、DB 接続機能設定を行った後に、変数の作成から DB にアクセスするまでのプログラミングについて説明しています。

3-1　DB にアクセスするまでの手順 3-2

3-2　構造体データ型の作成 3-3
3-2-1　概要 3-3
3-2-2　DB にアクセスするための構造体データ型の仕様 3-3
3-2-3　DB にアクセスするための構造体データ型の作成方法 3-11

3-3　DB マップ変数の作成 3-15
3-3-1　DB マップ変数、および DB へのマッピング 3-15
3-3-2　DB マップ変数の登録および属性 3-16
3-3-3　DB マップ変数の制約事項 3-17

3-4　DB マップ変数の DB へのマッピング 3-18
3-4-1　DB マップ生成命令による DB へのマッピング 3-18
3-4-2　DB マップ変数のマッピングの解除 3-18
3-4-3　DB マッピングの制約 3-18

3-5　DB 接続命令のプログラミング／転送 3-21
3-5-1　DB 接続サービスのプログラミングの概要 3-21
3-5-2　Sysmac Studio での DB 接続命令の表示 3-21
3-5-3　DB 接続命令一覧 3-22
3-5-4　システム定義変数 3-23
3-5-5　DB 接続命令のシミュレーションデバッグ 3-24
3-5-6　DB 接続設定とユーザプログラムの転送 3-24

3-6　設計、立ち上げ、運用時の各デバッグ 3-26
3-6-1　設計時 3-26
3-6-2　立ち上げ時 3-26
3-6-3　運用時 3-26

3-1 DB にアクセスするまでの手順

ここでは、DB 接続サービスを使用するとき特有のプログラミング手順について説明をしています。プログラミングの全体については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル ソフトウェア編（SBCA-467）』を参照してください。

DB 接続設定を行った後に、DB 接続命令によって DB にアクセスするまでの手順は、以下のとおりです。

DB マッピング*1 の後、DB 接続命令で、DB 操作を実行します。

DB マッピング*1	DB にアクセスするための構造体データ型を作成する	「3-2 構造体データ型の作成 (P.3-3)」
	↓	
	DB マップ変数（作成した構造体データ型を使用した変数）を作成する	「3-3 DB マップ変数の作成 (P.3-15)」
	↓	
	DB_Connect（DB コネクション接続）命令実行により、DB コネクションへ接続する	「4-2 各 DB コネクションの接続／切断 (P.4-6)」
	↓	
	DB_CreateMapping（DB マップ生成）命令もしくは DB_AttachProcedure（DB ストアドプロシージャ ハンドル生成）命令実行により、指定した DB のテーブルやストアドプロシージャの引数、戻り値、結果セットと DB マップ変数を SQL 種別ごとにマッピングする	「3-4 DB マップ変数の DB へのマッピング (P.3-18)」 ※ストアドプロシージャ呼出機能の場合「5-3-4 DB マップ変数の DB へのマッピング (P.5-18)」

↓

DB 操作	DB 接続命令を実行する	「3-5 DB 接続命令のプログラミング／転送 (P.3-21)」

*1. “DB マッピング”とは、DB にアクセスするための構造体型のデータ型の各メンバなどを、DB のテーブルの各カラムやストアドプロシージャの引数、戻り値、結果セットに関連付けることを指します。SQL 種別やストアドプロシージャごとに DB マッピングを実施する必要があります。

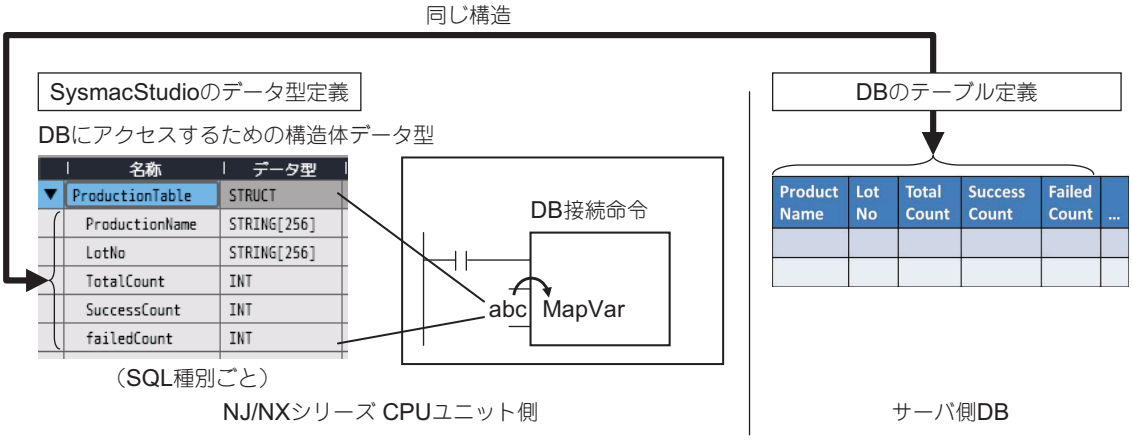
3-2　構造体データ型の作成

DB にアクセスするためには、あらかじめ DB のテーブル定義に合わせた、ユーザ定義の構造体データ型を作成する必要があります。
ここでは、その構造体データ型の仕様および作成方法を説明します。

3-2-1　概要

Sysmac Studio 上で、サーバ側 DB のテーブルに合わせて、ユーザ定義の構造体データ型を作成します。DB のテーブルのすべて、または一部のカラムを構造体メンバとして登録します。

構造体メンバの名称とデータ型をサーバ側 DB のテーブルのカラム（名称、データ型）と合わせます。



作成した構造体データ型は、DB マップ変数を作成するときにデータ型として指定します。

3-2-2　DB にアクセスするための構造体データ型の仕様

項目	仕様
構造体名	DB にアクセスするための構造体名は、任意です。
構造体のメンバのオフセット指定	“オフセット種別”は NJ/NX シリーズ共通で“NJ”とします。
構造体のメンバ	DB のテーブルのすべてのカラム、または一部のカラムをメンバとして登録します。
構造体のメンバ名	DB のテーブルのカラム名に合わせて定義します。大文字／小文字も合わせてください。
構造体のメンバのデータ型	メンバのデータ型は、DB のテーブルのカラムのデータ型に合わせて定義します。 後述の「NJ/NX シリーズ側のデータ型と DB 側のデータ型の対応 (P.3-4)」を参照してください。 ただし、メンバには、以下のデータ型およびその属性を指定することはできません。 • 派生データ型 • 配列属性

使用上の注意

DB 側のテーブルのカラム名側の制約：
NJ/NX シリーズ側の構造体メンバ名と、アクセス対象のサーバ側 DB のカラム名を同じ名称にする必要があります。
NJ/NX シリーズの構造体メンバ名は以下の制限があります。そのため、DB 側のカラム名も、以下の制限に従ってください。

項目	内容
使用できる文字	0-9、A-Z、a-z 半角カナ _（アンダーバー） マルチバイト文字（例：日本語）
組み合わせて使用できない文字	• 数字（0～9）で始まる文字列 • “P_” で始まる文字列 • アンダーバーで始まる文字列 • アンダーバー（_）を 2 個以上連続した文字列 • アンダーバー（_）で終端する文字列 • 識別子を構成する文字列の接頭または接尾に、1 文字以上の拡張空白類（全角空白など、Unicode まで拡張した空白系文字）を含む文字列

● NJ/NX シリーズ側のデータ型と DB 側のデータ型の対応

NJ/NX シリーズ側のデータ型と DB 側のデータ型の対応表をデータ種別ごとに以下に示します。

• Oracle の場合

データ型の種類	DB 側のデータ型	NJ/NX 側のデータ型
文字	VARCHAR2	STRING*1
	NVARCHAR2	STRING*1
	CHAR	STRING*1
	NCHAR	STRING*1
	LONG	なし
	CLOB	なし
	NCLOB	なし
数値*2		*3
	NUMBER(1)	BOOL
	NUMBER(3)	SINT
	NUMBER(5)	INT
	NUMBER(10)	DINT
	NUMBER(19)	LINT
	NUMBER(3)	USINT
	NUMBER(5)	UINT
	NUMBER(10)	UDINT
	NUMBER(20)	ULINT
	NUMBER(19)	TIME*4
	BINARY_FLOAT	REAL
	BINARY_DOUBLE	LREAL
	FLOAT	REAL
	INTEGER	DINT

データ型の種類	DB 側のデータ型	NJ/NX 側のデータ型
DATE	DATE	DATE
	TIMESTAMP	DATE DATE_AND_TIME
	TIMESTAMP WITH TIME ZONE	DATE_AND_TIME
	TIMESTAMP WITH LOCAL TIME ZONE	DATE_AND_TIME
	INTERVAL YEAR TO MONTH	なし
	INTERVAL DAY TO SECOND	なし
バイナリ	RAW	なし
	LONG RAW	なし
	BLOB	なし
その他	BFILE	なし
	ROWID	なし
	UROWID	なし
	XMLTYPE	なし

*1. 文字列の終端に NULL が付くため、STRING のデータ型で使用するバイト数は、DB 側データ型のバイト数+1 を設定する必要があります。STRING のデータ型で使用するバイト数は、DB 側データ型、文字コードに合わせて適切な値を設定する必要があります。NJ/NX シリーズでは、文字列を UTF-8 で扱います。半角英数字 1 文字は 1 バイト、マルチバイト文字列は複数バイトを使用します。日本語文字は 3 バイトが目安です。

*2. NUMBER(p , [,s])の小数点以下桁数(s)は、省略形式で記述しています。省略形式を使用した場合は、小数点以下桁数(s)は 0 です。もし省略せずに、小数点以下桁数(s)に 1 以上の数値を指定した場合、整数部のみが対象となります。

*3. 扱う範囲が異なるので、上記の対応どおりでもデータの桁あふれが生じるケースがあります。
例：DB 側のデータ型が NUMBER(3)、NJ/NX 側のデータ型が USINT の場合
- NUMBER(3)の範囲：0～999
- USINT の範囲：0～255

*4. ns 単位の整数型となります。

- SQL Server の場合

データ型の種類	DB 側のデータ型	NJ/NX 側のデータ型
数値*1	bigint	LINT UDINT
		TIME*2
	bit	BOOL
		*3
	decimal(1)	BOOL
	decimal(3)	SINT
	decimal(5)	INT
	decimal(10)	DINT
	decimal(19)	LINT
	decimal(20)	ULINT
	decimal(3)	USINT
	decimal(5)	UINT
	decimal(10)	UDINT
	decimal(19)	TIME
	int	DINT UINT
	money	LREAL*4

データ型の種類	DB 側のデータ型	NJ/NX 側のデータ型
		*3
	numeric(1)	BOOL
	numeric(3)	SINT
	numeric(5)	INT
	numeric(10)	DINT
	numeric(19)	LINT
	numeric(20)	ULINT
	numeric(3)	USINT
	numeric(5)	UINT
	numeric(10)	UDINT
	numeric(19)	TIME
	smallint	INT USINT
	smallmoney	REAL*5
	tinyint	USINT
	float	LREAL
	real	REAL
日付と時刻	date	DATE
	datetime2	DATE_AND_TIME*6
	datetime	DATE_AND_TIME
	datetimeoffset	DATE_AND_TIME*6
	smalldatetime	DATE_AND_TIME
	time	TIME_OF_DAY*6
文字列	char	STRING*7
	text	STRING*7
	varchar	STRING*7
	nchar	STRING*7
	ntext	STRING*7
	nvarchar	STRING*7
バイナリ	binary	なし
	image	なし
	varbinary	なし
その他	cursor	なし
	hierarchyid	なし
	sql_variant	なし
	table	なし
	uniqueidentifier	なし
	xml	なし

*1. decimal(p [,s])、numeric(p [,s])の小数点以下桁数(s)は、省略形式で記述しています。省略形式を使用した場合は、小数点以下桁数(s)は 0 です。もし省略せずに、小数点以下桁数(s)に 1 以上の数値を指定した場合、整数部のみが対象となります。

*2. ns 単位の整数型となります。

*3. 扱う範囲が異なるので、上記の対応どおりでもデータの桁あふれが生じるケースがあります。
例：DB 側のデータ型が decimal(3)、NJ/NX 側のデータ型が USINT の場合
- decimal(3)の範囲：0～999
- USINT の範囲：0～255

*4. 有効桁数は 15 桁となります。DB 接続命令で DB に書き込む時、小数第 5 位を四捨五入した値を書き込みます。
例：1.79769 を DB に書き込む時、1.7977 を書き込みます。

*5. 有効桁数は 7 桁となります。DB 接続命令で DB に書き込む時、小数第 5 位を四捨五入した値を書き込みます。
例：1.79769 を DB に書き込む時、1.7977 を書き込みます。

*6. 精度はミリ秒となります。
*7. 文字列の終端に NULL が付くため、STRING のデータ型で使用するバイト数は、DB 側データ型のバイト数+1 を設定する必要があります。
STRING のデータ型で使用するバイト数は、DB 側データ型、文字コードに合わせて適切な値を設定する必要があります。NJ/NX シリーズでは、文字列を UTF-8 で扱います。半角英数字 1 文字は 1 バイト、マルチバイト文字列は複数バイトを使用します。日本語文字は 3 バイトが目安です。

• DB2 の場合

データ型の種類	DB 側のデータ型	NJ/NX 側のデータ型
数値	INT	DINT
	INTEGER	DINT
	BIGINT	LINT TIME
	SMALLINT	INT
固定小数点*1		*2
	DECIMAL(1)	BOOL
	DECIMAL(3)	SINT
	DECIMAL(5)	INT
	DECIMAL(10)	DINT
	DECIMAL(20)	LINT
	DECIMAL(3)	USINT
	DECIMAL(5)	UINT
	DECIMAL(10)	UDINT
	DECIMAL(20)	ULINT
	DECIMAL(20)	TIME
実数	FLOAT	REAL LREAL
	REAL	REAL
	DOUBLE	LREAL
日付	DATE	DATE
	TIME	TIME_OF_DAY
	TIMESTAMP	DATE_AND_TIME
文字列	CHAR	STRING*3
	CHARACTER	STRING*3
	VARCHAR	STRING*3
	CHAR VARYING	STRING*3
	CHARACTER VARYING	STRING*3
	LONG VARCHAR	STRING*3
	CLOB	なし
バイナリ文字列	BLOB	なし
その他	GRAPHIC	なし
	VARGRAPHIC	なし
	LONG VARGRAPHIC	なし
	DBCLOB	なし
	DATALINK	なし

*1. DECIMAL(p [,s])の小数点以下桁数(s)は、省略形式で記述しています。省略形式を使用した場合は、小数点以下桁数(s)は 0 です。もし省略せずに、小数点以下桁数(s)に 1 以上の数値を指定した場合、整数部のみが対象となります。
*2. 扱う範囲が異なるので、上記の対応どおりでもデータの桁あふれが生じるケースがあります。
例：DB 側のデータ型が DECIMAL(3)、NJ/NX 側のデータ型が USINT の場合
• DECIMAL(3)の範囲：0～999
• USINT の範囲：0～255
*3. 文字列の終端に NULL が付くため、STRING のデータ型で使用するバイト数は、DB 側データ型のバイト数+1 を設定する必要があります。

STRING のデータ型で使用するバイト数は、DB 側データ型、文字コードに合わせて適切な値を設定する必要があります。NJ/NX シリーズでは、文字列を UTF-8 で扱います。半角英数字 1 文字は 1 バイト、マルチバイト文字列は複数バイトを使用します。日本語文字は 3 バイトが目安です。

- MySQL の場合

データ型の種類	DB 側のデータ型	NJ/NX 側のデータ型
数値*1	BIT	BOOL
	BOOL BOOLEAN	BOOL
	TINYINT	SINT USINT
	SMALLINT	INT UINT
	MEDIUMINT	DINT UDINT
	INT	DINT UDINT
	BIGINT	LINT ULINT TIME
		*2
	DECIMAL(1) DECIMAL(3) DECIMAL(5) DECIMAL(10) DECIMAL(20) DECIMAL(3) DECIMAL(5) DECIMAL(10) DECIMAL(20)	BOOL SINT INT DINT LINT USINT UINT UDINT ULINT
	DECIMAL(20)	TIME
	FLOAT	REAL
	DOUBLE	LREAL
日付と時刻	DATE	DATE
	DATETIME	DATE_AND_TIME
	TIMESTAMP	DATE_AND_TIME
	TIME	TIME_OF_DAY
文字列	CHAR	STRING*3
	VARCHAR	STRING*3
	TINYTEXT	STRING*3
	TEXT	STRING*3
	MEDIUMTEXT	STRING*3
	LONGTEXT	STRING*3
バイナリ	BINARY	なし
	VARBINARY	なし
	TINYBLOB	なし
	BLOB	なし
	MEDIUMBLOB	なし
	LONGBLOB	なし

データ型の種類	DB 側のデータ型	NJ/NX 側のデータ型
その他	ENUM	なし
	YEAR	なし
	SET	なし

*1. DECIMAL(p [,s])の小数点以下桁数(s)は、省略形式で記述しています。省略形式を使用した場合は、小数点以下桁数(s)は 0 です。もし省略せずに、小数点以下桁数(s)に 1 以上の数値を指定した場合、整数部のみが対象となります。

*2. 扱う範囲が異なるので、上記の対応どおりでもデータの桁あふれが生じるケースがあります。
例：DB 側のデータ型が DECIMAL(3)、NJ/NX 側のデータ型が USINT の場合
- DECIMAL(3)の範囲：0～999
- USINT の範囲：0～255

*3. 文字列の終端に NULL が付くため、STRING のデータ型で使用するバイト数は、DB 側データ型のバイト数+1 を設定する必要があります。
STRING のデータ型で使用するバイト数は、DB 側データ型、文字コードに合わせて適切な値を設定する必要があります。NJ/NX シリーズでは、文字列を UTF-8 で扱います。半角英数字 1 文字は 1 バイト、マルチバイト文字列は複数バイトを使用します。日本語文字は 3 バイトが目安です。

- Firebird の場合

データ型の種類	DB 側のデータ型	NJ/NX 側のデータ型
整数	INTEGER	DINT
	BIGINT	LINT TIME
	SMALLINT	INT
固定小数点*1		*2
	DECIMAL(1) DECIMAL(3) DECIMAL(5) DECIMAL(10) DECIMAL(18) DECIMAL(3) DECIMAL(5) DECIMAL(10) DECIMAL(18)	BOOL SINT INT DINT LINT*3 USINT UINT UDINT ULINT*3
		*2
	NUMERIC(1) NUMERIC (3) NUMERIC (5) NUMERIC (10) NUMERIC (18) NUMERIC (3) NUMERIC (5) NUMERIC (10) NUMERIC (18)	BOOL SINT INT DINT LINT*3 USINT UINT UDINT ULINT*3
実数	FLOAT	REAL
	DOUBLE PRECISION	LREAL
日付	DATE	DATE
	TIME	TIME_OF_DAY
	TIMESTAMP	DATE_AND_TIME
文字列	CHAR	STRING*4
	VARCHAR	STRING*4

データ型の種類	DB 側のデータ型	NJ/NX 側のデータ型
その他	BLOB	なし

*1. DECIMAL(p [,s])、NUMERIC(p [,s])の小数点以下桁数(s)は、省略形式で記述しています。省略形式を使用した場合は、小数点以下桁数(s)は 0 です。もし省略せずに、小数点以下桁数(s)に 1 以上の数値を指定した場合、整数部のみが対象となります。

*2. 扱う範囲が異なるので、上記の対応どおりでもデータの桁あふれが生じるケースがあります。
例：DB 側のデータ型が DECIMAL(3)、NJ/NX 側のデータ型が USINT の場合
- DECIMAL(3)の範囲：0～999
- USINT の範囲：0～255

*3. DB 側の最大桁数は 18 桁です。DB 接続命令で 19 桁以上の値を DB に書き込むとエラーとなります。

*4. 文字列の終端に NULL が付くため、STRING のデータ型で使用するバイト数は、DB 側データ型のバイト数+1 を設定する必要があります。
STRING のデータ型で使用するバイト数は、DB 側データ型、文字コードに合わせて適切な値を設定する必要があります。NJ/NX シリーズでは、文字列を UTF-8 で扱います。半角英数字 1 文字は 1 バイト、マルチバイト文字列は複数バイトを使用します。日本語文字は 3 バイトが目安です。

- PostgreSQL の場合

データ型の種類	DB 側のデータ型	NJ/NX 側のデータ型
整数	boolean	BOOL
	smallint	INT
	integer	DINT
	bigint	LINT TIME
	serial	UDINT
	bigserial	ULINT
固定小数点*1		*2
	decimal(3) decimal (5) decimal (10) decimal (20) decimal (3) decimal (5) decimal (10) decimal (20)	SINT INT DINT LINT USINT UINT UDINT ULINT
		*2
	numeric (3) numeric (5) numeric (10) numeric (20) numeric (3) numeric (5) numeric (10) numeric (20)	SINT INT DINT LINT USINT UINT UDINT ULINT
実数	real	REAL
	double precision	LREAL
日付	timestamp［(p)］[without time zone]	DATE_AND_TIME
	timestamp［(p)］with time zone	DATE_AND_TIME
	date	DATE
	time［(p)］[without time zone]	TIME_OF_DAY
	time［(p)］with time zone	TIME_OF_DAY
文字列	character(n)、char(n)	STRING*3
	character varying(n)、varchar(n)	STRING*3
	text	STRING*3

データ型の種類	DB 側のデータ型	NJ/NX 側のデータ型
その他	bit［(n)］	なし
	bit varying［(n)］	なし
	box	なし
	bytea	なし
	cidr	なし
	circle	なし
	inet	なし
	interval［fields］［(p)］	なし
	line	なし
	lseg	なし
	macaddr	なし
	money	なし
	path	なし
	point	なし
	polygon	なし
	tsquery	なし
	tsvector	なし
	txid_snapshot	なし
	uuid	なし
	xml	なし

*1. decimal(p［,s］)、numeric(p［,s］)の小数点以下桁数(s)は、省略形式で記述しています。省略形式を使用した場合は、小数点以下桁数(s)は 0 です。もし省略せずに、小数点以下桁数(s)に 1 以上の数値を指定した場合、整数部のみが対象となります。

*2. 扱う範囲が異なるので、上記の対応どおりでもデータの桁あふれが生じるケースがあります。
例：DB 側のデータ型が DECIMAL(3)、NJ/NX 側のデータ型が USINT の場合
- DECIMAL(3)の範囲：0～999
- USINT の範囲：0～255

*3. 文字列の終端に NULL が付くため、STRING のデータ型で使用するバイト数は、DB 側データ型のバイト数+1 を設定する必要があります。
STRING のデータ型で使用するバイト数は、DB 側データ型、文字コードに合わせて適切な値を設定する必要があります。NJ/NX シリーズでは、文字列を UTF-8 で扱います。半角英数字 1 文字は 1 バイト、マルチバイト文字列は複数バイトを使用します。日本語文字は 3 バイトが目安です。

使用上の注意

- NJ/NX 側のデータ型に、対応表に記載していないデータ型を使用した場合、データを正しく変換できない場合があります。
- DB 接続命令で DB から値を読み出す際、以下の原因によって型変換ができず、命令エラー（SQL 実行エラー）が発生することがあります。
  a）値が NULL のカラムを含むレコードを取得した場合
  b）対応表に記載していないデータ型の組み合わせの場合

3-2-3 DB にアクセスするための構造体データ型の作成方法

DB にアクセスするための構造体データ型は、以下のいずれかの方法で作成します。
- データ型エディタへの直接入力
- データ型エディタへの Microsoft Excel からの貼り付け

以下に、操作の概要を示します。操作の詳細は、『Sysmac Studio Version 1 オペレーションマニュアル（SBCA-470)』を参照してください。

データ型エディタへの直接入力

1 **マルチビューエクスプローラの［プログラミング］－［データ］－［データ型］をダブルクリックします。**

2 **データ型エディタのサイドタブ［構造体型］をクリックします。**

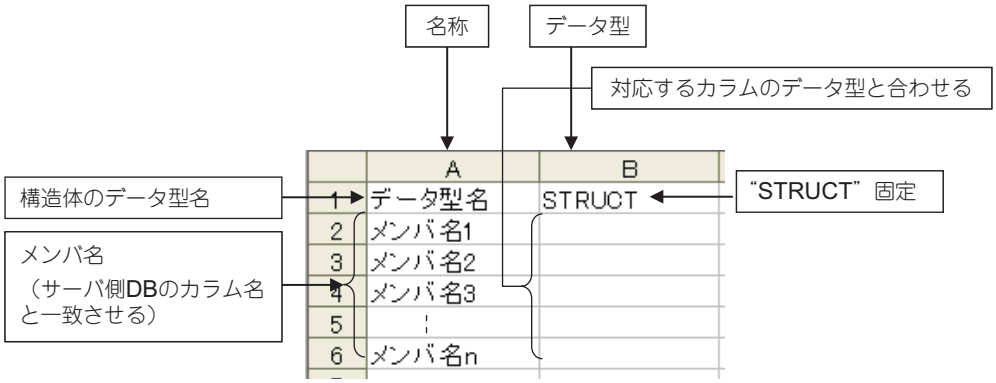
3 **構造体型のデータ型エディタで、データ型名を入力します。**

4 **右クリックして［メンバ新規作成］を選択し、各メンバ名とそのデータ型を入力します。**

データ型エディタへの Microsoft Excel からの貼り付け

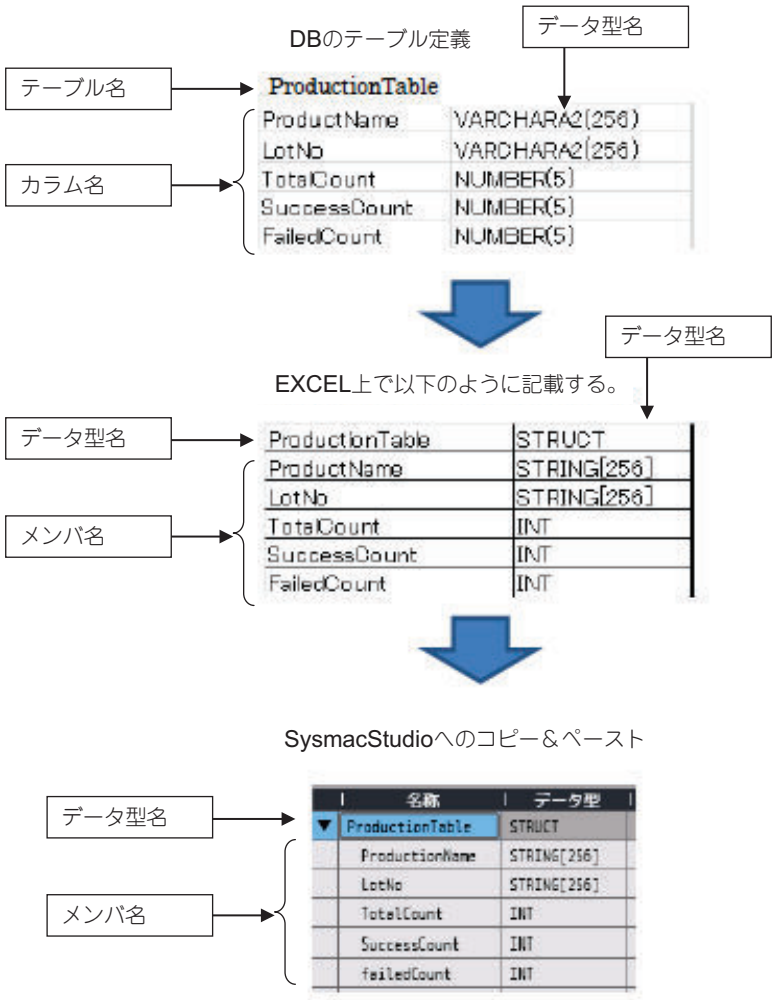
1 **Microsoft Excel の 2 つの列を、左から、名称、データ型とします。**

2 **上から、構造体データ型名、各メンバ名とします。**
データ型は、構造体のため、“STRUCT”固定です。



3 **Microsoft Excel 上で、名称、データ型の列にあるデータ領域をコピーします。**

4 **構造体型のデータ型エディタの名称、データ型の列に、貼り付けます。**
例）



使用上の注意

以下の場合は、構造体型の“データ型”エディタへの貼り付けを実行できません。
- 任意の構造体メンバが選択されている場合
- 何も選択されていない場合

任意の構造体型のデータ型のみを選択した状態で、貼り付けを実行してください。

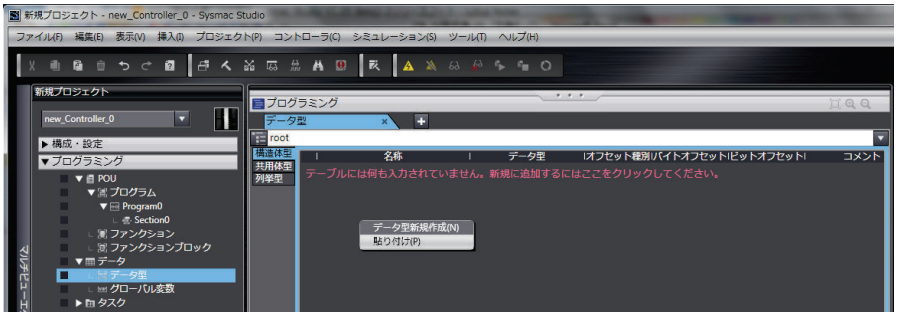
参考

DB の開発ツールのテーブル定義データを再利用して、DB にアクセスするための構造体データ型を作成することも可能です。
以下の手順によります。

1）DB の開発ツールのテーブル定義データのカラム名とデータ型をコピーします。
2）Microsoft Excel などの表計算ソフトのシート上にカラム名とデータ型の列を作成し、そこに貼り付けます。
3）各カラムのデータ型を、NJ/NX シリーズ CPU ユニットの変数のデータ型に変更します。
4）構造体のデータ型名の行を、カラム名とデータ型の上に挿入します。
5）構造体のデータ型名の行のデータ型に“STRUCT”と入力します。
6）カラム名とデータ型の下の列の情報を、以下のようにコピーします。

	A	B	C
1			
2	カラム名	データ型	
3	MyTable	STRUCT	←テーブル名
4	MYCOLUMN1	INT	←カラム1
5	MYCOLUMN2	STRING[20]	←カラム2
6	MYCOLUMN3	STRING[20]	←カラム3
7			← ・・・

7）SysmacStudio の構造体型の“データ型”エディタ上で右クリックし“貼り付け”を選択します。



以下のように構造体のデータ型が作成されます。

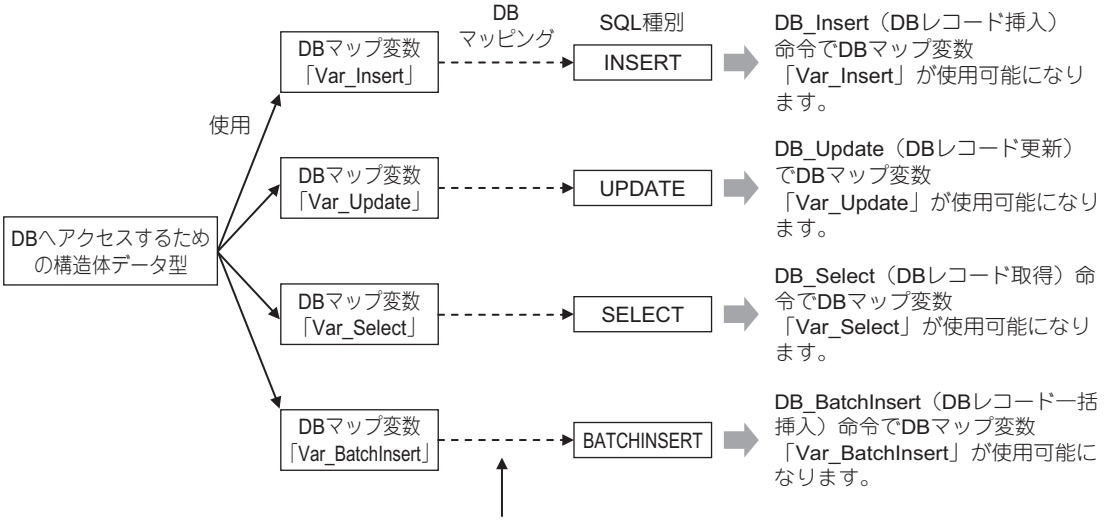


3-3 DB マップ変数の作成

DB にアクセスするためのユーザ定義の構造体データ型を作成した後、それをデータ型として指定した変数を作成します。作成した変数を“DB マップ変数”と呼びます。
ここでは、その DB マップ変数の仕様および作成方法を説明します。

3-3-1 DB マップ変数、および DB へのマッピング

DB マップ変数は、データ型として、DB にアクセスするための構造体データ型を使用した変数です。
DB マップ変数を、SQL 種別ごとに、DB_CreateMapping（DB マップ生成）命令を実行することで DB マッピング*1 します。
DB マッピングを実行することにより、レコード処理命令にて、DB マップ変数を使用したレコード処理やストアドプロシージャ実行が可能となります。*2



*1. “DB マッピング”とは、DB マップ変数の各メンバなどを DB のテーブルの各カラムやストアドプロシージャの引数、戻り値、結果セットに関連付けることを指します。レコード処理ごとにマッピングを実施する必要があります。
*2. ストアドプロシージャ呼出における DB マッピングは、DB_AttachProcedure（ストアドプロシージャ ハンドル生成）命令を実行することで行います。詳細は、「5-3 ストアドプロシージャ呼出機能 (P.5-15)」を参照してください。

1 つの DB コネクションに対して、複数の DB マップ変数を DB マッピングすることができます。
DB のテーブルの一部のカラムをメンバに持つ構造体データ型を作成した場合、レコード処理を実行したときの動作を以下に示します。

レコード処理	実行時の動作
レコード挿入（INSERT）	マッピングされている DB のカラムに、値を設定したレコードを書き込みます。 マッピングされていない DB のカラムには NULL が入ります。DB 側で NULL を許容する設定が必要です。
レコード更新（UPDATE）	マッピングされている DB のカラムのみ、値が更新されます。 マッピングされていない DB のカラムは、値はそのまま保持されます。
レコード取得（SELECT）	マッピングされている DB のカラムの値のみ、取得できます。 NULL が格納されていない DB のカラムのみをマッピングする必要があります。

レコード処理	実行時の動作
レコード一括挿入（BATCHINSERT）	マッピングされている DB のカラムに、値を設定した複数のレコードを書き込みます。 複数のレコードのうちの 1 つのレコードに対する動作は、レコード挿入（INSERT）時の実行時の動作と同じです。マッピングされていない DB のカラムには NULL が入ります。

使用上の注意

DB_Select（DB レコード取得）命令実行時、値が NULL のカラムを含むレコードを取得した場合、DB_Select（DB レコード取得）命令は異常終了（SQL 実行エラー）します。

参考

DB_CreateMapping（DB マップ生成）命令による DB マップ変数のマッピング時には、構造体メンバと DB のテーブルのカラムが一致しているかチェックしません。レコード処理命令の実行結果がエラーとなります。

3-3-2 DB マップ変数の登録および属性

DB マップ変数として指定可能な種類および属性は、以下のとおりです。

項目		指定可能な種類	制約
変数の登録		グローバル変数 プログラムのローカル変数 ファンクションブロックのローカル変数	ファンクションのローカル変数は指定不可*1
属性	変数名	任意	変数名などのプログラム関連の名称の制限については、『NJ/NX シリーズ CPU ユニット　ユーザーズマニュアル　ソフトウェア編（SBCA-467）』を参照してください。
	データ型	DB にアクセスするための構造体データ型	「3-2 構造体データ型の作成 (P.3-3)」を参照してください。
	AT（割付先）指定	任意	
	保持	任意	
	初期値	任意	
	コンスタント	任意	SELECT 用には指定できません。DB_Select（DB レコード取得）命令でコンパイルエラーとなります。
	ネットワーク公開	任意	
	エッジ	指定できない	
配列指定		指定できる（SELECT、BATCHINSERT 用）	INSERT 用、UPDATE 用には指定できません。 DB_CreateMapping（DB マップ生成）命令の命令エラーとなります。 詳細は、「3-3-3 DB マップ変数の制約事項 (P.3-17)」を参照してください。

*1. DB_CreateMapping（DB マップ生成）命令は FB 型のため、ファンクション POU 内で使用することはできません。

使用上の注意

ファンクションブロック内で DB 接続命令を使用し、ファンクションブロックの入出力変数を DB マップ変数として指定した場合、DB 接続命令実行時に DB マップ変数の各メンバの値はデータ型に従って、ユーザ指定値で動作します。ファンクションブロックの入出力変数を DB マップ変数として指定しないでください。
入出力変数を DB 接続命令で使用する必要がある場合は、ファンクションブロックの内部変数を DB マップ変数として指定し、DB_Insert／DB_Update／DB_BatchInsert 命令実行前、または DB_Select 命令実行後に、入出力変数と内部変数間で MOVE 命令などでデータ転送してください。

3-3-3 DB マップ変数の制約事項

DB マップ変数の制約事項を以下に示します。

SQL 種別に応じた、DB マップ変数のデータ型の配列指定の可否

DB マップ変数のデータ型における構造体を配列要素とした指定の可否は、SQL 種別によります。詳細を以下に示します。

SQL 種別	DB マップ変数で構造体を配列要素とした指定の可否
INSERT	不可
UPDATE	
SELECT	可能
BATCHINSERT	

DB マップ変数のマッピング不可の条件

以下の場合は、DB マップ変数のマッピングができません。DB_CreateMapping（DB マップ生成）命令の実行結果がエラーとなります。

- DB マップ変数のデータ型が構造体型でない場合
- DB マップ変数の構造体のメンバに派生データ型が含まれている場合
- INSERT 用または UPDATE 用の DB マップ変数に、構造体型の配列指定変数を指定した場合
- BATCHINSERT 用の DB マップ変数に構造体変数を指定した場合

レコード処理命令実行時にエラーとなる条件

以下の場合は、DB_CreateMapping（DB マップ生成）命令による DB マップ変数のマッピング時にはエラーが検出されません。レコード処理命令の実行結果がエラーとなります。

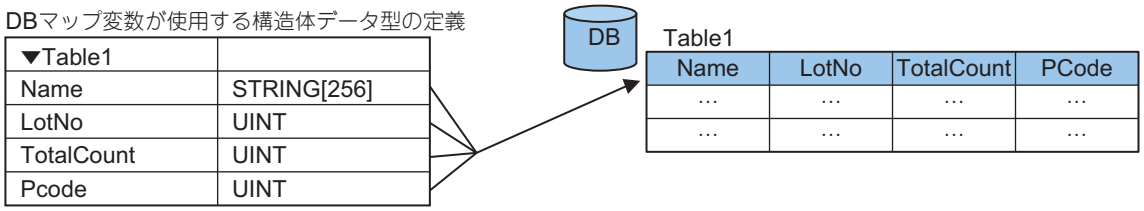
- DB に接続できない場合
- DB に、指定したテーブルが存在しない場合
- DB マップ変数のメンバ名と同じ名前のカラムが DB のテーブルに存在しない場合
- DB マップ変数のメンバのデータ型と、DB のカラムのデータ型が対応していない場合

3-4 DB マップ変数の DB へのマッピング

レコード処理命令を実行する前に、SQL 種別ごとに DB マップ変数を DB にマッピングすることが必要です。
ここでは、レコード処理命令のための DB マッピングの方法、DB マッピングの解除方法と制約事項を説明します。

3-4-1 DB マップ生成命令による DB へのマッピング

DB マップ変数の DB へのマッピングは、DB_CreateMapping（DB マップ生成）命令を実行します。
DB_CreateMapping（DB マップ生成）命令で、“テーブル名”と“DB マップ変数”と“SQL 種別”を指定します。
これにより、DB マップ変数を SQL 種別ごとに DB へマッピングできます。
詳細は、「DB_CreateMapping（DB マップ生成）（P.7-11）」を参照してください。



3-4-2 DB マップ変数のマッピングの解除

DB マップ変数のマッピングは、以下の操作により自動的に解除されます。

- DB コネクションを切断したとき
- DB 接続サービスを停止したとき*1
- DB 接続サービスを終了したとき
- DB_CreateMapping（DB マップ生成）命令によって既にマッピングされている DB マップ変数を、別のテーブルまたは SQL 種別でマッピングしたとき

*1. DB 接続サービスの停止についての詳細は、「4-1-3 DB 接続サービスの開始不可および停止の条件（P.4-4）」を参照してください。

使用上の注意

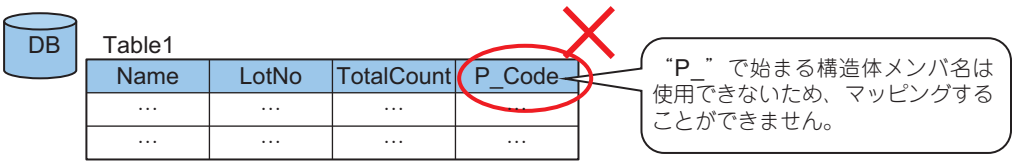
DB へのマッピングは、DB コネクションを切断すると自動的に解除されます。
そのため、DB へのマッピングを実行する前に、DB_Connect（DB コネクション接続）命令を実行するようにプログラミングしてください。

3-4-3 DB マッピングの制約

DB マッピングには以下の制約があります。

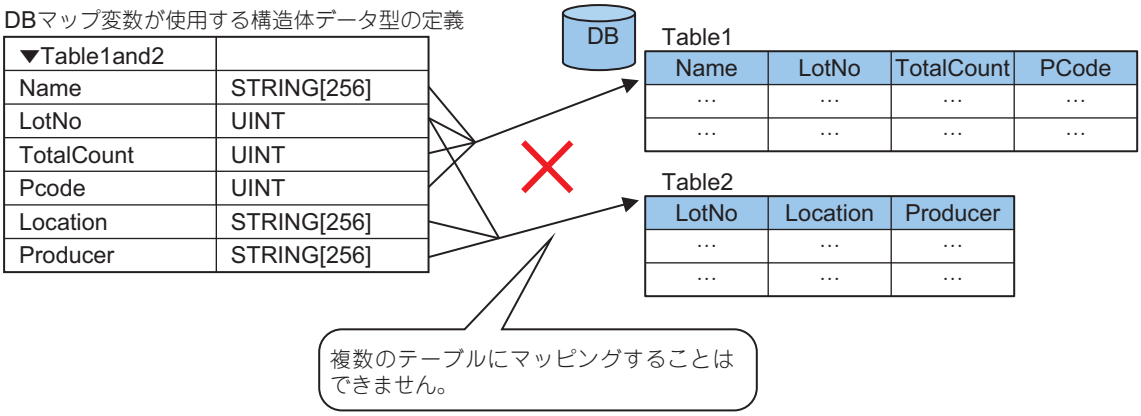
- DB 側のテーブルのカラム名の制約：

DB のテーブルのカラム名に、構造体メンバ名として指定できない文字を使用している場合は、マッピングすることができません。サーバ側 DB のテーブルのカラム名を変更する必要があります。
例）
サーバ側 DB のテーブルのカラム名が「P_Code」の場合

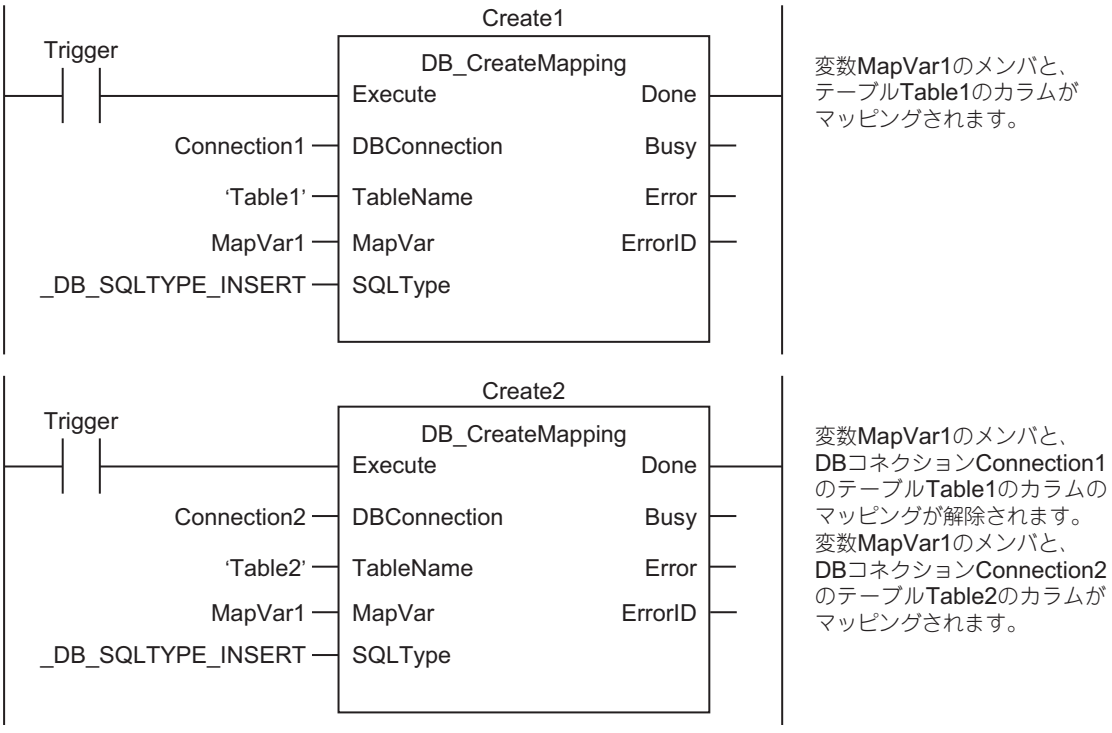


構造体メンバ名として指定できない文字については、「3-2-2 DB にアクセスするための構造体データ型の仕様 (P.3-3)」の使用上の注意を参照してください。

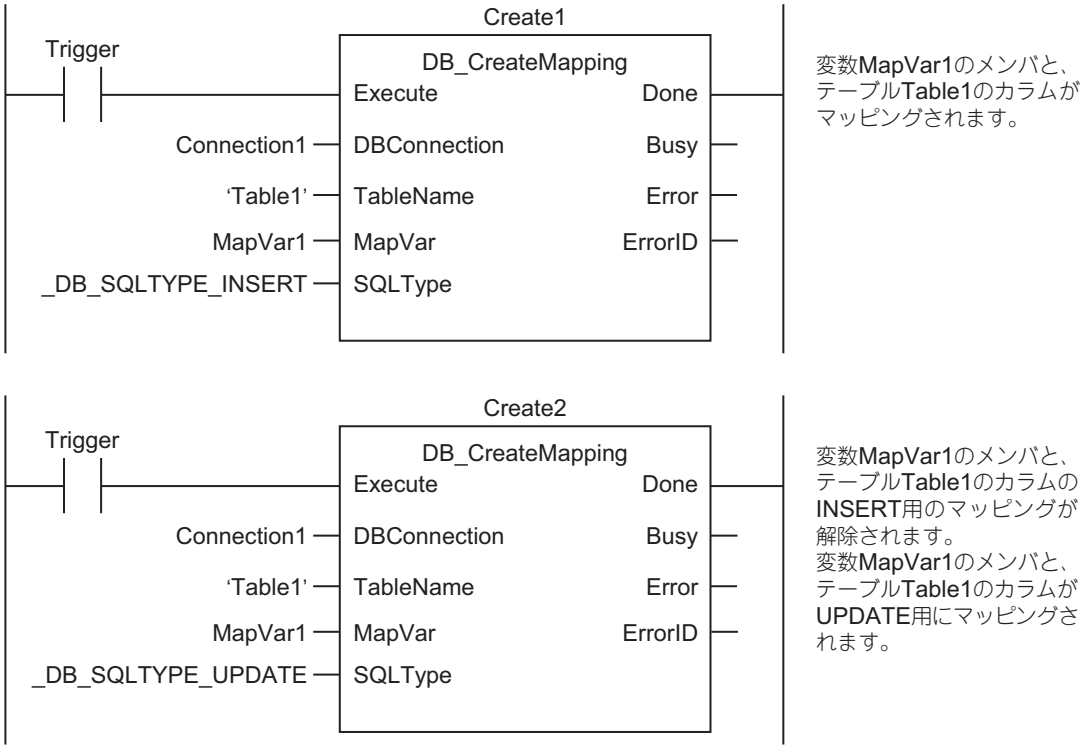
• DB 側の複数のテーブルへのマッピングの制約：
同一の DB マップ変数のメンバを、それぞれ別のテーブルのカラムにマッピングすることはできません。
例）



• DB 側の複数のテーブルへのマッピングの制約：
同一の DB マップ変数を複数のテーブルにマッピングすることはできません。
同一の DB マップ変数を複数のテーブルにマッピングするように、複数の DB_CreateMapping（DB マップ生成）命令を実行した場合、最後に実行した DB_CreateMapping（DB マップ生成）命令のマッピングが有効となります。
例）



• 複数の SQL 種別でのマッピングの制約：
同一の DB マップ変数を複数の SQL 種別にマッピングすることはできません。
同一の DB マップ変数を複数の SQL 種別にマッピングするように、複数の DB_CreateMapping（DB マップ生成）命令を実行した場合、最後に実行した DB_CreateMapping（DB マップ生成）命令のマッピングが有効となります。
例）



• マッピング可能な DB マップ変数の最大数
すべての DB コネクションでマッピング可能な DB マップ変数の最大数は、接続対象のデータベース種別によって異なります。マッピング可能な DB マップ変数の最大数については、「1-2 DB 接続サービスの仕様としくみ (P.1-4)」を参照してください。上限値を超えた場合は、DB_CreateMapping（DB マップ生成）命令実行時に命令エラー（データ容量超過）が発生します。
ただし、関連付けする DB マップ変数の数が上限値に満たない場合でも、すべての DB コネクションで、DB マップ変数のデータ型として使用する構造体定義のメンバ総数が 10000 個を超える場合は、命令エラー（データ容量超過）が発生します。

• DB マップ変数の定義
レコード処理命令を、DB_CreateMapping（DB マップ生成）命令と異なる POU インスタンスで実行する場合は、DB マップ変数はグローバル変数とする必要があります。

3-5 DB 接続命令のプログラミング／転送

ここでは、DB 接続サービスのプログラミングの概要、DB 接続命令一覧、システム定義変数などについて、説明します。
実際のプログラム例については、「第 7 章 DB 接続命令 (P.7-1)」の各サンプルプログラムを参照してください。

3-5-1 DB 接続サービスのプログラミングの概要

DB 接続サービスのプログラミングは、以下の順で行います。

1 **Sysmac Studio のプログラムエディタにて、ツールボックスの“DB 接続”の命令カテゴリから、DB 接続命令を選択します。DB 接続命令を、以下の順で記述します。*1**

1) 初期処理

- 命令によって DB 接続サービスを開始する時*2、DB_ControlService（DB 接続サービスの制御）命令を記述します。
- DB_Connect（DB コネクション接続）命令を記述します。
- DB_CreateMapping（DB マップ生成）命令もしくは DB_AttachProcedure（DB ストアドプロシージャ ハンドル生成）命令を記述します。

2) 運転中処理*3

- レコード処理命令、ストアドプロシージャ命令などを記述します。

3) 終了処理

- DB_Close（DB コネクション切断）命令を記述します。

4) 電源断処理*4

- DB_Shutdown（DB 接続サービスのシャットダウン）命令を記述します。

*1. DB 接続命令の一覧については、「3-5-3 DB 接続命令一覧 (P.3-22)」を参照してください。
*2. DB 接続サービスを［自動開始する］に設定している場合、CPU ユニットの動作モードをプログラムモードから運転モードに切り替えたとき、DB 接続サービスが自動的に開始されます。
*3. レコード処理命令、ストアドプロシージャ命令などを継続して実行する場合は、(2) 運転中処理を繰り返します。
*4. システムの電源を OFF にする場合は、DB_Shutdown 命令を実行してください。
DB_Shutdown 命令を実行せずに電源を OFF にした場合は、稼働ログファイルが壊れる、または内容が消失する場合があります。

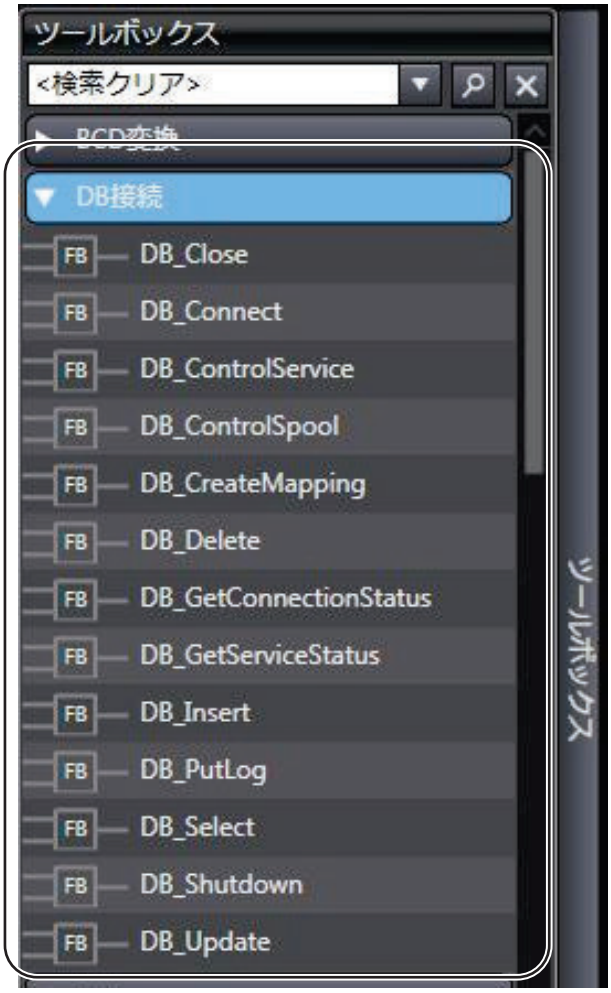
2 **DB 接続サービスのステータスは、システム定義変数で参照します。**
ステータスとして、稼働モードで動作中、テストモードで動作中、アイドル中、異常停止中、終了を確認できます。

3 **DB 接続設定とユーザプログラムを転送します。**
NJ/NX シリーズ CPU ユニットに DB 接続設定およびユーザプログラムを転送します。

4 **コントローラの電源を再投入します。**
接続するデータベース種別を変更した場合、コントローラの電源を再投入します。

3-5-2 Sysmac Studio での DB 接続命令の表示

Sysmac Studio のツールボックスの“DB 接続”の命令カテゴリに、DB 接続命令が表示されます。



3-5-3 DB 接続命令一覧

DB 接続命令の一覧を以下に示します。
DB 接続命令の詳細は、「第 7 章 DB 接続命令 (P.7-1)」を参照してください。

分類	命令	命令名	機能
接続／切断	DB_Connect	DB コネクション接続	指定した DB に接続します。
	DB_Close	DB コネクション切断	DB_Connect 命令で接続した DB との DB コネクションを切断します。
マッピング	DB_CreateMapping	DB マップ生成	DB のテーブルと DB マップ変数を関連付け（マッピング）します。
レコード処理	DB_Insert	DB レコード挿入	DB のテーブルへ DB マップ変数の値を、1 つのレコードとして挿入します。
	DB_Update	DB レコード更新	テーブルのレコードの値を、DB マップ変数の値で更新します。
	DB_Select	DB レコード取得	DB の 1 つのテーブルからレコードを抽出し、DB マップ変数に取得します。
	DB_Delete	DB レコード削除	指定したテーブルから条件に一致するレコードを削除します。
	DB_BatchInsert	DB レコード一括挿入	DB のテーブルへ DB マップ変数の各配列要素の値を 1 つのレコードとして一括挿入します。

分類	命令	命令名	機能
ストアドプロシージャ呼出*1	DB_AttachProcedure	DB ストアドプロシージャハンドル生成	DB のストアドプロシージャを呼び出すための準備処理を行います。
	DB_ExecuteProcedure	DB ストアドプロシージャ実行	DB_AttachProcedure で取得したプロシージャハンドルを使って、ストアドプロシージャを呼び出します。
	DB_DetachProcedure	DB ストアドプロシージャハンドル解放	DB_AttachProcedure で取得したプロシージャハンドルを解放します。
その他	DB_ControlService	DB 接続サービスの制御	DB 接続サービスの開始／停止、またはデバッグログの記録開始／終了を行います。
	DB_GetServiceStatus	DB 接続サービスの状態取得	DB 接続サービスの現在の状態を取得します。
	DB_GetConnectionStatus	DB コネクションの状態取得	DB コネクションの状態を取得します。
	DB_ControlSpool	スプールの再送・クリア	DB_Insert（DB レコード挿入）命令または DB_Update（DB レコード更新）命令でスプールした SQL 文を、再送またはクリアします。
	DB_PutLog	稼働ログ記録	実行ログまたはデバッグログにユーザ指定のログを記録します。
シャットダウン	DB_Shutdown	DB 接続サービスのシャットダウン	DB 接続サービスに対して、終了を要求し、DB 接続サービスをシャットダウンします。*2

*1. ストアドプロシージャに関する詳細については、「5-3 ストアドプロシージャ呼出機能 (P.5-15)」を参照してください。

*2. システムの電源を OFF にする場合は、DB_Shutdown 命令を実行してください。DB_Shutdown 命令を実行せずに電源を OFF にした場合は、稼働ログファイルが壊れる、または内容が消失する場合があります。

各命令の詳細、サンプルプログラムは、「第 7 章 DB 接続命令 (P.7-1)」を参照してください。

3-5-4 システム定義変数

DB 接続サービスでは、以下のシステム定義変数を使用できます。ユーザプログラム内でシステム定義変数の値を参照することにより、DB 接続サービスの稼働状態や、DB 接続サービスのバージョンなどに応じた処理を行うことができます。

変数名 メンバ名	データ型	名称	機能	初期値
_DBC_Status	_sDBC_STATUS	DB 接続サービスステータス	DB 接続サービスの稼働状態を示します。 DB 接続サービスの稼働状態の詳細は、「4-3-1 DB 接続サービスの稼働状態 (P.4-7)」を参照してください。	
Run	BOOL	動作中フラグ	DB 接続サービスが稼働モードまたはテストモードで動作しているとき、TRUE となります。	FALSE
Test	BOOL	テストモード	DB 接続サービスがテストモードで動作しているとき、TRUE となります。	FALSE
Idle	BOOL	アイドル	DB 接続サービスの稼働状態がアイドルのとき、TRUE となります。	FALSE
Error	BOOL	異常停止中フラグ	DB 接続サービスの稼働状態が異常停止中のとき、TRUE となります。	FALSE

変数名 メンバ名	データ型	名称	機能	初期値
Shutdown	BOOL	終了	DB 接続サービスの稼働状態が終了状態のとき、TRUE となります。	FALSE
_DBC_Version*1	ARRAY[0..1] OF USINT	DB 接続サービスバージョン	DB 接続サービスのバージョンが格納されます。*2 要素番号 0 にバージョンの整数部が格納されます。 要素番号 1 にバージョンの小数部が格納されます。	
_JRE_Version*1	ARRAY[0..1] OF USINT	JRE バージョン	JRE のバージョンが格納されます。*3 要素番号 0 にバージョンの整数部が格納されます。 要素番号 1 にバージョンの小数部が格納されます。	
_DBC_Unused*4	BOOL	DB 接続入力変数省略	ストアドプロシージャの引数、戻り値、または、結果セットのいずれかが存在しないときに、DB_AttachProcedure 命令の入力変数を省略するために設定します。 「_DBC_Unused」の値が TRUE、FALSE のどちらでも、命令の実行結果は同じです。	TRUE

*1. DB 接続サービス Ver.1.03 以降で使用可能です。
*2. 例 1）DB 接続サービス Ver.1.00 のとき、要素番号 0 に 1、要素番号 1 に 0 が格納されます。
例 2）DB 接続サービス Ver.1.10 のとき、要素番号 0 に 1、要素番号 1 に 10 が格納されます。
*3. 例 1）JRE バージョン Ver.1.00 のとき、要素番号 0 に 1、要素番号 1 に 0 が格納されます。
例 2）JRE バージョン Ver.1.10 のとき、要素番号 0 に 1、要素番号 1 に 10 が格納されます。
*4. DB 接続サービス Ver.2.00 以降で使用可能です。

3-5-5 DB 接続命令のシミュレーションデバッグ

Sysmac Studio のシミュレーション機能により、ユーザプログラムの動作確認を行うことができます。
シミュレーション実行中に、DB 接続命令は、以下の動作をします。

- DB_Connect、DB_Close、DB_Insert、DB_BatchInsert など結果取得を伴わない命令は、正常終了します。
- DB_Select、DB_ExecuteProcedure など結果取得を伴う命令は、正常終了し、取得データなしとして動作します。

3-5-6 DB 接続設定とユーザプログラムの転送

Sysmac Studio の同期機能を使用して、NJ/NX シリーズ CPU ユニットに、DB 接続設定およびユーザプログラムを転送します。
同期ウィンドウで追加表示される DB 接続設定の照合の単位は以下のとおりです。

同期データ名	階層	個数	詳細照合	備考
上位接続	2	1	なし	
DB 接続	3	1	なし	
DB 接続サービス設定	4	1	なし	
DB コネクション設定	4	1	なし	

DB 接続設定の反映タイミングは、DB 接続サービス開始時です。

使用上の注意

- Sysmac Studio から各操作を実行したとき、操作の失敗または通信異常が発生する場合は、以下を実施した後に各操作を再実行してください。
  a）ケーブル接続を確認
  b）通信設定を確認
  c）通信設定のレスポンス監視時間を延ばす
  d）システムサービスの実行時間割合を増やす
  e）DB 接続サービスの稼働状態が“初期処理中”、“異常停止中”、“終了”以外であることを確認
  DB 接続サービスの稼働状態の詳細は、「4-3-1 DB 接続サービスの稼働状態 (P.4-7)」を参照してください。
- Sysmac Studio がオンライン接続できない場合は、『NJ/NX シリーズ トラブルシューティングマニュアル（SBCA-469)』を参照してください。

3-6 設計、立ち上げ、運用時の各デバッグ

設計、立ち上げ、運用のフェーズ別、および実機環境別に、以下に示すデバッグを行います。

3-6-1 設計時

設計時のデバッグ方法を示します。

実機の環境		デバッグ方法	
CPU ユニット	サーバ側 DB	確認内容	実施内容
存在する	存在しない、または、接続しない	実機 CPU ユニットでの DB 接続命令実行の確認	• 動作モードが“テストモード”での DB 接続サービス開始 • DB 接続命令実行 注：“テストモード”では、SQL 文は実際には送信せず、送信したとして正常終了します。 • 稼働ログ（実行ログ、デバッグログ）の確認

3-6-2 立ち上げ時

立ち上げ時のデバッグ方法を示します。

実機の環境		デバッグ方法	
CPU ユニット	サーバ側 DB	確認内容	実施内容
存在する	接続する	DB 接続	• 動作モードが“稼働モード”での DB 接続サービス開始 • Sysmac Studio からの、DB 接続サービスおよび各コネクションの状態の確認
		DB 読み書き、そのタイミングの確認	• DB 接続命令実行 • 稼働ログ（実行ログ、デバッグログ、SQL 実行失敗ログ）の確認 （DB 接続、SQL 文の実行、レスポンス確認含む）

3-6-3 運用時

運用時のトラブルシュート方法を示します。

実機の環境		トラブルシュート方法	
CPU ユニット	サーバ側 DB	確認内容	実施内容
存在する	接続する	定常的な確認	• イベントログの確認 • 稼働ログ（実行ログ、SQL 実行失敗ログ）の確認 • Sysmac Studio からの、DB 接続サービスおよび各コネクションの状態の確認 • DB 接続命令による、DB 接続サービスおよび各コネクションの状態の確認

4

DB 接続サービスの開始／コネクションの接続と状態確認

この章では、DB 接続サービスの開始／停止方法、各 DB コネクションの接続／切断と、DB 接続サービスおよび各コネクションの状態の確認方法について説明しています。

4-1　DB 接続サービスの動作モードと開始／停止 4-2
4-1-1　DB 接続サービスの動作モード 4-2
4-1-2　DB 接続サービスを開始／停止する方法 4-2
4-1-3　DB 接続サービスの開始不可および停止の条件 4-4
4-1-4　DB 接続サービスの動作モードの変更 4-5
4-2　各 DB コネクションの接続／切断 4-6
4-3　DB 接続サービスおよび各コネクションの状態の確認 4-7
4-3-1　DB 接続サービスの稼働状態 4-7
4-3-2　DB 接続サービスの状態の確認 4-8
4-3-3　DB コネクションの接続状態 4-10
4-3-4　各 DB コネクションの状態の確認 4-11

4-1 DB 接続サービスの動作モードと開始／停止

ここでは、DB 接続サービスの動作モードおよび開始／停止方法について説明します。

4-1-1 DB 接続サービスの動作モード

DB 接続サービスには、DB に実際にアクセスするかどうかに応じて、“稼働モード”と“テストモード”の 2 つの動作モードがあります。
以下に、DB 接続サービスの動作モードの内容と、その用途を示します。

DB 接続サービスの動作モード

サーバ側 DB に接続せずに DB 接続サービスの動作をテストするか、サーバ側 DB に接続して本運転または試運転をするかに応じて、以下の 2 つの動作モードを切り替えます。

動作モード	内容	用途	選択する環境条件の例
テストモード	• DB 接続命令を実行しても、DB への SQL 文の送信を実行しません。 • DB 接続サービス命令は正常終了します。ただし、DB から読み出す命令は、読み出し先の DB マップ変数には何も出力しません。 • スプール機能は動作しません。	DB に接続しない状態で、DB 接続命令を使用したユーザプログラムの動作確認をするとき	DB が存在しない場合、 または 存在するが接続しない場合
稼働モード	• DB 接続命令を実行すると、DB への SQL 文の送信を実行します。 • スプール機能が動作します。	DB に接続した状態で、システムを本運転または試運転するとき	DB が存在し接続する場合

4-1-2 DB 接続サービスを開始／停止する方法

DB 接続サービスを開始する方法、および停止する方法には、以下の 3 つがあります。
- 運転時のサービス開始
- Sysmac Studio からのオンライン操作による開始／停止
- DB_ControlService（DB 接続サービスの制御）命令実行

DB 接続サービス動作中に、DB 接続サービスの動作モードを切り替えることはできません。
いったん、DB 接続サービスを停止してから、動作モードを指定して DB 接続サービスを開始してください。

運転時のサービス開始

Sysmac Studio の［構成・設定］－［上位接続設定］－［DB 接続］－［DB 接続サービス設定］の［サービス設定］にて、“運転時のサービス開始”を、“自動開始する（テストモード）”または“自動開始する（稼働モード）”に設定します。
［運転時のサービス開始］のデフォルトは、形 NX502-1□00 以外の場合は［自動開始する（稼働モード）］、形 NX502-1□00 の場合は［使用しない］となります。

CPU ユニットの動作モードをプログラムモードから運転モードに切り替えた時、DB 接続サービスが自動的に開始されます。

使用上の注意

DB 接続サービスを“自動開始する”とした場合でも、DB 接続サービスの起動処理が完了するまでは、DB 接続命令は実行できません。命令実行エラーとなります。
そのため、システム定義変数「_DBC_Status.Run」（DB 接続サービスステータスの動作中フラグ）で DB 接続サービスの稼働状態が“動作中”であることを確認した上で、DB 接続命令を実行するようにユーザプログラムを作成してください。
ユーザプログラム例：

```
IF _DBC_Status.Run = FALSE THEN
        RETURN;         (* DB 接続サービスが動作中でないので処理終了 *)
END_IF;
(* DB 接続命令実行 *)
(以下省略)
```

Sysmac Studio からのオンライン操作による開始／停止

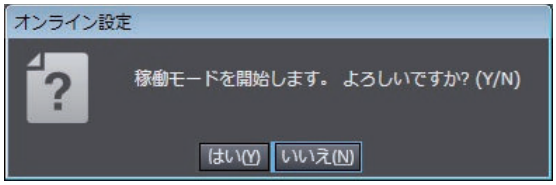
1 **NJ/NX シリーズ CPU ユニットにオンライン接続している状態で、Sysmac Studio の［構成・設定］－［上位接続設定］－［DB 接続］の下の［DB 接続サービス設定］で、右クリックして［オンライン設定］を選択します。**
以下の［オンライン設定］画面が表示されます。



以下の各ボタン操作により、DB 接続サービスに対する開始、停止を行うことができます。

分類	項目	ボタン	動作
サービス	開始・停止	［稼働モード開始］	DB 接続サービスを稼働モードで開始します。
		［テストモード開始］	DB 接続サービスをテストモードで開始します。
		［停止］	DB 接続サービスを停止します。

2 **DB 接続サービスを開始するとき：**
［稼働モード開始］または［テストモード開始］ボタンをクリックします。
DB 接続サービスを停止するとき：
［停止］ボタンをクリックします。
以下のような確認メッセージが表示されます。以下は DB 接続サービスを稼働モードで開始するときの画面例です。



3 **［はい（Y)］ボタンをクリックします。**

（注）CPU ユニットの動作モードにかかわらず、DB 接続サービスを開始または停止することが可能です。

参考

［シャットダウン］ボタンをクリックした場合、DB 接続サービス自体をシャットダウンさせることができます。シャットダウン機能については、「5-5 DB 接続サービスシャットダウン機能 (P.5-25)」を参照してください。

DB_ControlService（DB 接続サービスの制御）命令の実行

DB_ControlService（DB 接続サービスの制御）命令の入力変数 Cmd に、以下のいずれかのコマンドを指定して実行します。

- 稼働モードで開始
- テストモードで開始
- 停止

DB_ControlService（DB 接続サービスの制御）命令の詳細は、「第 7 章 DB 接続命令 (P.7-1)」を参照してください。

4-1-3 DB 接続サービスの開始不可および停止の条件

DB 接続サービスが開始できない条件、停止する条件は、以下のとおりです。

● DB 接続サービスが開始できない条件

DB 接続サービスは、以下のいずれかの場合に、開始できません。

- DB 接続サービス設定が壊れているとき
- DB 接続サービスの稼働状態が“初期処理中”であるとき
- DB 接続サービスの稼働状態が“終了”であるとき
- いずれかのコネクションのサーバ証明書のフォーマットが破壊されている、または正しくないとき

● DB 接続サービスが停止する条件

DB 接続サービスは、以下のいずれかの場合に、停止します。

- DB_ControlService（DB 接続サービスの制御）命令または Sysmac Studio から、DB 接続サービス停止を指示したとき
- CPU ユニットの動作モードがプログラムモードに遷移したとき
- 同期（ダウンロード）を実行したとき（DB 接続設定の転送有無とは関係がありません）
- メモリオールクリアを実行したとき
- Sysmac Studio からコントローラバックアップ機能のリストアを実行したとき
- 全停止フォールトレベルのコントローラ異常発生時
- DB 接続サービスのシャットダウンを指示したとき



参考

- DB に SQL 文を送信し、DB からの応答を待っている状態で、DB 接続サービスを停止した場合は、DB からの応答が返ってきた後、または通信異常を検知した後に DB 接続サービスが停止します。
- DB 接続サービス停止時に接続中の DB コネクションが存在する場合は、DB コネクションを切断します。

4-1-4 DB 接続サービスの動作モードの変更

DB 接続サービス動作中に、DB 接続サービス動作モードを稼働モードとテストモード間で変更することはできません。
DB 接続サービスの動作モードを変更する場合は、一度、DB 接続サービスを停止してから、再度、DB 接続サービスを開始してください。

4-2　各 DB コネクションの接続／切断

DB サービス開始した後、各 DB コネクションの接続／切断を行います。
各 DB コネクションの接続／切断は、以下のように、命令によって実行します。

● 各 DB コネクションを接続するとき

DB_Connect（DB コネクション接続）命令を、DB コネクション名を指定して実行します。



使用上の注意

DB へのマッピングは、コネクションを切断すると自動的に解除されます。
そのため、DB_CreateMapping（DB マップ生成）命令を実行する前に、DB_Connect（DB コネクション接続）命令を実行するようにプログラミングしてください。

● 各 DB コネクションを切断するとき

DB_Close（DB コネクション切断）命令で、DB_Connect（DB コネクション接続）命令で取得した DB コネクションを指定して実行します。

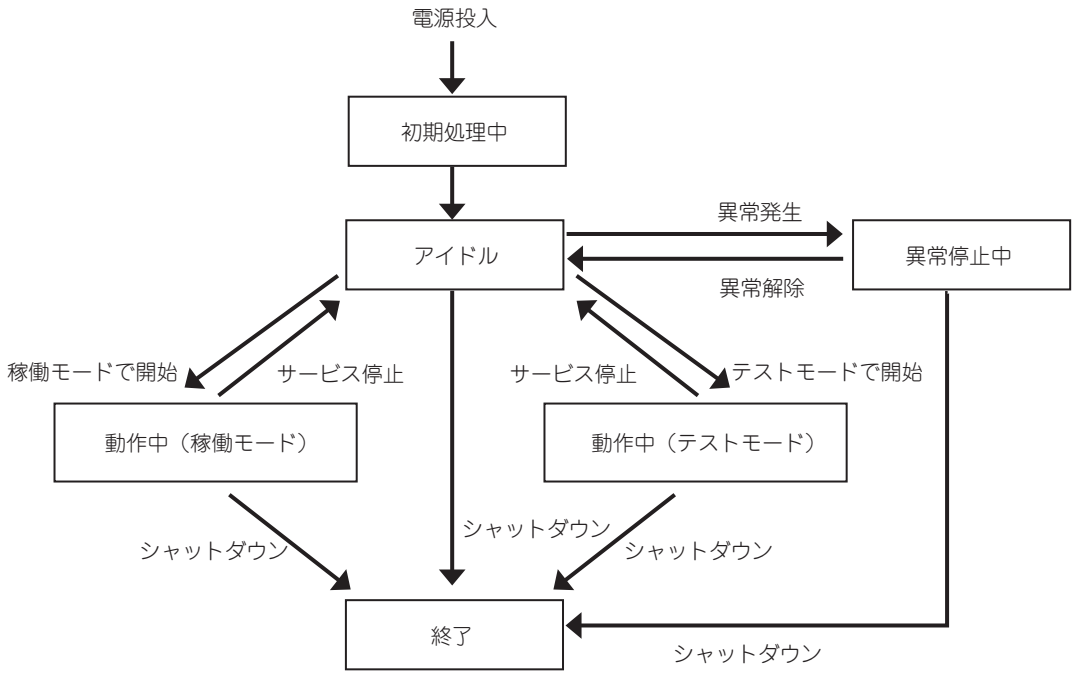
各命令の詳細は、「第 7 章 DB 接続命令 (P.7-1)」を参照してください。

4-3 DB 接続サービスおよび各コネクションの状態の確認

ここでは、以下の状態を確認する方法を説明します。
- DB 接続サービス
- 各 DB コネクション

4-3-1 DB 接続サービスの稼働状態

ここでは、DB 接続サービスの稼働状態について、説明しています。



DB 接続サービスは、“初期処理中”、“アイドル”、“動作中（稼働モード）”、“動作中（テストモード）”、“異常停止中”、“終了”の 6 つの稼働状態をもちます。
CPU ユニットの電源投入後、DB 接続サービスは“初期処理中”状態となります。初期処理が完了すると、“アイドル”状態になります。“アイドル”状態において、DB 接続サービス設定が壊れている場合、“異常停止中”となります。異常が解除されると“アイドル”状態に復帰します。
DB 接続サービスを開始すると、DB 接続サービスの動作モードにしたがって、“動作中（稼働モード）”または“動作中（テストモード）”になります。
“動作中（稼働モード）”または“動作中（テストモード）”状態において、DB 接続サービスを停止すると“アイドル”状態となります。
DB 接続サービスのシャットダウン機能を実行すると、“終了”状態になります。
各状態についての詳細を以下に示します。

状態	説明	備考
初期処理中	CPU ユニットの電源投入時に、DB 接続サービスを起動し、アイドル状態になるまでの状態	DB 接続サービスを開始できません。
アイドル	異常停止を除き、DB 接続サービスが停止している状態	DB 接続設定の変更ができます。 DB 接続命令は実行不可です。
動作中（稼働モード）	稼働モードで DB 接続サービスが動作している状態	DB 接続設定は変更不可です。 DB 接続命令は実行可です。

状態	説明	備考
動作中（テストモード）	テストモードで DB 接続サービスが動作している状態	DB 接続設定は変更不可です。 DB 接続命令は実行可です（ただし、DB に SQL は送信しません）。
異常停止中	DB 接続サービスの異常により動作できない状態	以下の場合、異常停止中となります。 • DB 接続サービス設定が壊れている場合
終了	DB 接続サービスが終了している状態	命令または Sysmac Studio から、DB 接続サービスのシャットダウンを指示したときにこの状態に移行します。 DB 接続サービスのシャットダウン処理が終了した後は、安全に CPU ユニットの電源を OFF することができます。コントローラリセットまたはコントローラの電源再投入を行うまで、DB 接続サービスを開始することはできません。

使用上の注意

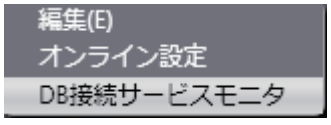
- DB 接続設定の［運転時のサービス開始］を［使用しない］から［使用する］に変更し、ダウンロード（同期）またはリストアした場合に、DB 接続サービスを開始するにはコントローラの電源を再投入またはコントローラリセットをする必要があります。
  コントローラの電源再投入またはコントローラリセットをしないと、DB 接続サービスを開始できません。
- 形 NX502-1□00 の場合、以下の条件でシャットダウン機能が実行され“終了”状態となります
  - メモリオールクリアを実行したとき
  - ［運転時のサービス開始］が［使用しない］に設定されているプロジェクトを同期したとき
  - ［運転時のサービス開始］が［使用しない］に設定されているプロジェクトを、Sysmac Studio からコントローラバックアップ機能でリストアしたとき

4-3-2 DB 接続サービスの状態の確認

DB 接続サービスの状態を、以下のいずれかの方法で確認できます。
- Sysmac Studio の DB 接続サービスモニタ
- DB 接続サービスの状態取得命令
- システム定義変数

Sysmac Studio の DB 接続サービスモニタによる確認

NJ/NX シリーズ CPU ユニットにオンライン接続している状態で、Sysmac Studio の［構成・設定］－［上位接続設定］－［DB 接続］の下の［DB 接続サービス設定］で、右クリックし［DB 接続サービスモニタ］を選択します。



以下の［DB 接続サービスモニタ］画面が表示されます。



DB 接続サービスの稼働状態が“初期処理中”“終了”以外のとき、以下の状態を確認することができます。

分類	項目	内容	値
稼働情報	稼働状況	DB 接続サービスの稼働状態です。	• 動作中（稼働モード） • 動作中（テストモード） • アイドル • 異常停止中 詳細は、「4-3-1 DB 接続サービスの稼働状態 (P.4-7)」を参照してください。
	稼働時間	DB 接続サービス開始時からの経過時間です。	時間表示 （単位：d：h：m：s）
稼働ログ	デバッグログ	デバッグログを記録中のとき、ON となります。*1	ON/OFF
クエリ実行	正常実行回数	すべてのコネクションで正常に SQL 文を実行した総回数です。 スプールに保留された SQL 文を再送した回数も含みます。 DB 接続サービスを開始したときにクリアされます。	正常実行回数
	異常実行回数	すべてのコネクションで SQL 文の実行に失敗した総回数です。 スプールに保留されず SQL 文の実行を破棄した回数です。スプールに保留された数は含めません。 DB 接続サービスを開始したときにクリアされます。	異常実行回数
スプール	スプールデータ数	すべてのコネクションでスプールに保留されている SQL 文の数です。	スプールデータ個数

*1. 以下の場合、ログが停止してもデバッグログフラグは ON したままです。
- “ログが上限に達した時の動作”を“ログを停止”に設定し、上限に達した場合
- SD メモリカードの容量が不足している場合
- SD メモリカードへの書き込みに失敗した場合

DB 接続サービスの状態取得命令による確認

DB_GetServiceStatus（DB 接続サービスの状態取得）命令によって、以下の DB 接続サービスの稼働情報を確認することができます。

情報名	説明
デバッグログフラグ	デバッグログを記録中のとき、TRUE となります。*1
稼働時間	DB 接続サービスを開始してからの経過時間です。 DB 接続サービスを停止した場合は、開始から停止までの時間を保持します。次回の開始でクリアされます。

情報名	説明
正常実行回数	すべてのコネクションで正常に SQL 文を実行した総回数です。 スプールに保留された SQL 文を再送した回数も含みます。 DB 接続サービスを開始したときにクリアされます。
異常実行回数	すべてのコネクションで SQL 文の実行に失敗した総回数です。 DB 接続サービスを開始したときにクリアされます。
スプールデータ数	すべてのコネクションでスプールに保留されている SQL 文の数です。

*1. 以下の場合、ログが停止してもデバッグログフラグは TRUE のままです。
- “ログが上限に達した時の動作”を“ログを停止”に設定し、上限に達した場合
- SD メモリカードの容量が不足している場合
- SD メモリカードへの書き込みに失敗した場合

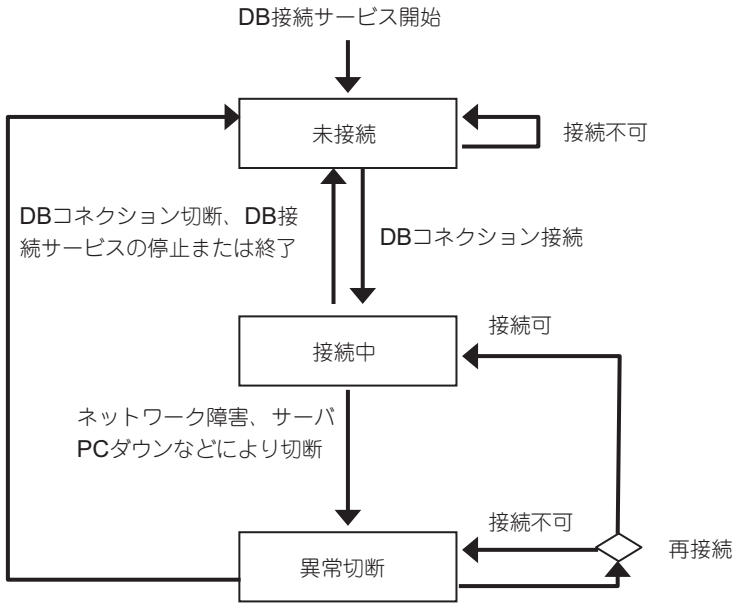
システム定義変数による確認

システム定義変数「_DBC_Status」によって、DB 接続サービスの稼働状態を確認することができます。
ユーザプログラムから DB 接続サービスの状態を参照する場合や、表示器から DB 接続サービスの終了を確認する場合に使用します。

システム定義変数 _DBC_Status		状態					
メンバ名	名称	初期処理中	動作中（稼働モード）	動作中（テストモード）	アイドル	異常停止中	終了
Run	動作中フラグ	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE
Test	テストモード	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE
Idle	アイドル	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE
Error	異常停止中フラグ	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE
Shutdown	終了	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE

4-3-3 DB コネクションの接続状態

ここでは、DB コネクションの接続状態について、説明しています。



DB コネクションは、“未接続”、“接続中”、“異常切断”の 3 つ接続の状態をもちます。
DB 接続サービス開始後、DB コネクションは“未接続”状態となります。“未接続”状態において、DB コネクションを接続すると、“接続中”状態となります。DB コネクションの接続に失敗すると、“未接続”状態のままです。
“接続中”状態において、ネットワーク障害、サーバ PC ダウンなどが発生すると、“異常切断”状態となります。
“異常切断”状態において、定期的に再接続を試み、接続可能であれば“接続中”状態に、接続不可であれば“異常切断”状態のままとなります。
“接続中”または“異常切断”状態において、DB コネクションを切断、DB 接続サービスを停止または終了すると“未接続”状態になります。
各状態についての詳細を以下に示します。

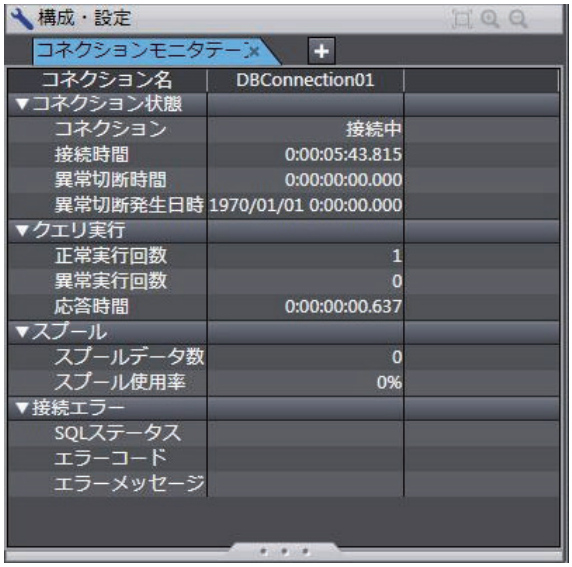
状態	説明	備考
未接続	DB に接続していない状態	
接続中	DB に接続している状態	命令から INSERT/SELECT など SQL 文を正常に実行できます
異常切断	接続中にネットワーク障害、サーバ PC ダウンなどにより切断された状態	命令実行中にこの状態に遷移した場合は、SQL 文はスプールに保留されます。 定期的に再接続を試みます。

4-3-4　各 DB コネクションの状態の確認

各 DB コネクションの状態を、以下のいずれかの方法で確認できます。
- Sysmac Studio のコネクションモニタテーブル
- DB コネクションの状態取得命令

Sysmac Studio のコネクションモニタテーブルによる確認

NJ/NX シリーズ CPU ユニットにオンライン接続している状態で、Sysmac Studio の［構成・設定］－［上位接続設定］－［DB 接続］の下の［DB コネクション設定］で、右クリックし［コネクションモニタテーブル］を選択します。
以下の［コネクションモニタテーブル］画面が表示されます。



DB 接続サービスの稼働状態が“アイドル”“終了”以外のとき、各 DB コネクションの以下の接続状態をモニタできます。

分類	項目	内容	値
コネクション状態	コネクション	DB コネクションの接続状態です。	• 未接続 • 接続中 • 異常切断 詳細は、「4-3-3 DB コネクションの接続状態 (P.4-10)」を参照してください。
	接続時間	接続中状態の時間です。 未接続状態から接続中状態になったときにクリアされます。	時間表示（単位：d：h：m：s.ms）
	異常切断時間	異常切断状態の時間です。 未接続状態から接続中状態になったときにクリアされます。	時間表示（単位：d：h：m：s.ms）
	異常切断発生日時	接続中に、ネットワーク障害、サーバ PC ダウンなどにより DB コネクションが異常切断された日時です。*1 DB 接続サービスを開始したときにクリアされます。	日時表示
クエリ実行	正常実行回数	正常に SQL 文を実行した回数です。 スプールに保留された SQL 文を再送した回数も含みます。 DB 接続サービスを開始したときにクリアされます。	正常実行回数
	異常実行回数	SQL 文の実行に失敗した回数です。 スプールに保留されず SQL 文の実行を破棄した回数です。スプールに保留された数は含めません。 DB 接続サービスを開始したときにクリアされます。	異常実行回数
	応答時間	SQL 文実行時に、CPU ユニットから SQL 文を送信してから、CPU ユニットに SQL 実行結果が返ってくるまでの時間です。 DB コネクションで最後に実行した SQL 文の送信*2 後、DB から正常応答が返ったときのみ DB 応答時間が格納されます。 DB 接続命令実行タイムアウト発生時は、命令の実行完了時点（つまり、出力変数 Error が FALSE→TRUE に変化した時点）では、DB 応答時間は格納されません。前回の DB 応答時間が保持されます。 DB 接続命令実行タイムアウト発生後、DB から正常応答が返った時点で、DB 応答時間が格納されます。 DB 接続サービスを開始したときにクリアされます。	時間表示（単位：d：h：m：s.ms）

分類	項目	内容	値
スプール	スプールデータ数	スプールに保留されている SQL 文の数です。	スプールデータ個数
	スプール使用率	DB コネクションあたりのスプールの容量に対する、スプールの使用率です。	スプール使用率（単位：%）
接続エラー	SQL ステータス	ネットワーク障害発生時または SQL 実行エラー発生時の、SQL 標準規格（ISO/IEC9075、JISX3005-2）で定義されたエラーコードです。*3 コネクションで最後に発生したエラーの値が保持されます。 DB 接続サービスを開始したときにクリアされます。	-
	エラーコード	ネットワーク障害発生時または SQL 実行エラー発生時の、DB ベンダ固有のエラーコードです。*3 ネットワーク障害発生時は、エラーコードが 0 と表示される場合があります。0 と表示されている場合は、SQL ステータスを確認してください。 コネクションで最後に発生したエラーコードが保持されます。DB 接続サービスを開始したときにクリアされます。	-
	エラーメッセージ	ネットワーク障害発生時または SQL 実行エラー発生時の、DB ベンダ固有のエラーメッセージです。*3 コネクションで最後に発生したエラーメッセージが保持されます。 DB 接続サービスを開始したときにクリアされます。	-

*1. コントローラの電源を ON したときのタイムゾーンに従います。タイムゾーンを変更した場合、コントローラの電源を再投入してください。
*2. レコード処理命令、ストアドプロシージャ呼出命令、またはスプールの再送（自動または DB_ControlSpool 命令による手動送信）を指します。命令については「3-5-3 DB 接続命令一覧 (P.3-22)」を参照してください。
*3. CPU ユニットのユニットバージョンによって、値が異なる場合があります。
CPU ユニット Ver.1.08 で、SQL Server の接続エラーの値が変更されています。

DB コネクションの状態取得命令による確認

DB_GetConnectionStatus （DB コネクションの状態取得）命令によって、DB コネクションの接続状態および接続情報を確認できます。

情報名	説明
DB コネクションの接続状態	DB コネクションの接続状態（未接続／接続中／異常切断）です。

情報名		説明
DB コネクションの接続情報	接続時間	接続中状態の時間です。 未接続状態から接続中状態になったときにクリアされます。
	異常切断時間	異常切断状態の時間です。 未接続状態から接続中状態になったときにクリアされます。
	正常実行回数	正常に SQL 文を実行した回数です。 スプールに保留された SQL 文を再送した回数も含みます。 DB 接続サービスを開始したときにクリアされます。
	異常実行回数	SQL 文の実行に失敗した回数です。 異常実行回数は、スプールに保留されず SQL 文の実行を破棄した回数です。 スプールに保留された数は含みません。 DB 接続サービスを開始したときにクリアされます。
	スプールデータ数	スプールに保留されている SQL 文の数です。 スプールをクリアしたときに 0 に戻ります。
	スプールの使用率	DB コネクションあたりのスプールの容量に対する、スプールの使用率（単位は%）です。 スプールをクリアしたときに 0 に戻ります。
	異常切断発生日時	接続中にネットワーク障害、サーバ PC ダウンなどにより異常切断された日時です。*1 DB 接続サービスを開始したときにクリアされます。
	SQL ステータス	ネットワーク障害発生時または SQL 実行エラー発生時の SQL 標準規格（ISO/IEC9075、JISX3005-2）で定義されたエラーコードです。*2 DB 接続サービスを開始したときにクリアされます。
	エラーコード	ネットワーク障害発生時または SQL 実行エラー発生時の DB ベンダ固有のエラーコードです。*2 ネットワーク障害発生時は、エラーコードが 0 となる場合があります。0 となっている場合は、SQL ステータスを確認してください。 DB 接続サービスを開始したときにクリアされます。
	エラーメッセージ	ネットワーク障害発生時または SQL 実行エラー発生時の DB ベンダ固有のエラーメッセージです。*2 DB 接続サービスを開始したときにクリアされます。

*1. コントローラの電源を ON したときのタイムゾーンに従います。タイムゾーンを変更した場合、コントローラの電源を再投入してください。

*2. CPU ユニットのユニットバージョンによって、値が異なる場合があります。
CPU ユニット Ver.1.08 で、SQL Server の接続エラーの値が変更されています。

5

各種の機能

この章では、DB 接続サービスの概要について説明しています。

5-1　各種機能の使用例 5-3

5-2　スプール機能 5-5
5-2-1　概要 5-5
5-2-2　スプール機能のしくみ 5-5
5-2-3　スプール機能の対象命令とスプール実行条件 5-5
5-2-4　スプール機能で使用するメモリ 5-6
5-2-5　スプール機能の設定 5-8
5-2-6　スプールに保留された SQL 文の再送方法 5-9
5-2-7　スプールに保留された SQL 文のクリア条件 5-10
5-2-8　DB 接続命令の動作との関係 5-11
5-2-9　スプールに保留可能な SQL 文数の目安の計算 5-13

5-3　ストアドプロシージャ呼出機能 5-15
5-3-1　概要 5-15
5-3-2　DB ごとのストアドプロシージャ呼出機能の仕様 5-16
5-3-3　ストアドプロシージャ呼出機能の実行方法 5-17
5-3-4　DB マップ変数の DB へのマッピング 5-18
5-3-5　ストアドプロシージャ呼出時の異常 5-20

5-4　バッチインサート機能 5-23
5-4-1　概要 5-23
5-4-2　バッチインサート機能の実行方法 5-23

5-5　DB 接続サービスシャットダウン機能 5-25
5-5-1　概要 5-25
5-5-2　シャットダウン機能のしくみ 5-25
5-5-3　シャットダウン機能の実行方法 5-26
5-5-4　DB 接続サービスのシャットダウンの確認方法 5-26

5-6　電源断発生時の SQL 文消失の回避策 5-28
5-6-1　概要 5-28
5-6-2　回避の方法 5-28

5-7　タイムアウト監視機能 5-32
5-7-1　タイムアウト監視機能の種類 5-32
5-7-2　ログインタイムアウト 5-33
5-7-3　クエリ実行タイムアウト 5-33
5-7-4　通信タイムアウト 5-33
5-7-5　命令実行タイムアウト 5-34
5-7-6　KeepAlive 監視時間 5-34

5-8　その他の機能 5-35

5-8-1　DB 接続サービスにおける、バックアップ／リストア機能 5-35
5-8-2　DB 接続サービスにおける、操作権限認証 5-36
5-8-3　暗号化通信 5-36

5-1 各種機能の使用例

ここでは、本章で説明する各種機能の使用例を説明します。
DB 接続サービスには、サーバ側のリレーショナルデータベースとのデータ交換時に発生する各種事象に対応した各機能があります。
以下に、発生し得る典型的な事象およびその対処方法の概要と、DB 接続サービスの機能の関係を示します。対処方法の詳細については、本章で説明する各項を参照してください。

No.	典型的な事象	事象発生時の影響	対処方法の概要	DB 接続サービスの機能	本章の参照先
1	複数テーブルのデータを使用した集計など、複雑な処理が必要になった場合	• コントローラ側ですべての処理を行うと、全体の処理が非効率となる可能性	リレーショナルデータベース側に作成したストアドプロシージャを呼び出すストアドプロシージャ呼出機能を使用して、1 つの処理にまとめるなどを行い、事象に対する対策を実施してください。	• ストアドプロシージャ呼出機能	「5-3 ストアドプロシージャ呼出機能 (P.5-15)」
2	複数のレコード挿入(INSERT)が必要になった場合	• ユーザプログラムが複雑になり非効率となる可能性や、データをアプリケーションが要求する時間内に格納できない可能性	バッチインサート機能を使用して、事象に対する対策を実施してください。	• バッチインサート機能	「5-4 バッチインサート機能 (P.5-23)」
3	コントローラの電源断が発生した場合	• 送信しようとしている SQL 文の消失の可能性 • その結果、リレーショナルデータベースに格納するデータの欠損の可能性	次回電源投入時に SQL 文の再送を行う場合、スプール機能とユーザプログラムを組み合わせた、ユーザプログラム上の対策を実施してください。	• スプール機能	「5-2 スプール機能 (P.5-5)」 「5-6 電源断発生時の SQL 文消失の回避策 (P.5-28)」
			コントローラの電源側に無停電電源装置を使用してください。無停電電源装置による給電が維持できない場合は、DB 接続サービスをシャットダウンしてください。	• DB 接続サービスシャットダウン機能	「5-5 DB 接続サービスシャットダウン機能 (P.5-25)」
		• 稼働ログの消失の可能性	コントローラの電源側に無停電電源装置を使用してください。無停電電源装置による給電が維持できない場合は、DB 接続サービスをシャットダウンしてください。	• DB 接続サービスシャットダウン機能	「5-5 DB 接続サービスシャットダウン機能 (P.5-25)」
4	一時的にサーバ側の負荷が増加した場合	• DB 接続サービスの遅延の可能性 • その結果、リレーショナルデータベースに格納するデータの欠損の可能性	タイムアウト監視機能を使用して、各種の対策を実施してください。	• タイムアウト監視機能 • スプール機能	「5-7 タイムアウト監視機能 (P.5-32)」 「5-2 スプール機能 (P.5-5)」

No.	典型的な事象	事象発生時の影響	対処方法の概要	DB 接続サービスの機能	本章の参照先
5	サーバ側の応答速度が比較的長時間継続して低下した場合	スプール容量不足により、リレーショナルデータベースに格納するデータの欠損の可能性	データの欠損を防ぐため、スプールされている SQL 文が再送され容量不足が解消するまで、以下のいずれかを実施してください。 • 装置の運転の一時中断または低速化 • ユーザプログラムによるユーザ定義変数へのデータ退避	• スプール機能	「5-2 スプール機能 (P.5-5)」
6	長時間のサーバ側の障害が発生した場合 例： • Ethernet ネットワーク断線、ノイズ • ネットワーク機器・サーバの電源喪失 • サーバの DB の停止 • サーバのハードウェア故障	リレーショナルデータベースに格納するデータの欠損の可能性	• 上記 No.2 および 3 の対処を行ってください。 • サーバ側に無停電電源装置を使用してください。無停電電源装置による給電が維持できない場合は、DB 接続サービスをシャットダウンしてください。	• DB 接続サービスシャットダウン機能	「5-5 DB 接続サービスシャットダウン機能 (P.5-25)」

5-2 スプール機能

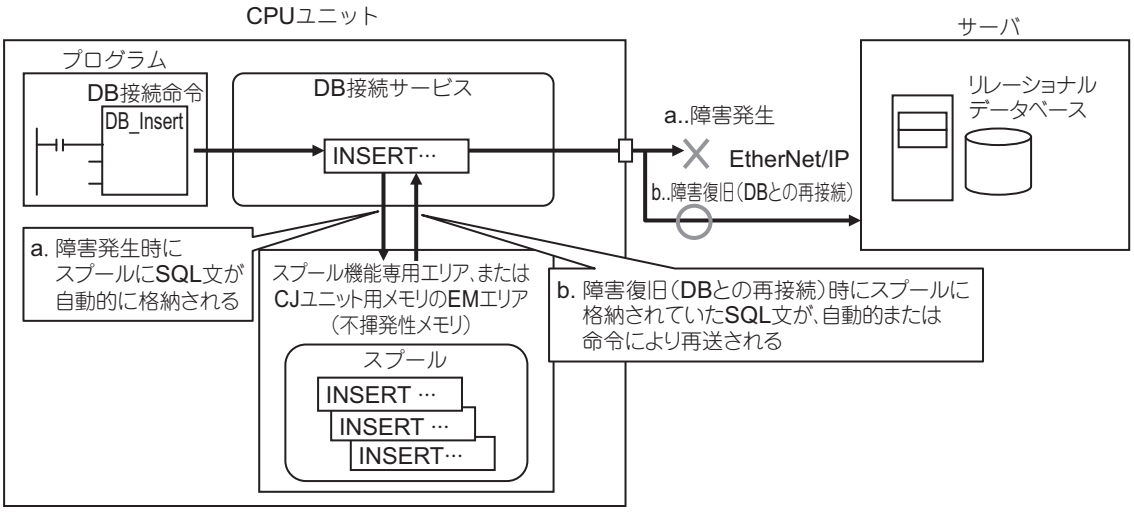
ここでは、DB 接続サービスにて、送信できなかった SQL 文をスプールする機能について説明をします。

5-2-1 概要

スプール機能とは、DB 接続サービスとサーバ側 DB との間の情報交換に障害が発生した場合に、送信できなかった SQL 文を CJ ユニット用メモリに保留し、障害が解消した時点で保留しておいた SQL 文を再送する機能です。
スプール機能の有効／無効は、DB コネクションごとに設定可能です。

5-2-2 スプール機能のしくみ

スプール機能のしくみを以下の図を使用して説明します。



a.　DB 接続サービスとサーバ側 DB との間の情報交換に障害が発生すると、送信できなかった SQL 文がスプール（NX シリーズの場合スプール機能専用エリア、NJ シリーズの場合 CJ ユニット用メモリの EM エリア）に自動的に保留されます。
b.　障害が復旧し、DB と再接続されると、スプールに保留されていた SQL 文が、自動的にまたは命令によって再送されます。

5-2-3 スプール機能の対象命令とスプール実行条件

対象命令

スプール機能の対象命令は、以下の 2 種類です。
- DB_Insert（DB レコード挿入）命令
- DB_Update（DB レコード更新）命令

使用上の注意

レコード挿入およびレコード更新以外の処理は、スプールに保留されません。再度、各処理の命令を実行してください。

スプール実行条件

以下のいずれかが発生したときに、スプールに SQL 文を保留します。
- 対象命令実行時に、ネットワーク障害が発生し、SQL 文を送信できなかったとき
- 対象命令実行時に、ネットワーク障害が発生し、DB からのレスポンスを受信できなかったとき
- 対象命令実行時に、サーバ側のパソコンのダウンなどにより、サーバ側の DB が停止しているとき
- 対象命令実行時に、既にスプールに保留された SQL 文が存在するとき
- 対象命令実行時に、DB 接続命令実行タイムアウトが発生したとき

使用上の注意

- スプール実行条件に該当する、命令が異常終了したときのエラーコードは以下のとおりです。以下のエラーコード以外で異常終了した場合、SQL 文はスプールに保留されません。
  3011Hex:DB コネクション異常切断発生中
  3012Hex:DB 接続命令実行タイムアウト
  3014Hex:スプールデータあり
  3016Hex:DB 処理中
- 命令エラー（SQL 実行エラー）が発生する場合、送信した SQL 文自体が SQL 実行エラーの原因である可能性があります。SQL 文を再送しても、再度 SQL 実行エラーが発生する可能性があるため、SQL 文はスプールに保留されません。
- DB からのレスポンスを受信できなかったときでも、送信した SQL 文がサーバ側 DB で処理されている可能性があります。

5-2-4　スプール機能で使用するメモリ

スプール機能で使用するメモリエリアは、以下のとおりです。NX シリーズと NJ シリーズで異なります。

形 NX701-□□20、形 NX502-1□00、形 NX102-□□20 の場合

スプール機能では、以下のメモリを使用します。

使用メモリ	説明	
スプール機能専用エリア	• 送信できなかった SQL 文は、スプール機能専用エリアに保留されます。	• スプール全体の容量： NX701-□□20：最大 2MB NX502-1□00：最大 2MB NX102-□□20：最大 192KB • DB コネクションあたりのスプールの容量：スプール機能を有効にした DB コネクション数で等分したサイズ。

スプール機能専用エリアは不揮発のメモリのため、CPU ユニットの電源断発生時に、スプールに保留された SQL 文の消失を回避することができます。

使用上の注意

- スプール機能専用エリアはバッテリによってデータが保持されています。
  バッテリが装着されていない状態、またはバッテリが消耗している状態になると、CPU ユニットがバッテリバックアップメモリチェックエラーを検出します。その場合は、スプールデータがクリアされます。
- 次の場合、スプールデータがクリアされます。
  a）“スプール設定”を“使用する”にして、プロジェクトをダウンロードする。この場合、ダウンロードするプロジェクトの“スプール設定”の設定には無関係にスプールデータがクリアされる。
  b）バックアップデータをリストアする。
- 形 NX102-□□20 のスプール機能専用エリアは保持変数メモリを使用します。そのため、Sysmac Studio の［メモリ使用状況］画面の［保持変数メモリ］に、初期状態でも 192KB 分の使用量を表示します。

形 NJ501-□□20 または形 NJ101-□□20 の場合

スプール機能では、以下のメモリを使用します。

使用メモリ	説明	
CJ ユニット用メモリ（EM）	• 送信できなかった SQL 文は、CJ ユニット用メモリの以下の EM エリアに保留されます。 EM バンク： NJ501-□□20：No.9Hex～18Hex の計 16 バンク NJ101-□□20：No.1Hex～3Hex の計 3 バンク	• スプール全体の容量： NJ501-□□20：最大 1MB NJ101-□□20：最大 192KB • DB コネクションあたりのスプールの容量：スプール機能を有効にした DB コネクション数で等分したサイズ。

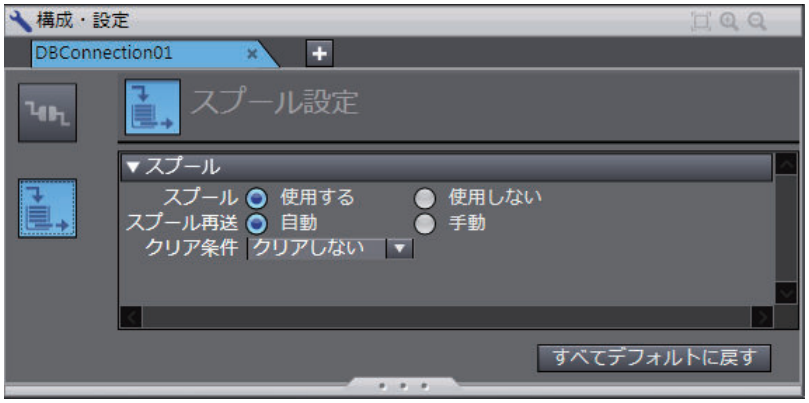
CJ ユニット用メモリの EM エリアは不揮発のメモリのため、CPU ユニットの電源断発生時に、スプールに保留された SQL 文の消失を回避することができます。

使用上の注意

- スプール機能が有効のとき、DB 接続サービスで EM バンクを使用します。
  以下の用途で DB 接続サービスが使用する EM バンクを使用するとスプールデータが破壊されますので、使用しないよう設計してください。
  a）ユーザ変数の割付先指定
  b）タグデータリンクのタグの I/O メモリアドレス指定
  c）通信コマンドによるアクセス
  d）表示器からのアクセス
  e）CJ 高機能ユニットの“高機能ユニット拡張割付メモリ”指定
- CJ ユニット用メモリの EM の値はバッテリによってデータが保持されています。
  バッテリが装着されていない状態、またはバッテリが消耗している状態になると、CPU ユニットがバッテリバックアップメモリチェックエラーを検出します。その場合は、スプールデータがクリアされます。
- “DB コネクション設定”の“スプール設定”のデフォルトは、“使用する”です。
  スプール機能を使用しない場合に、DB コネクションを追加するときは、必ず、“スプール設定”を“使用しない”に設定してから、DB 接続設定をダウンロードしてください。
  “使用する”の設定のまま、DB 接続設定をダウンロードすると、DB 接続サービスが使用する EM バンクに格納されている値が、スプール機能の初期化処理によって書き換わってしまいます。
- Sysmac Studio で変数／メモリのバックアップ／リストア実行時、対象のメモリ種別に“CJ ユニット用メモリの DM、EM、HR エリアの現在値”を選択すると、スプールデータもバックアップ／リストアされます。リストア実行後、スプールデータが不要な場合は、スプールに保留された SQL 文のクリアを実行してください。スプールに保留された SQL 文のクリアについては、「5-2-7 スプールに保留された SQL 文のクリア条件 (P.5-10)」を参照してください。

5-2-5　スプール機能の設定

Sysmac Studio の［構成・設定］－［上位接続設定］－［DB 接続］－［DB コネクション設定］の下の、DB コネクション名を右クリックし、［編集］を選択します。
［スプール設定］－［スプール］にて、設定をします。



スプール機能に関する、以下の項目を設定します。

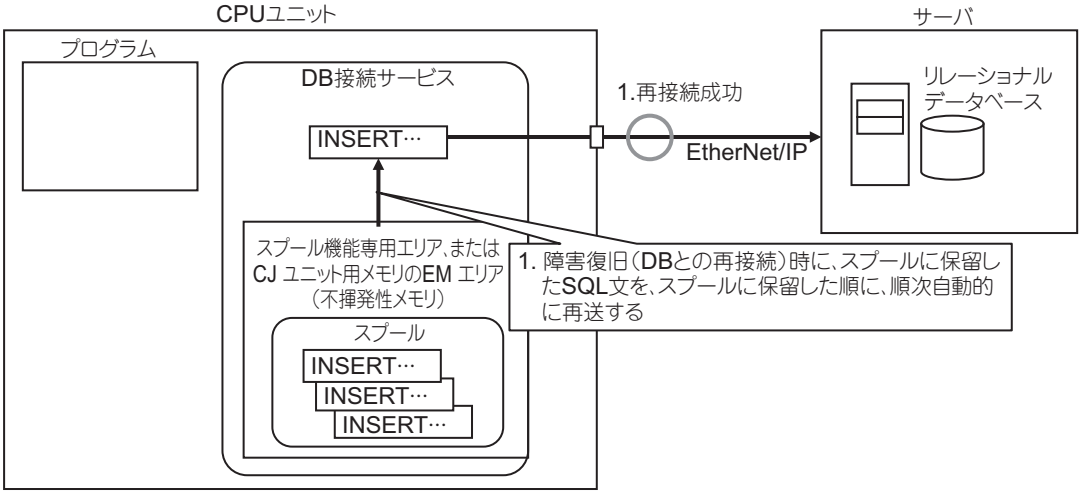
項目	内容	値
スプール	スプール機能を使用するかを設定します。	• 使用する（デフォルト） • 使用しない
スプール再送	“スプール”を“使用する”にした場合に設定する項目です。 スプールに保留された SQL 文を自動再送するか手動再送するかを設定します。	• 自動（デフォルト） • 手動
クリア条件	“スプール再送”を“自動再送”とした場合に設定する項目です。 スプールに保留された SQL 文をクリアする条件を設定します。	• クリアしない（デフォルト） • 電源投入時 • DB 接続サービス開始時 • コネクション確立時

5-2-6　スプールに保留された SQL 文の再送方法

スプールに保留された SQL 文の再送方法には、自動再送と手動再送の 2 種類があります。
スプール機能の設定の“スプール再送”にて、選択できます。

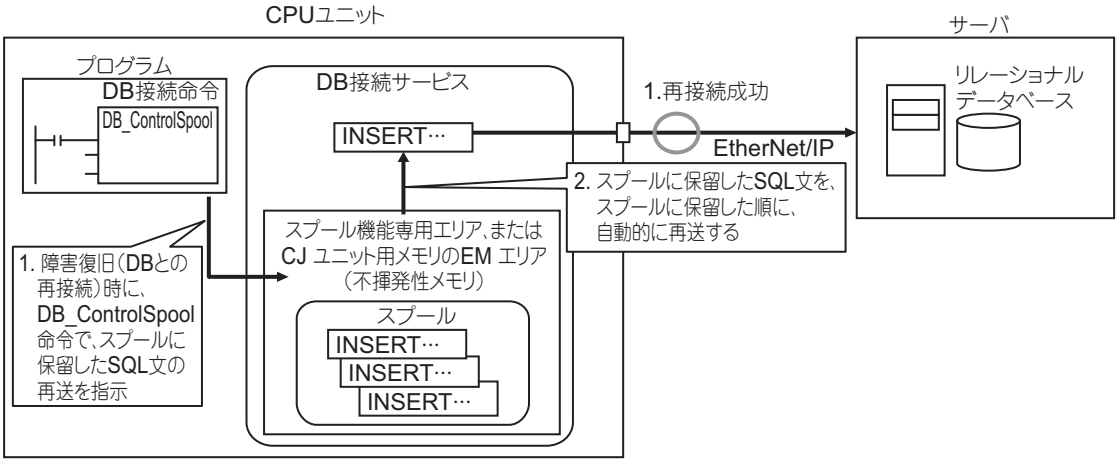
自動再送

DB との再接続が成功した時点で、スプールに保留された SQL 文を自動的に再送します。



手動再送

DB_ControlSpool （スプールの再送・クリア）命令実行時、スプールに保留された SQL 文を再送します。
1 回の DB_ControlSpool 命令実行で、スプールに保留されたすべての SQL 文をスプールに保留された順に再送します。



再送時に DB との情報交換に障害が発生した場合

スプールに保留された SQL 文の再送時に、再度、障害が発生した場合は、再送できなかった SQL 文はスプールに保留されたままになります。スプールに保留された SQL 文は、自動再送または手動再送により再送されます。再送順は変わりません。

5-2-7　スプールに保留された SQL 文のクリア条件

以下の場合に、スプールに保留された SQL 文がクリアされます
- 設定したクリア条件に一致
- DB_ControlSpool（スプールの再送・クリア）命令の実行
- Sysmac Studio からのオンライン操作
- 自動的にクリアされる条件に一致

バージョン関連情報

DB 接続サービスのバージョンが Ver.1.04 以降の場合、スプールに保留された SQL 文がクリアされると、イベントログに「スプールクリア（一般情報）」が登録されます。

設定したクリア条件に一致でクリアする

再送方法が自動再送の場合、Sysmac Studio の［DB コネクション設定］－［スプール設定］の［クリア条件］にて、スプールに保留された SQL 文のクリア条件を、DB コネクションごとに設定できます。以下のいずれかを選択します。

クリア条件	説明
クリアしない（デフォルト）	スプールに保留された SQL 文をクリアしません。
電源投入時	CPU ユニットの電源投入時にスプールに保留された SQL 文をクリアします。
DB 接続サービス開始時	DB 接続サービスを開始したときにスプールに保留された SQL 文をクリアします。
コネクション確立時	DB コネクションを接続（未接続→接続）したときにスプールに保留された SQL 文をクリアします。 このとき、スプールに保留された SQL 文は再送せずにクリアされます。

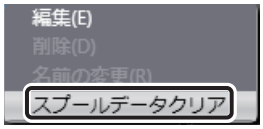
DB_ControlSpool 命令でクリアする

DB_ControlSpool（スプールの再送・クリア）命令の実行により、スプールに保留された SQL 文をクリアします。

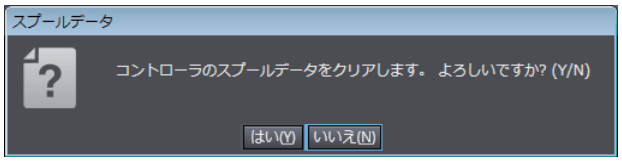
Sysmac Studio からのオンライン操作でクリアする

Sysmac Studio からの以下の操作でスプールに保留された SQL 文をクリアします。

1　NJ/NX シリーズ CPU ユニットにオンライン接続している状態で、Sysmac Studio のマルチビューエクスプローラ上のコネクションを右クリックし［スプールデータクリア］を選択します。



以下のメッセージが表示されます。



2 **［はい（Y)］をクリックします。**

自動的にクリアされる条件（注）

以下の場合には、スプール再送設定にかかわらず、スプールに保留された SQL 文は自動的にクリアされます。

- メモリオールクリアを実行したとき
- “バッテリバックアップメモリチェックエラー”発生時
- SD メモリカードバックアップ機能、および、Sysmac Studio コントローラバックアップ機能のリストアを実行したとき
- Sysmac Studio の変数・メモリのリストアで、メモリのリストアを実行したとき
- Sysmac Studio で、同期（ダウンロード）を実行したとき

（注）ただし、DB 接続サービスの稼働状態が終了状態（Shutdown が TRUE）のときは、自動的にクリアされる条件が成立しても、スプールに保留されたデータはクリアされません。クリアするには、同期後に Ethernet ケーブルをはずすなど DB との接続を物理的に切断し、CPU ユニットの電源を再投入します。その後、Sysmac Studio からのオンライン操作で［スプールデータクリア］を実行してください。

バージョン関連情報

DB 接続サービスのバージョンが Ver.1.03 以前の場合、DB コネクション設定を変更せずに同期（ダウンロード）を実行すると、スプールに保留された SQL 文はクリアされません。

使用上の注意

- ［DB 接続サービス設定］の［運転時のサービス開始］設定を複数回切り替えた場合、イベントログに“スプール破壊”が登録されることがあります。これは DB 接続の起動時に、前回の DB 接続稼働時から DB コネクション設定の変更があった場合に、スプール設定およびスプールを破壊することで発生します。
- リストア実行時に、リストア対象のコントローラの［運転時のサービス開始］が［使用しない］となっている場合、スプールはクリアされません。スプールをクリアするには、DB 接続サービスの稼働開始時にスプールのクリアを実行してください。

5-2-8 DB 接続命令の動作との関係

スプールに SQL 文が存在しているときの DB 接続命令の動作と、DB 接続命令を実行して命令実行タイムアウトが発生した場合のスプール機能への影響について、説明します。

スプールに保留された SQL 文が存在するときの DB 接続命令の動作

スプールに保留された SQL 文が存在する DB コネクションに対して、DB 接続命令を実行したときの動作は、以下のとおりです。

命令	動作
DB_Insert（DB レコード挿入）	SQL 文（INSERT）は、スプールに保留されます*1。 命令は異常終了します（Error=TRUE, SendStatus=_DBC_SEND_SPOOLED） 命令実行エラーの ErrorID については、「第 7 章 DB 接続命令 (P.7-1)」を参照してください。
DB_Update（DB レコード更新）	SQL 文（UPDATE）は、スプールに保留されます*1。 命令は異常終了します（Error=TRUE, SendStatus=_DBC_SEND_SPOOLED） 命令実行エラーの ErrorID については、「第 7 章 DB 接続命令 (P.7-1)」を参照してください。

命令	動作
DB_Select（DB レコード取得）	SQL 文（SELECT）は DB に送信されません。 命令実行エラーとなります（Error=TRUE）。 命令実行エラーの ErrorID については、「第 7 章 DB 接続命令 (P.7-1)」を参照してください。
DB_Delete（DB レコード削除）	SQL 文（DELETE）は DB に送信されません。 命令実行エラーとなります（Error=TRUE）。 命令実行エラーの ErrorID については、「第 7 章 DB 接続命令 (P.7-1)」を参照してください。

*1. ただし、スプールに SQL 文が保留されるときに、残りのスプール容量が足りない場合は、SQL 文はスプールに保留されず、破棄されます。

命令	動作
DB_Insert（DB レコード挿入）	SQL 文（INSERT）はサーバ側 DB に送信されません。 命令実行エラーとなります（Error=TRUE, SendStatus=_DBC_SEND_SENDING）。 命令実行エラーの ErrorID については、「第 7 章 DB 接続命令 (P.7-1)」を参照してください。
DB_Update（DB レコード更新）	SQL 文（UPDATE）はサーバ側 DB に送信されません。 命令実行エラーとなります（Error=TRUE, SendStatus=_DBC_SEND_SENDING）。 命令実行エラーの ErrorID については、「第 7 章 DB 接続命令 (P.7-1)」を参照してください。

DB 接続命令実行タイムアウト発生時の命令、および DB 接続サービスの動作

スプール機能が有効に設定されている場合に DB 接続命令実行タイムアウトが発生したとき、送信された SQL 文はスプールに保留されます。

DB 接続命令を実行してから、クエリ実行タイムアウト+10[*1] 秒が経過するまで、DB 接続サービスはサーバ側 DB からのレスポンスを待ちます。
サーバ側 DB からレスポンスが返ってきた時点で、スプールに保留された SQL 文が削除されます。

DB 接続命令を実行してから、クエリ実行タイムアウト+10[*1] 秒が経過しても、サーバ側 DB からレスポンスが返ってこない場合、DB コネクションの接続状態は“異常切断”に遷移します。

DB 接続サービスがサーバ側 DB からのレスポンスを待っている間に、レコード処理命令、ストアドプロシージャ命令を実行した場合、命令は異常終了（DB 処理中）します。

使用上の注意

DB 接続命令を実行してから、クエリ実行タイムアウト時間が経過した場合、該当の SQL 操作の取り消し要求が DB に送信されます。SQL 操作の取り消し処理の詳細は、以下の通りです。

(1)SQL 操作の取り消し処理が 10[*1] 秒以内に終了した場合
- 命令は異常終了（SQL 実行エラー）します。

(2)SQL 操作の取り消し処理が 10[*1] 秒以内に終了しない場合
- 通信タイムアウトが発生します。通信タイムアウトが発生した場合、命令は異常終了（DB コネクション異常切断発生中）し、DB コネクションの接続状態は“異常切断”に遷移します。
- DB_Insert（DB レコード挿入）命令、DB_Update（DB レコード更新）命令の場合、スプールに SQL 文が保留されます。
- スプール再送と DB コネクション異常切断が繰り返し発生する場合があります。この場合、クエリ実行タイムアウト時間を長くするか、SQL 操作を見直し、通信タイムアウトが発生しないように調整してください。タイムアウト監視機能については、「5-7 タイムアウト監視機能 (P.5-32)」を参照してください。

*1. DB の種別または DB の状態により、時間は変動します。

- **DB_Insert（DB レコード挿入）命令、DB_Update（DB レコード更新）命令の場合**

  スプール機能が有効であれば、送信しようとした SQL 文はスプールに保留されます。
  スプール再送設定にかかわらず、前回実行した DB 接続命令のレスポンスが返ってきたあとに、スプールに保留された SQL 文が送信されます。

- **DB_Insert（DB レコード挿入）命令、DB_Update（DB レコード更新）命令以外のレコード処理命令の場合**

  前回実行した DB 接続命令のレスポンスが返ってきたあとに、DB_Insert（DB レコード挿入）命令、DB_Update（DB レコード更新）命令以外のレコード処理命令を実行する場合は、実行した命令が正常終了するまでリトライするユーザプログラムを作成してください。

5-2-9　スプールに保留可能な SQL 文数の目安の計算

スプールに保留可能な SQL 文の数は、ユーザプログラムによって異なります。
ここでは、スプールに保留可能な SQL 文数の目安の計算方法について、説明します。

各 SQL 文のバイト数計算方法

各 SQL 文のバイト数を計算する方法は DB 接続サービスのバージョンによって異なります。DB 接続サービスのバージョンの確認方法は、「バージョン (P.21)」を参照してください。
SQL 文の内容は、デバッグログで確認できます。
デバッグログについては、「6-3 デバッグログ (P.6-14)」を参照してください。

- **DB 接続サービスのバージョンが、Ver.1.04 以降の場合**

命令	SQL 文	各 SQL 文のバイト数計算式*1
DB_Insert（DB レコード挿入）	<テーブル名><DB マップ変数名><DB マップ変数の値>	38+（<テーブル名>のバイト数） +（<DB マップ変数名>のバイト数） +（<DB マップ変数の値>のバイト数）
DB_Update（DB レコード更新）	<テーブル名><DB マップ変数名><DB マップ変数の値><抽出条件>	38+（<テーブル名>のバイト数） +（<DB マップ変数名>のバイト数） +（<DB マップ変数の値>のバイト数） +（<抽出条件>のバイト数）

*1.　SQL 文の文字列は UTF-8 で扱います。半角英数字 1 文字は 1 バイト、マルチバイト文字列は複数バイトを使用します。日本語文字は 3 バイトが目安です。

- **DB 接続サービスのバージョンが、Ver.1.03 以前の場合**

命令	SQL 文	各 SQL 文のバイト数計算式*1
DB_Insert（DB レコード挿入）	insert into <テーブル名> (<カラム名 1>, <カラム名 2>, <カラム名 3>..., <カラム名 n>) values(<値 1>, <値 2>, <値 3>..., <値 n>)	50+(<テーブル名>のバイト数) +(<カラム名 1>のバイト数) +(2+<カラム名 2>のバイト数) +(2+<カラム名 3>のバイト数) ... +(2+<カラム名 n>のバイト数) +(<値 1>のバイト数) +(2+<値 2>のバイト数) +(2+<値 3>のバイト数) ... +(2+<値 n>のバイト数)

命令	SQL 文	各 SQL 文のバイト数計算式*1
DB_Update（DB レコード更新）	update <テーブル名> set <カラム名 1>=<値 1>,<カラム名 2>=<値 2>...,<カラム名 n>=<値 n> where <抽出条件>	45+(<テーブル名>のバイト数) +(3+<カラム名 1>のバイト数+<値 1>のバイト数) +(5+<カラム名 2>のバイト数+<値 2>のバイト数) +(5+<カラム名 3>のバイト数+<値 3>のバイト数) ... +(5+<カラム名 n>のバイト数+<値 n>のバイト数) +(抽出条件のバイト数)

*1. SQL 文の文字列は UTF-8 で扱います。半角英数字 1 文字は 1 バイト、マルチバイト文字列は複数バイトを使用します。日本語文字は 3 バイトが目安です。

スプールに保留可能な SQL 文数の計算方法

各 DB コネクションで、スプールに保留可能な SQL 文数の目安を計算する方法は以下のとおりです。

スプールに保留可能な SQL 文数 =
DB コネクション 1 つあたりのスプールの容量［バイト］ ÷ 各 SQL 文のバイト数

DB コネクション 1 つあたりのスプールの容量［バイト］ =
スプール全体の容量（NX701-□□20 の場合：2097152［バイト］、NJ501-□□20 の場合：1048576［バイト］、NX102-□□20 および NJ101-□□20 の場合：196608［バイト］） ÷ スプール機能を有効にした DB コネクション数

5-3 ストアドプロシージャ呼出機能

ここでは、ストアドプロシージャ呼出機能について説明します。
ストアドプロシージャ呼出機能は、DB 接続サービスのバージョンが Ver.2.00 以降で使用可能です。
DB 接続サービスのバージョンと DB 接続機能の関係についての詳細は、「A-4 バージョン関連情報 (P.A-26)」を参照してください。

5-3-1 概要

ストアドプロシージャ呼出機能は、ネットワーク接続された他の DB に定義されたストアドプロシージャまたはストアドファンクションを、コントローラから呼び出す機能です。
本機能によって、複雑な処理演算をサーバ側で行うことができます。これによってコントローラのユーザプログラムを簡素化するだけでなく、演算処理能力の高いサーバで演算処理させることによる処理時間の短縮や、データの一貫性をサーバで担保できるなど、既存の INSERT/UPDATE/DELETE では実現が難しい操作を実現することができます。
ストアドプロシージャ呼出機能は、DB 接続命令を使って利用できます。通常、DB_AttachProcedure、DB_ExecuteProcedure、DB_DetachProcedure の順に DB 接続命令を呼び出して利用します。
ストアドプロシージャ呼出機能のしくみについては、「1-2-2 DB 接続機能のしくみ (P.1-10)」を参照してください。

ストアドプロシージャ呼出機能の仕様概要は、以下のとおりです。
- ストアドプロシージャに関する用語も用いて表記しています。
  ストアドプロシージャに関する用語については、対応する DB をリリースしている各社のマニュアルを参照してください。
- 各 DB の種別における仕様については、「5-3-2 DB ごとのストアドプロシージャ呼出機能の仕様 (P.5-16)」を参照してください。

項目	仕様
サポートする DB の種別	SQL Server Oracle Database MySQL Community Edition PostgreSQL
ストアドプロシージャの引数	最大 256 変数（1 構造体のメンバまたは基本データ型の配列変数による） • 引数を使用する場合は、DB_AttachProcedure（DB ストアドプロシージャ ハンドル生成）命令の入力変数「ArgIn」、「ArgOut」、「ArgInOut」を使用します。詳細は、「DB_AttachProcedure（DB ストアドプロシージャ ハンドル生成）（P.7-99)」を参照してください。 • 引数を省略する場合は、システム定義変数「_DBC_Unused」を入力変数に代入してください。
ストアドプロシージャの戻り値	1 変数（基本型であること） • 戻り値を使用する場合は、DB_AttachProcedure（DB ストアドプロシージャ ハンドル生成）命令の入力変数「ReturnVal」を使用します。詳細は、「DB_AttachProcedure（DB ストアドプロシージャ ハンドル生成）（P.7-99)」を参照してください。 • 戻り値を省略する場合は、システム定義変数「_DBC_Unused」を入力変数に代入してください。
ストアドプロシージャの結果セット	最大 256 × 65535 変数相当（構造体定義のメンバ数が最大 256 で、それらの構造体配列変数による） • 結果セットを使用する場合は、DB_AttachProcedure（DB ストアドプロシージャ ハンドル生成）命令の入力変数「ResultSet」を使用します。詳細は、「DB_AttachProcedure（DB ストアドプロシージャ ハンドル生成）（P.7-99)」を参照してください。 • 結果セットを省略する場合は、システム定義変数「_DBC_Unused」を入力変数に代入してください。

項目	仕様
スプール機能	未対応

使用上の注意

ストアドプロシージャ呼出機能の実行の際には、実行するストアドプロシージャの名称、処理内容および引数の値の確認を行った上で実行してください。
稼働ログを使用すると、実行時のエラーについて確認できます。
稼働ログについては、「第 6 章 稼働ログの使用手順 (P.6-1)」を参照してください。

5-3-2 DB ごとのストアドプロシージャ呼出機能の仕様

DB ごとのストアドプロシージャ呼出機能の仕様について説明します。
DB ごとのストアドプロシージャの引数、戻り値、結果セットとして NJ/NX シリーズ側がサポートする型については、「NJ/NX シリーズ側のデータ型と DB 側のデータ型の対応 (P.3-4)」を参照してください。
ストアドプロシージャに関する用語も用いて説明します。
ストアドプロシージャに関する用語については、対応する DB をリリースしている各社のマニュアルを参照してください。

使用上の注意

- ストアドプロシージャの名前は、大文字／小文字が区別されます。
- ストアドプロシージャの名前の禁則文字は、各 DB の仕様に従います。
- ストアドプロシージャの名前の長さは、稼働ログに出力できるプロシージャ名のデータサイズ以下となるようにしてください。

SQL Server

項目	内容
サポートするストアドプロシージャの種類	Transact-SQL によって作られたストアドプロシージャ、スカラー関数
ストアドプロシージャの名前の指定方法	スキーマを指定する場合：[スキーマ名].[プロシージャ名] カレントスキーマを省略する場合：[プロシージャ名]
注意事項	• SQLServer のストアドプロシージャの引数名は接頭辞として@を付ける必要がありますが、コントローラでは変数名の先頭に@を付けることができません。 このため、ストアドプロシージャの引数名から@を除いた文字列と、DB_AttachProcedure 命令の入力変数の「ArgIn」、「ArgOut」、「ArgInOut」、「ResultSet」に設定する構造体のメンバ変数名を一致させてください。 • SQL Server のストアドプロシージャにて、引数として OUTPUT キーワード付きで宣言された変数を、DB_AttachProcedure 命令の入力変数「ArgOut」に指定してください。 • DBCon Ver2.01.01 以前では、ストアドプロシージャが返す結果セットのデータサイズを合計 500KB 未満としてください。 データサイズが 500KB 以上となった場合、DB 接続サービスが停止することがあります。ストアドプロシージャが返す結果セットのデータサイズを見直してください。その後、コントローラを再起動してください。 • DBCon Ver2.01.02 以降では、ストアドプロシージャが返す結果セットを格納する構造体配列変数の配列要素数が多い場合、DB_ExecuteProcedure 命令がエラーコード 041BHex（データ容量超過）で異常終了することがあります。 その際は、構造体配列変数の配列要素数を減らし、コントローラを再起動してください。

Oracle Database

項目	内容
サポートするストアドプロシージャの種類	PL/SQL によって作られたスタンドアロンプロシージャ、パッケージプロシージャ
ストアドプロシージャの名前の指定方法	スキーマを指定して、パッケージプロシージャを指定する場合：[スキーマ名].[パッケージ名].[プロシージャ名] スキーマを指定する場合：[スキーマ名].[プロシージャ名] カレントスキーマを省略して、パッケージプロシージャを指定する場合：[パッケージ名].[プロシージャ名] カレントスキーマを省略する場合（パッケージプロシージャ以外）：[プロシージャ名]
注意事項	[スキーマ名]、[パッケージ名]、[プロシージャ名]は、それぞれ 30 バイト以下としてください。31 バイト以上の場合、命令エラー（ストアドプロシージャ名不正）が発生します。 ストアドプロシージャの結果セットを返したい場合は、DB_AttachProcedure 命令の入力変数の「ResultSet」に結果セットを受け取りたい構造体配列をセットし、ストアドプロシージャ側では SYS_REFCURSOR 型の OUT 引数を 1 個だけ用意し、そのカーソルに結果セットを代入してください。

MySQL Community Edition

項目	内容
サポートするストアドプロシージャの種類	ストアドプロシージャ、ストアドファンクション
ストアドプロシージャの名前の指定方法	DB を指定する場合：[DB 名].[プロシージャ名] カレント DB を省略する場合：[プロシージャ名]

PostgreSQL

項目	内容
サポートするストアドプロシージャの種類	• PL/pgSQL によって作られたストアドファンクション
ストアドプロシージャの名前の指定方法	スキーマを指定する場合：[スキーマ名].[プロシージャ名] カレントスキーマを省略する場合：[プロシージャ名]
注意事項	• PostgreSQL の SQL 関数は、引数名が存在しないため、サポート対象外です。 • PL/pgSQL でも引数名の省略が可能ですが、引数名を省略しているストアドファンクションはサポート対象外です。 • 結果セットは、REFCURSOR 型の OUT 引数を 1 個だけ用意し、そのカーソルに結果セットを代入してください。 • DBCon Ver2.01.01 以前では、ストアドファンクションが返す結果セットのデータサイズを合計 500KB 未満としてください。 データサイズが 500KB 以上となった場合、DB 接続サービスが停止することがあります。ストアドファンクションが返す結果セットのデータサイズを見直してください。その後、コントローラを再起動してください。

5-3-3　ストアドプロシージャ呼出機能の実行方法

ストアドプロシージャ呼出機能は、プロシージャハンドルおよび DB 接続命令を使用して実行します。以下にプロシージャハンドル、使用する命令および使用手順について説明します。

プロシージャハンドルとは

プロシージャハンドルは、ストアドプロシージャを一意に特定するための識別子（ハンドル）です。
DB 接続命令では、プロシージャハンドルを用いて個々のストアドプロシージャを識別します。

使用する命令

以下の命令を使用します。命令の詳細については各命令の説明を参照してください。

命令	参照
DB_AttachProcedure	「DB_AttachProcedure（DB ストアドプロシージャ ハンドル生成）（P.7-99)」
DB_ExecuteProcedure	「DB_ExecuteProcedure（DB ストアドプロシージャ 実行）（P.7-103)」
DB_DetachProcedure	「DB_DetachProcedure（DB ストアドプロシージャ ハンドル解放）（P.7-115)」

使用手順

使用手順は以下のとおりです。この使用手順は、DB の種別によらず共通です。
実行処理や入出力変数の詳細については、各命令の説明を参照してください。

手順	実行処理	該当命令	参照	備考
1	DB との接続を確立する	DB_Connect	「DB_Connect（DB コネクション接続）(P.7-6)」	
2	プロシージャハンドルを取得する	DB_AttachProcedure	「DB_AttachProcedure（DB ストアドプロシージャ ハンドル生成）（P.7-99)」	ストアドプロシージャ呼出機能使用時のみ必要
3	ストアドプロシージャを呼び出す	DB_ExecuteProcedure	「DB_ExecuteProcedure（DB ストアドプロシージャ 実行）（P.7-103)」	
4	プロシージャハンドルを解放する	DB_DetachProcedure	「DB_DetachProcedure（DB ストアドプロシージャ ハンドル解放）（P.7-115)」	
5	DB との接続を切断する	DB_Close	「DB_Close（DB コネクション切断）(P.7-9)」	

5-3-4 DB マップ変数の DB へのマッピング

ストアドプロシージャ呼出を実行する前に、DB マップ変数を DB にマッピングすることが必要です。
ここでは、ストアドプロシージャ呼出のための DB マッピングの方法、DB マッピングの解除方法および制約事項を説明します。

概要

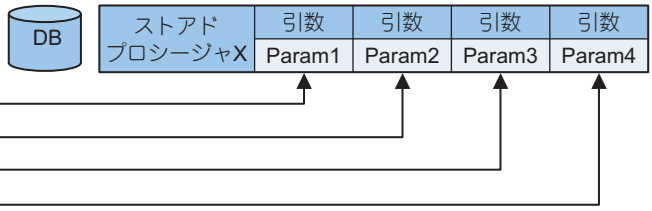
DB マップ変数の DB へのマッピングは、DB_AttachProcedure（DB ストアドプロシージャ ハンドル生成）命令にて行います。
DB_AttachProcedure（DB ストアドプロシージャ ハンドル生成）命令で、“ストアドプロシージャ名”や“引数”、“戻り値”、“結果セット”などを指定します。これにより、DB マップ変数を DB へマッピングできます。
詳細は、「DB_AttachProcedure（DB ストアドプロシージャ ハンドル生成）（P.7-99)」を参照してください。

引数へのマッピング

以下に引数へのマッピングについて例示します。

DBマップ変数が使用する構造体データ型の定義

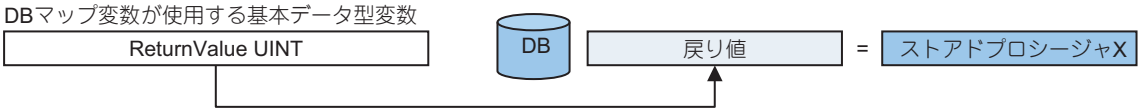
▼Structure1	
Param1	STRING[256]
Param2	UINT
Param3	UINT
Param4	UINT



戻り値へのマッピング

以下に戻り値へのマッピングについて例示します。

DBマップ変数が使用する基本データ型変数

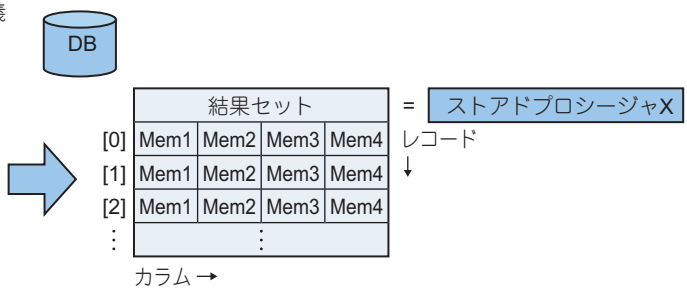


結果セットへのマッピング

以下に結果セットへのマッピングについて例示します。

DBマップ変数が使用する構造体データ型の定義

▼Structure1[0]	
Mem1	STRING[256]
Mem2	UINT
Mem3	UINT
Mem4	UINT
▼Structure1[1]	
Mem1	STRING[256]
Mem2	UINT
Mem3	UINT
Mem4	UINT
▼Structure1[2]	
Mem1	STRING[256]
Mem2	UINT
Mem3	UINT
Mem4	UINT
⋮	



DB マップ変数のマッピングの解除

ストアドプロシージャ呼出機能では、DB マップ変数のマッピングは DB_DetachProcedure 命令を実行したときに解除されます。

上記以外で解除される場合の詳細は、「3-4-2 DB マップ変数のマッピングの解除 (P.3-18)」を参照してください。

ストアドプロシージャの要素と DB マップ変数の対応関係

ストアドプロシージャの要素と DB マップ変数の対応関係は、以下のとおりです。

ストアドプロシージャの要素	DB_AttachProcedure 命令の仕様	備考
名前	ストアドプロシージャの名前を示す文字列であること。	「5-3-2 DB ごとのストアドプロシージャ呼出機能の仕様 (P.5-16)」を参照してください。
引数	構造体変数であること。 構造体メンバ名が引数名と一致していること。 ストアドプロシージャの IN 引数、INOUT 引数、OUT 引数の入力変数に指定された構造体のメンバ数の総和が 256 以下であること。	構造体メンバを順不同とすることは可能です。 構造体メンバの数の過不足はエラーとなります。 構造体メンバ名として指定できない文字については、「3-2-2 DB にアクセスするための構造体データ型の仕様 (P.3-3)」の使用上の注意を参照してください。
戻り値	基本データ型の変数であること。 ※派生型変数（配列、構造体）は使用できません。	
結果セット	構造体変数、または、構造体配列変数であること。 構造体のメンバ数が 256 以下であること。 配列要素数は、配列変数のシステム上限である 65535 要素まで使用可能です。	構造体メンバの数の過不足はエラーとなります。 構造体メンバ名として指定できない文字については、「3-2-2 DB にアクセスするための構造体データ型の仕様 (P.3-3)」の使用上の注意を参照してください。 ストアドプロシージャが返す結果セットのデータサイズには、制限があります。DB ごとの注意事項については、「5-3-2 DB ごとのストアドプロシージャ呼出機能の仕様 (P.5-16)」参照してください。

5-3-5 ストアドプロシージャ呼出時の異常

ここでは、ストアドプロシージャ呼出時の異常について説明します。
以下に、ストアドプロシージャ（プロシージャ名“proc1”）の例と、proc1 実行時の例外などを含む終了時の位置と終了要因および、位置と終了要因ごとの DB_ExecuteProcedure 命令の実行結果と対処について示します。

ストアドプロシージャの例（プロシージャ名“proc1”）

処理の概要	終了時の位置と終了要因
CREATE PROCEDURE proc1	
処理 1 BEGIN TRY 　IF（実行前チェック） 　　BEGIN 　　処理 2 　　RETURN 0 　　END 　ELSE 　　BEGIN 　　RETURN 1 　　END END TRY	1. TRY-Statement 外の処理 1 で例外終了 2. TRY-Statement 内の処理 2 で例外発生 3. TRY-Statement 内の処理 2 で正常終了 4. 実行前チェックエラー処理で終了
BEGIN CATCH 　例外発生時処理 　RETURN 2 END CATCH;	5. CATCH-Statement 内の例外発生時処理で例外終了 6. CATCH-Statement 内の例外発生時処理で終了
	7. クエリ実行タイムアウト時間内に応答がない

位置と終了要因ごとの DB 接続命令命令の実行結果と対処
※番号は前表における『終了時の位置と要因』列内の番号を表します。

番号	位置と終了要因	DB_ExecuteProcedure 命令の実行結果			戻り値	対処
		Done	Error	ErrorID		
1	TRY-Statement 外の処理 1 で例外終了	False	True	0x300B	変化しない（命令実行前と同じ値）	• DB_GetConnectionStatus 命令を実行し、出力変数「コネクション状態」を確認し、DB 側の状況に応じリトライするなどの対処をユーザプログラムに実装する • 失敗ログを確認する
2	TRY-Statement 内の処理 2 で例外発生：CATCH-Statement へ	－	－	－	－	－
3	TRY-Statement 内の処理 2 で正常終了	True	False	0x0000	0	戻り値ごとの対処をユーザプログラムに実装する
4	実行前チェック失敗時の処理で終了	True	False	0x0000	1	戻り値ごとの対処をユーザプログラムに実装する
5	CATCH-Statement 内の例外発生時処理で例外終了	False	True	0x300B	変化しない（命令実行前と同じ値）	• DB_GetConnectionStatus 命令を実行し、出力変数「コネクション状態」を確認し、DB 側の状況に応じリトライするなどの対処をユーザプログラムに実装する • 失敗ログを確認する
6	CATCH-Statement 内の例外発生時処理で終了	True	False	0x0000	2	戻り値ごとの対処をユーザプログラムに実装する
7	クエリ実行タイムアウト時間内に応答がない	False	True	0x300B	変化しない（命令実行前と同じ値）	• DB_GetConnectionStatus 命令を実行し、出力変数「コネクション状態」を確認し、DB 側の状況に応じリトライするなどの対処をユーザプログラムに実装する • 失敗ログを確認する

番号	位置と終了要因	DB_ExecuteProcedure 命令の実行結果			戻り値	対処
		Done	Error	ErrorID		
上記以外の、通信経路における異常など		False	True	0x300B 以外	変化しない（命令実行前と同じ値）	ErrorID の内容に応じた、異常終了時の処理を行う

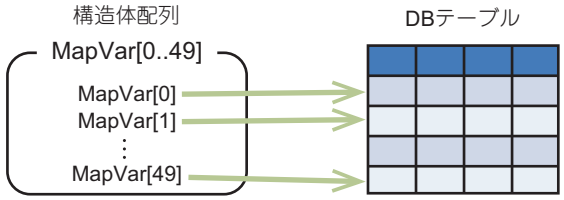
5-4 バッチインサート機能

ここでは、バッチインサート機能について説明します。
バッチインサート機能は、DB 接続サービスのバージョンが Ver.2.00 以降で使用可能です。
DB 接続サービスのバージョンと DB 接続機能の関係についての詳細は、「A-4 バージョン関連情報 (P.A-26)」を参照してください。

5-4-1 概要

バッチインサート機能は、複数レコードを挿入する場合に、一括で複数レコードを挿入する機能です。
バッチインサート機能を使用することで、挿入処理を複数回実行する場合よりも実行時間を短縮することができます。
バッチインサート機能は、DB 接続命令を使って利用できます。

バッチインサート機能の、データの準備や実行されることの概要は下図のとおりです。



構造体配列の1要素を1レコードとして挿入

- 図中の「MapVar」変数は、構造体配列で定義します。
- 配列要素数を n とすると、命令実行によって関連付けたテーブルへ、n レコード分一括で挿入されます。

バッチインサート機能の仕様は以下のとおりです。

項目	仕様
サポートする DB の種別	SQL Server Oracle Database MySQL Community Edition PostgreSQL
使用可能なデータやサイズ	バッチインサート機能では、一括挿入するデータとして、構造体配列を用います。 構造体配列として使用可能なデータやサイズは、以下のとおりです。 • 構造体配列の最大配列要素数：65,535 • 構造体配列内の 1 つの構造体のメンバ数（＝カラム数）：1,000 • 構造体配列内の 1 つの構造体のサイズ（=1 レコードのサイズ）：16KB 上記を満たさない場合、DB_CreateMapping 命令の実行は失敗します。
スプール機能	未対応
デバッグログ	情報が出力されますが、SQL 文は出力されません。
SQL 実行失敗ログ	DB 接続サービスの設定で SQL 実行失敗ログが有効の場合で、DB_BatchInsert 命令の入力変数の「SQLFailLog」が TRUE の場合のみ、ログを出力します。

5-4-2 バッチインサート機能の実行方法

バッチインサート機能は、DB 接続命令を使用して実行します。以下に使用命令と使用手順を説明します。

使用する命令

以下の命令を使用します。命令の詳細については各命令の説明を参照してください。

命令	参照
DB_BatchInsert（DB レコード一括挿入）	「DB_BatchInsert（DB レコード一括挿入）（P.7-87）」

使用手順

使用手順は以下のとおりです。この使用手順は、DB の種別によらず共通です。
実行処理の詳細については、各命令の説明を参照してください。

手順	実行処理	該当命令	参照	備考
1	DB との接続を確立する	DB_Connect	「DB_Connect（DB コネクション接続）（P.7-6）」	
2	DB マップ生成	DB_CreateMapping	「DB_CreateMapping（DB マップ生成）（P.7-11）」	
3	DB マップ変数に値をセットする			ユーザプログラムを作成し処理します
4	バッチインサートを実行する	DB_BatchInsert	「DB_BatchInsert（DB レコード一括挿入）（P.7-87）」	バッチインサート機能使用時のみ必要
5	DB との接続を切断する	DB_Close	「DB_Close（DB コネクション切断）（P.7-9）」	

5-5 DB 接続サービスシャットダウン機能

ここでは、稼働ログの内容が消失しないように、DB 接続サービスをシャットダウンする機能について説明します。
DB 接続サービスの稼働状態については、「4-3-1 DB 接続サービスの稼働状態 (P.4-7)」を参照してください。

5-5-1 概要

DB 接続サービスシャットダウン機能（以降“シャットダウン機能”）とは、稼働ログファイルを SD メモリカードに保存した上で、DB 接続サービスを終了する機能です。
CPU ユニットの電源を OFF する前にシャットダウン機能を実行してください。シャットダウン機能の実行により、稼働ログの内容は消失しません。

使用上の注意

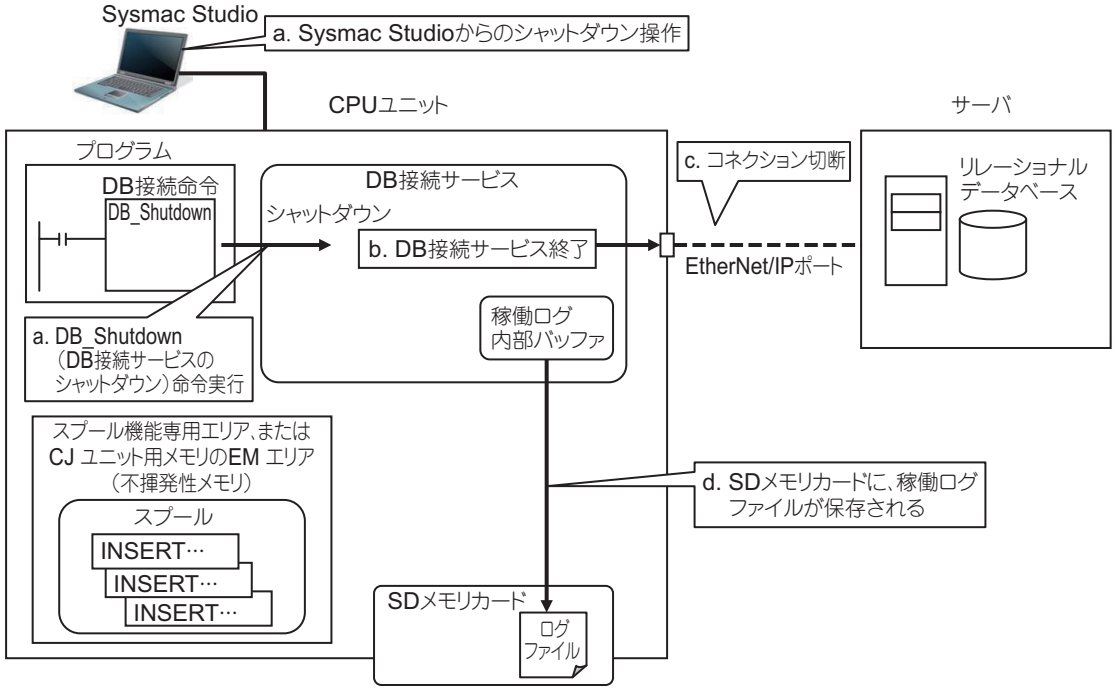
シャットダウン機能実行を実行せずに、DB 接続サービス実行中に CPU ユニットの電源を OFF にした場合は、稼働ログの内容は保証できません。稼働ログファイルが壊れる、または内容が消失する場合があります。

参考

予期せぬ停電などによるデータの消失を防ぐためには、無停電電源装置などによる停電対策をお奨めします。

5-5-2 シャットダウン機能のしくみ

シャットダウン機能のしくみを以下の図を使用して説明します。



a. Sysmac Studio からの操作で DB 接続サービスをシャットダウン、または、DB_Shutdown（DB 接続サービスのシャットダウン）命令を実行します。
b. DB 接続サービスが終了します。

c. DB コネクションが切断されます。
d. SD メモリカードに稼働ログファイル（実行ログファイル、デバッグログファイル、SQL 実行失敗ログファイル）が保存されます。

5-5-3 シャットダウン機能の実行方法

シャットダウン機能は、以下のいずれかによって実行できます。
- Sysmac Studio からの操作
- 命令実行による操作

バージョン関連情報

形 NX502-1□00 の場合は、以下の場合もシャットダウン機能が実行されます。
- メモリオールクリアを実行したとき
- ［運転時のサービス開始］が［使用しない］に設定されているプロジェクトを同期したとき
- ［運転時のサービス開始］が［使用しない］に設定されているプロジェクトを、Sysmac Studio からコントローラバックアップ機能でリストアしたとき

Sysmac Studio からの操作

NJ/NX シリーズ CPU ユニットにオンライン接続している状態で、Sysmac Studio の［構成・設定］－［上位接続設定］－［DB 接続］の下の［DB 接続サービス設定］で、右クリックして［オンライン設定］を選択します。［サービス］－［終了］の［シャットダウン］ボタンをクリックします。



参考

Sysmac Studio から“コントローラリセット”を実行したときは、コントローラリセット前に DB 接続サービスシャットダウン機能が自動で実行されます。

命令実行による操作

DB_Shutdown（DB 接続サービスのシャットダウン）命令を実行します。

5-5-4 DB 接続サービスのシャットダウンの確認方法

CPU ユニットの電源を OFF する前に、以下のいずれかの方法で、DB 接続サービスがシャットダウンしたことを確認してください。
- システム定義変数による確認

システム定義変数「_DBC_Status.Shutdown」（DB 接続サービスステータスの終了フラグ）が TRUE であることを確認します。

- 命令実行による確認
- DB_Shutdown（DB 接続サービスのシャットダウン）命令の出力変数「Done」が TRUE であることを確認します。

5-6 電源断発生時の SQL 文消失の回避策

ここでは、電源断が発生しても SQL 文が消失しないようにするユーザプログラム上の対策を示します。

5-6-1 概要

電源断発生時、スプール機能*1 やユーザプログラムを組み合わせることで、送信しようとした SQL 文の消失を回避することができます。
SQL 文消失の回避策について、対応方法を以下に示します。

	対応方法
スプール機能*1 をサポートする命令（例：DB_Insert など）	スプール設定を使用すると設定し、ユーザプログラムによる再送によって対応します。*2
スプール機能*1 をサポートしない命令（例：DB_BatchInsert など）	ユーザプログラムによる再送によって対応します。*2

*1. スプール機能については「5-2 スプール機能 (P.5-5)」を参照してください。
*2. 詳細は「5-6-2 回避の方法 (P.5-28)」を参照してください。

5-6-2 回避の方法

回避の方法は以下の方法によります。

DB 接続命令の進行状態の確認方法

DB 接続命令の進行状態は、以下の出力変数 SendStatus に列挙型のデータで出力されます。このデータを使用して、ユーザプログラムを作成します。

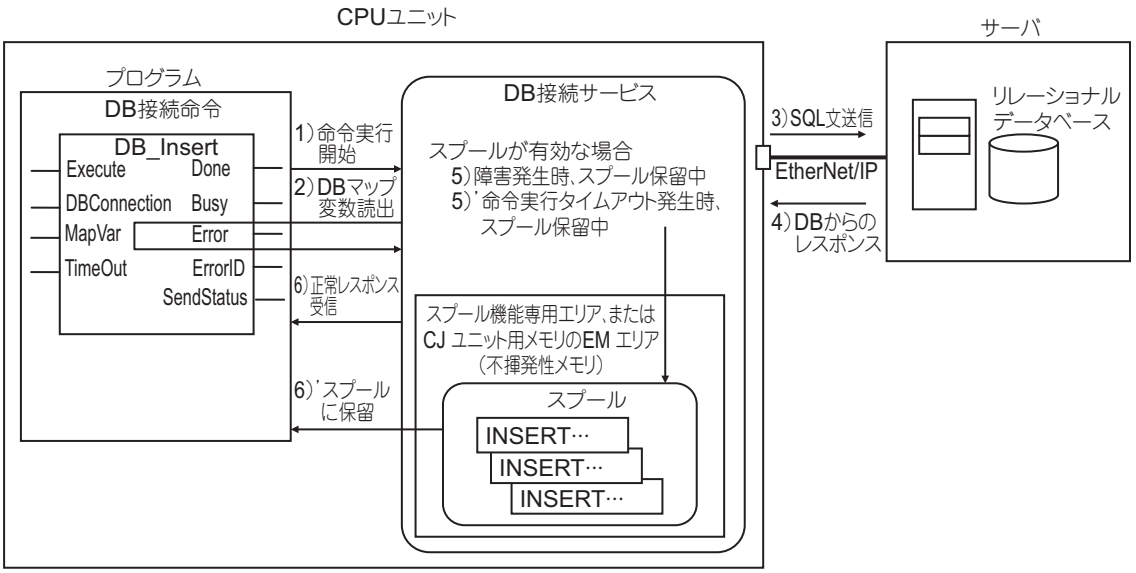
出力変数	名称	データ型	内容
SendStatus	送信状態	_eDBC_SEND_STATUS	_DBC_SEND_INIT(0)：初期状態 _DBC_SEND_UNSENT(1)：SQL 文未送信 _DBC_SEND_SENDING(2)：SQL 文送信中 _DBC_SEND_SPOOLED(3)：SQL 文がスプールに保留された _DBC_SEND_COMPLETE(4)：SQL 文送信完了

変数の設定

- 入力変数 MapVar の入力パラメータ（DB マップ変数）の保持属性を“保持する”に設定します。
- 出力変数 Busy の出力パラメータの保持属性を“保持する”に設定します。
- 出力変数 SendStatus の出力パラメータの保持属性を“保持する”に設定します。

電源断が発生するタイミングに応じて必要な対策

電源断が発生するタイミングに応じた対策が必要となります。
具体的な対策を、下図を使用して説明します。



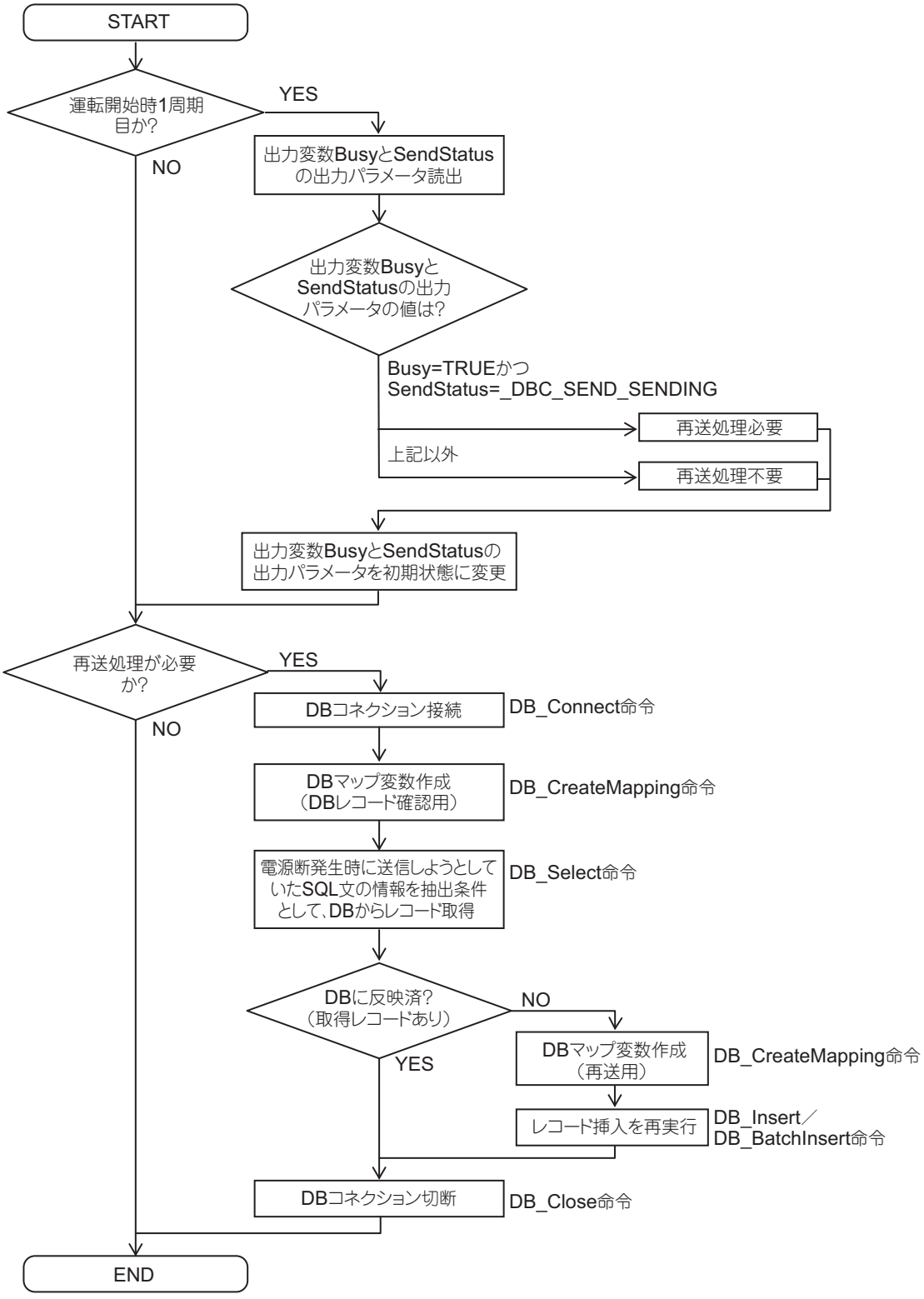
以下の DB 接続命令実行中の電源断のタイミングは、上図の番号に対応しています。

DB 接続命令実行中の電源断のタイミング		出力変数 SendStatus の値	対策
1) 命令実行開始	DB 接続命令の Execute が FALSE→TRUE になってから、DB 接続サービスが、DB マップ変数の現在値を読み出す前まで	_DBC_SEND_SENDING：SQL 文送信中	ユーザプログラムによる再送
2) DB マップ変数読出	DB 接続サービスが、DB マップ変数の現在値の読出を開始してから、SQL 文を DB に送信する前まで		
3) SQL 文送信	DB 接続サービスが、SQL 文を DB に送信する直前から、送信が終わるまで		
4) DB からのレスポンス	SQL 文を DB に送信してから、DB からのレスポンスを受信するまで		
5) 障害発生時、スプール保留中	障害発生時に SQL 文をスプールに保留している間（スプール機能が有効の場合）		
5)' 命令実行タイムアウト発生時、スプール保留中	命令実行タイムアウト発生時に SQL 文をスプールに保留している間（スプール機能が有効の場合）		
6) 正常レスポンス受信	DB から正常レスポンスを受信した以降	_DBC_SEND_COMPLETE：SQL 文送信完了	不要
6)' スプールに保留	SQL 文のスプールへの保留が完了した以降（スプール機能が有効の場合）	_DBC_SEND_SPOOLED：SQL 文をスプールに保留した	スプール機能による再送（自動再送または手動再送）

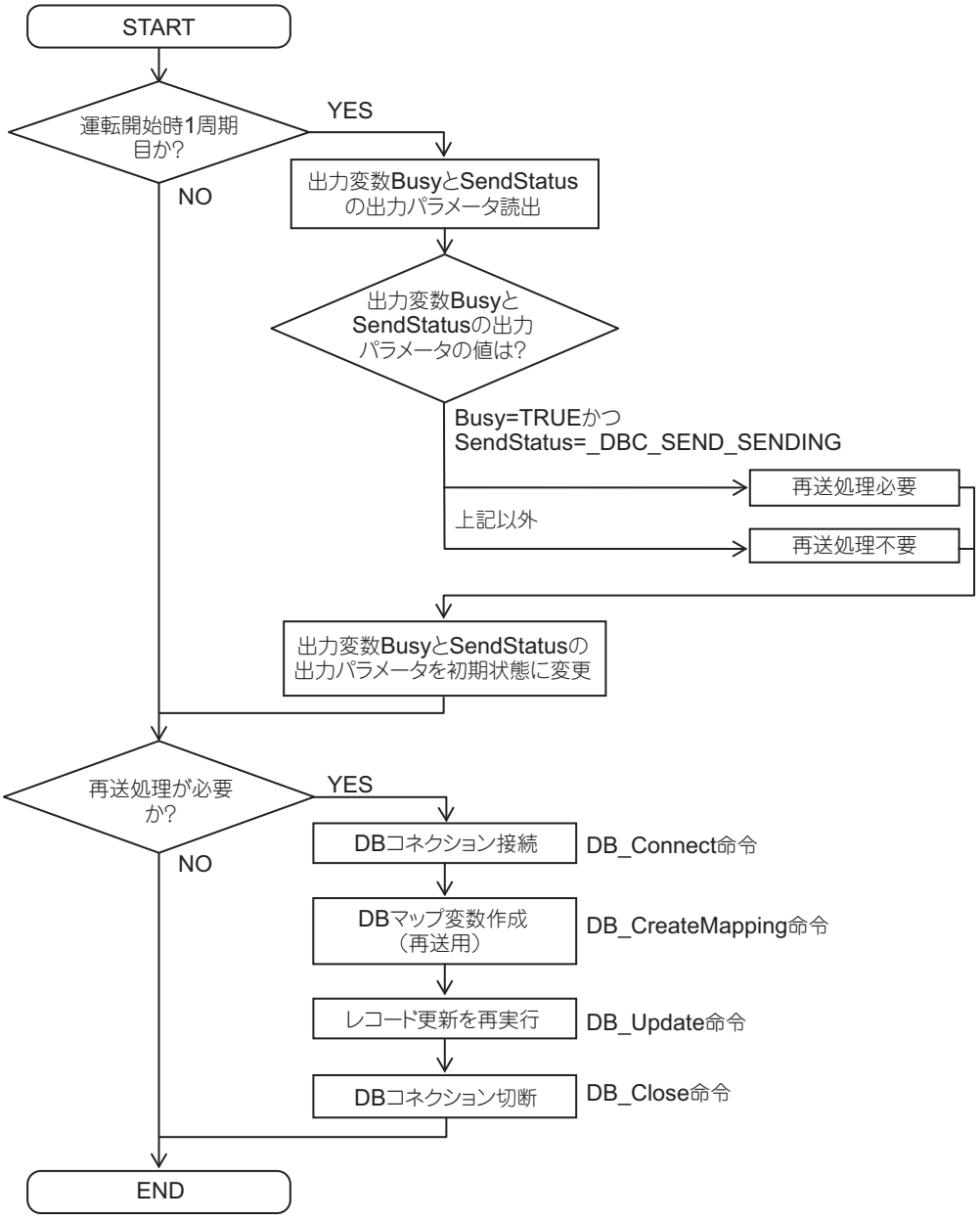
ユーザプログラムによる再送フロー

電源断発生時に、実行していた命令を再実行するユーザプログラムを作成します。
電源断発生時に、実行していた命令が DB_Insert／DB_BatchInsert 命令か、DB_Update 命令かで再送フローが異なります。

● **DB_Insert／DB_BatchInsert 命令の場合**



● **DB_Update 命令の場合**



使用上の注意

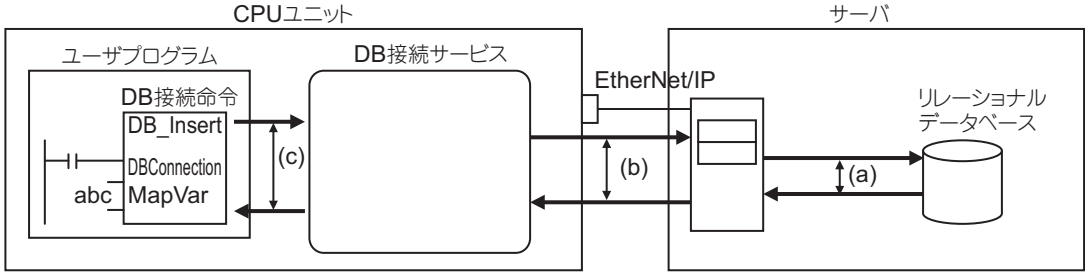
- 出力変数 SendStatus の出力パラメータは、入力変数 Execute の値にかかわらず、入力変数 Execute の値を評価したときに、出力変数 SendStatus の値で上書きされます。そのため、運転開始時 1 周期目に、DB 接続命令の入力変数 Execute の値が評価される前に出力変数 SendStatus の出力パラメータ読出を実行するようにユーザプログラムを作成してください。
- 運転開始時にすでに入力変数 Execute の値が TRUE の場合は、DB 接続命令を実行しません。命令を実行させるためには、Execute の値をいったん FALSE にする必要があります。

5-7　タイムアウト監視機能

ここでは、DB 接続サービスに関連した、タイムアウト監視について、説明します。

5-7-1　タイムアウト監視機能の種類

タイムアウト監視機能の種類を、下図を使用して説明します。



機能名	設定範囲	説明	参照
ログインタイムアウト時間	1～60 秒 デフォルト：10 秒	DB 接続サービスと DB 間の通信障害や、サーバ側の障害により、DB 接続サービスがログイン失敗を検出するまでの時間です。	「2-2-2 DB コネクション設定 (P.2-7)」
クエリ実行タイムアウト時間（図 a）	1～600 秒 デフォルト：30 秒	サーバ側 DB のクエリ実行に時間がかかった場合に、DB 接続サービスが異常を検出するまでの時間です。 サーバ側 DB のクエリ実行に想定以上の時間がかかった場合に、SQL 操作を取り消すことができます。	「2-2-2 DB コネクション設定 (P.2-7)」
通信タイムアウト時間（図 b）	クエリ実行タイムアウト時間＋ 10 秒*1	DB 接続サービスと DB 間に通信障害が発生した場合に、DB 接続サービスが異常を検出するまでの時間です。	－
命令実行タイムアウト時間（図 c）	監視なし、0.05～180 秒 デフォルト：監視なし	DB 接続サービスと DB 間の通信障害やサーバ側の障害または負荷により、レコード処理命令、ストアドプロシージャ命令の実行時間が長くなった場合に、DB 接続サービスが異常を検出するまでの時間です。 装置性能（タクト時間）を延ばしたくない場合に使用します。	「A-2-4 命令実行タイムアウト監視による装置性能（タクト時間）の確保 (P.A-24)」
KeepAlive 監視時間	1～65535 秒 デフォルト：300 秒	サーバとの接続が有効であるかを確認する機能です。 KeepAlive 監視時間を設定することで、サーバ側 DB がクエリ実行中で、DB 接続サービスがサーバからのレスポンスを待っている状態でも、通信障害を検出することができます。	• NJ/NX シリーズ CPU ユニットの EtherNet/IP ポートを利用する場合：『NJ/NX シリーズ CPU ユニット内蔵 EtherNet/IP ポート ユーザーズマニュアル (SBCD-377)』 • CPU ユニットに接続した NX シリーズ EtherNet/IP ユニットの EtherNet/IP ポートを利用する場合：『NX シリーズ EtherNet/IP ユニット ユーザーズマニュアル (SBCD-382)』

*1.　DB の種別または DB の状態により、通信タイムアウト時間は変動します。

5-7-2 ログインタイムアウト

ログインタイムアウトは以下のときに監視します。
- DB_Connect（DB コネクション接続）命令で、サーバ側 DB に接続するとき
- DB コネクションが“異常切断”状態のときの再接続時

ログインタイムアウトが発生したときの動作は以下のとおりです。

発生タイミング	発生後の DB コネクションの状態	命令実行結果
DB_Connect 命令実行時	未接続	ErrorID=3005Hex（DB コネクション接続失敗）
DB コネクション再接続時	異常切断	－

5-7-3 クエリ実行タイムアウト

クエリ実行タイムアウトは以下のときに監視します。
- レコード処理命令、ストアドプロシージャ命令でサーバ側 DB に SQL 文を送信するとき
- スプールに保留された SQL 文を再送するとき

クエリ実行タイムアウトが発生したときの動作は以下のとおりです。

発生タイミング	発生後の DB コネクションの状態	命令実行結果
DB_Insert 命令、DB_Update 命令、DB_ExecuteProcedure 命令、DB_BatchInsert 命令実行時	接続中	ErrorID=300BHex（SQL 実行エラー） SendStatus=_DBC_SEND_COMPLETE スプールに SQL 文は保留されません。*1
DB_Select 命令、DB_Delete 命令実行時	接続中	ErrorID=300BHex（SQL 実行エラー）
スプール再送時	接続中	SQL 文は、スプールに再度保留されません。*1

*1. 送信した SQL 文自体が SQL 実行エラーの原因である可能性があります。SQL 文を再送しても、再度 SQL 実行エラーが発生する可能性があるため、SQL 文はスプールに保留されません。

5-7-4 通信タイムアウト

通信タイムアウトは以下のときに監視します。
- レコード処理命令、ストアドプロシージャ命令でサーバ側 DB に SQL 文を送信するとき
- スプールに保留された SQL 文を再送するとき

通信タイムアウトが発生したときの動作は以下のとおりです。

発生タイミング	発生後の DB コネクションの状態	スプール機能	命令実行結果
DB_Insert 命令、DB_Update 命令、DB_ExecuteProcedure 命令、DB_BatchInsert 命令実行時	異常切断	有効	ErrorID=3011Hex（DB コネクション異常切断発生中） SendStatus=_DBC_SEND_SPOOLED スプールに SQL 文が保留されます。
		無効	ErrorID=3011Hex（DB コネクション異常切断発生中） SendStatus=_DBC_SEND_SENDING
DB_Select 命令、DB_Delete 命令実行時	異常切断	－	ErrorID=3011Hex（DB コネクション異常切断発生中） SendStatus=_DBC_SEND_SENDING
スプール再送時	異常切断	有効	SQL 文はスプールに再度保留されます。

5-7-5 命令実行タイムアウト

命令実行タイムアウトについての詳細は、「5-2-8 DB 接続命令の動作との関係 (P.5-11)」を参照してください。

5-7-6 KeepAlive 監視時間

DB コネクションの接続状態が“接続中”のときに、サーバとの接続が有効であるかを監視します。通信障害やサーバ側のパソコンのダウンが発生し、“KeepAlive 監視時間+12 秒”の間、サーバとの接続が確認できなかった場合に、レコード処理命令、ストアドプロシージャ命令を実行したとき、または、スプールを再送したときに、DB コネクションの接続状態は“異常切断”状態に遷移します。

KeepAlive について、DB 接続サービスでは以下の動作となります。
- KeepAlive の設定にかかわらず、常に“使用する”で動作
- Linger オプションの設定にかかわらず、常に“指定する”で動作

KeepAlive 監視により、DB コネクションを切断したあとの動作は、通信タイムアウト発生時と同じです。「5-7-4 通信タイムアウト (P.5-33)」を参照してください。

使用上の注意

- KeepAlive 監視時間は、EtherNet/IP ポートの共通の設定です。KeepAlive 監視時間を設定する場合は、EtherNet/IP ポートの以下の機能の動作に影響がないことを確認してから設定値を変更してください。
- ソケットサービス、FTP サーバ機能、Sysmac Studio との通信、FINS/TCP

5-8 その他の機能

ここでは、DB 接続サービスに関連した、NJ/NX シリーズコントローラのバックアップ／リストア機能関連、Sysmac Studio からの操作権限認証、および暗号化通信について説明します。

5-8-1 DB 接続サービスにおける、バックアップ／リストア機能

バックアップ機能とは、NJ/NX シリーズコントローラ内の各種設定データをまとめて SD メモリカードまたはパソコンに保存する機能です。
リストア機能とは、SD メモリカードまたはパソコンからコントローラに復元する機能です。

バックアップ／リストア機能の対象データ

DB 接続サービスの各データがコントローラのバックアップ／リストア機能の対象かどうかを、以下に示します。

データ		バックアップ／リストア機能の対象／対象外	対象の機能の種類	備考
DB 接続設定	DB 接続サービス設定 DB コネクション設定	対象	バックアップ／リストア	バックアップ機能でのデータグループは、“ユーザプログラム・設定”です。
	サーバ証明書			
イベントログ			バックアップのみ	バックアップ機能でのデータグループは、“イベントログ”です。
稼働ログ		対象外	－	以下の参考を参照してください。
スプール				リストア時にスプールをクリアします

参考

稼働ログは、バックアップ／リストアの対象外です。
CPU ユニットのハード交換後も、稼働ログの内容を引き継ぎたい場合は、リストア完了後、使用していた SD メモリカードをリストア先の CPU ユニットに装着してください。

バックアップ／リストア機能が使用できる CPU ユニットの組み合わせ

バックアップ／リストア機能は、以下の CPU ユニットの組み合わせで使用可能です。

バックアップ元	リストア先
形 NX701-□□20	形 NX701-□□20
形 NX502-1□00	形 NX502-1□00
形 NX102-1□20	形 NX102-1□20
形 NX102-9020	形 NX102-9020
形 NJ501-1□20	形 NJ501-1□20
形 NJ501-4320	形 NJ501-4320
形 NJ101-1□20	形 NJ101-1□20
形 NJ101-9020	形 NJ101-9020
形 NJ501-1□20	形 NX701-□□20

ただし、以下の場合は使用できません。
- リストア先の CPU ユニットのユニットバージョンが、バックアップ元の CPU ユニットのユニットバージョンより古い場合
- リストア先の DB 接続サービスのバージョンが、バックアップ元の DB 接続サービスのバージョンより古い場合

バックアップ／リストアが使用できない CPU ユニットの組み合わせでリストアしようとした場合、イベントログに“リストア開始失敗”が登録されます。
DB 接続機能をもたない標準の CPU ユニットと、DB 接続 CPU ユニットとの間には、変数／メモリのバックアップ機能の互換性はありません。

5-8-2 DB 接続サービスにおける、操作権限認証

操作権限認証とは、操作権限に応じて、Sysmac Studio からの CPU ユニットへのオンラインでの操作機能を制限する機能です。
以下に、DB 接続サービスに関連した Sysmac Studio からの操作権限認証機能について、示します。
操作権限認証機能の詳細は、『NJ/NX シリーズ　CPU ユニット ユーザーズマニュアル　ソフトウェア編（SBCA-467)』および『Sysmac Studio Version 1 オペレーションマニュアル (SBCA-470)』を参照してください。

DB 接続サービスにおける、オンライン操作の種類ごとの操作の可否は、以下のとおりです。

○：操作可能、△：操作時に都度認証実施、×：操作不可能

ステータスモニタ機能	管理者	設計者	保全者	操作者	観察者
DB 接続サービスモニタ	○	○	○	○	○
コネクションモニタテーブル	○	○	○	○	○

コントローラ操作機能	管理者	設計者	保全者	操作者	観察者
稼働ログの表示	○	○	○	○	×
稼働ログのクリア	○	○	○	×	×
DB 接続サービスの開始・停止	○	○	×	×	×
DB 接続サービスの終了	○	○	×	×	×
デバッグログの開始・停止	○	○	△	×	×
スプールデータのクリア	○	○	×	×	×

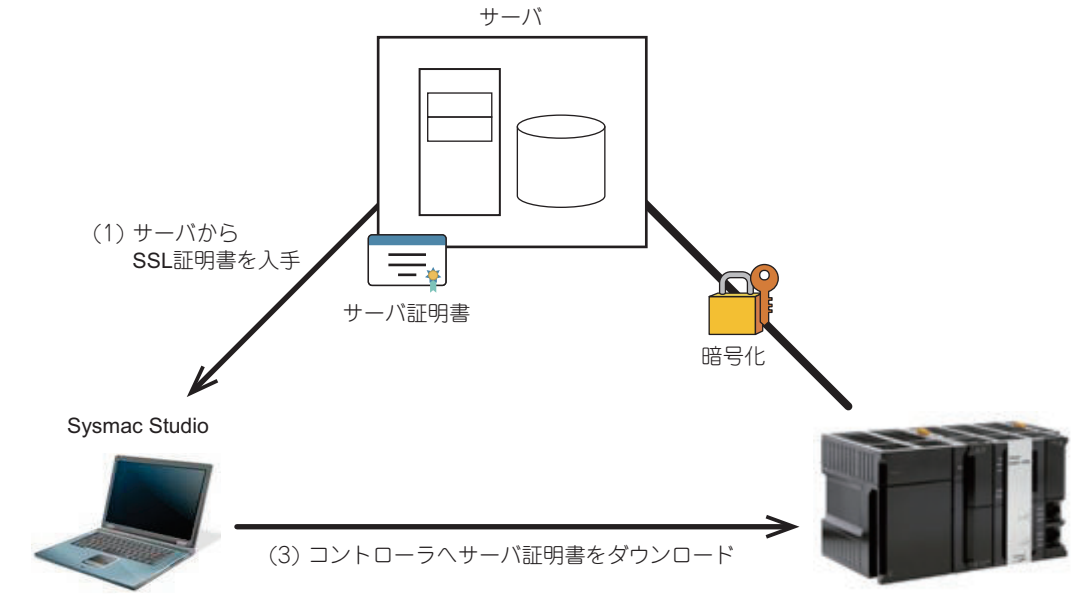
DB 接続テスト機能	管理者	設計者	保全者	操作者	観察者
通信テスト	○	○	○	×	×

5-8-3 暗号化通信

暗号化通信は、コントローラと DB の間の通信データを暗号化することによって、盗聴や改ざんを防止する機能です。
DB 接続サービスでは、SSL/TLS 暗号化通信を用いて暗号化通信を行います。
暗号化通信は、DB 接続サービスのバージョンが Ver.2.00 以降で、Sysmac Studio の DB コネクション設定を行うことによって使用できます。
暗号化通信の設定の詳細については、「2-2-2 DB コネクション設定 (P.2-7)」を参照してください。

暗号化通信のしくみと仕様

以下に、SSL 証明書を用いた暗号化通信のしくみを説明します。



(1) サーバが利用している SSL 証明書を入手して、Sysmac Studio がインストールされているパソコンにコピーする。
(2) Sysmac Studio の DB コネクション設定画面で暗号化を有効にして、SSL 証明書を選択する。
(3) Sysmac Studio の通常転送（ダウンロード）で DB 設定ファイルおよび SSL 証明書をダウンロードする。
(4) ダウンロード後、プログラムを実行して DB 接続命令を実行する

【証明書を更新したい場合】
(5) Sysmac Studio の DB コネクション設定画面で再度、SSL 証明書を選択する。
(6) Sysmac Studio のダウンロードで DB コネクション設定の内容および変更後の SSL 証明書をダウンロードする。
【証明書を削除したい場合】
(5) Sysmac Studio の DB コネクション設定画面で暗号化を無効にする。
(6) Sysmac Studio でダウンロードする。

暗号化通信の仕様は、以下のとおりです。

項目	仕様
サポートする DB の種別	SQL Server Oracle Database MySQL Community Edition PostgreSQL
プロトコル	TLS 1.2
暗号化スイート	サポートする DB の種別において使用可能な暗号化スイートのうち、以下のもの • RFC5246 (TLS1.2) において mandatory（必須）とされているもの • 優先度（適用順）の最高位に位置するもの（最も推奨される暗号化スイート）
通信テスト	暗号化通信を使用する場合の通信テストも可能

項目		仕様
サーバ証明書	最大数（全体）	15
	証明書チェーン	以下を 5 階層まで使用可能 • サーバ証明書 • ルート証明書 • 中間 CA 証明書
	フォーマット	X.509 準拠
	ファイル名に使用可能な文字	「」で囲んだ以下の文字。"～"はその範囲の文字を示します。 「0～9」「A～Z」「a～z」「%」「'」「-」「_」「@」「!」「(」「)」「~」「#」「&」「^」「{」「}」「=」「[」「]」「+」「;」「,」「.」「$」「`」
	CA 証明書	サポート Sysmac Studio の DB コネクション設定画面で証明書を設定する
	バックアップ・リストア	対象

参考

サポートする DB のバージョンとプロトコルの関係は、以下のとおりです。
詳細は対応する DB をリリースしている各社のマニュアルを参照してください。

（○：対応、×：非対応、△：条件付対応）

DB の種別	バージョン	プロトコル(TLS1.2)対応
SQL Server	2012	△（アップデートが必要）
	2014	△（アップデートが必要）
	2016	○
	2017	○
	2019*1	○
Oracle Database	11g	×
	12c	○
	18c	○
	19c*1	○
MySQL Community Edition	5.6	×
	5.7	△（パッチ適用が必要）
	8.0	○
PostgreSQL	9.4	×
	9.5	○
	9.6	○
	10	○
	11*1	○
	12*1	○
	13*1	○

*1. SQL Server 2019、Oracle Database 19c、PostgreSQL 11/12/13 は、DB 接続サービスのバージョン Ver.2.01 以降で使用可能です。

使用手順

使用手順は以下のとおりです。この使用手順は、DB の種別によらず共通です。
実行処理の設定の詳細については、「2-2-2 DB コネクション設定 (P.2-7)」を参照してください。

手順	操作対象	実行処理	備考
1	ツール	DB コネクション設定画面で、暗号化通信を ［使用する］（初期値：［使用しない］）を設定する	暗号化通信を使用する場合のみ必要
2	ツール	DB コネクション設定画面で、サーバ証明書を設定する	暗号化通信を使用する場合のみ必要
3	ツール	コントローラへダウンロードする	
4	コントローラ	DB 接続命令を実行する	自動的に DB との通信が暗号化されます

サーバ証明書を設定したときの動作

暗号化通信機能を使用する場合の、サーバ証明書を設定したときの動作については、以下のとおりです。

データベース種別	サーバ証明書の設定要否	動作
Oracle	必須	SSL/TLS 通信を確立するときに、設定したサーバ証明書が DB サーバのサーバ証明書と一致しない場合、DB_Connect 命令が失敗します。
SQL Server	省略可	サーバ証明書を設定した場合、SSL/TLS 通信を確立するときに、設定したサーバ証明書は使用されません。このため設定したサーバ証明書が DB サーバのサーバ証明書と一致しない場合も、DB_Connect 命令は成功します。（DB 接続が可能です）
MySQL		
PostgreSQL		

6

稼働ログの使用手順

この章では、DB 接続サービスの動作をトレースする稼働ログの使用手順について説明しています。

6-1 稼働ログとは 6-2

6-2 実行ログ 6-3
6-2-1 概要 6-3
6-2-2 使用手順 6-3
6-2-3 実行ログの設定 6-3
6-2-4 実行ログの確認 6-4
6-2-5 実行ログファイルの仕様 6-4

6-3 デバッグログ 6-14
6-3-1 概要 6-14
6-3-2 使用手順 6-14
6-3-3 デバッグログの設定 6-14
6-3-4 デバッグログの開始 6-15
6-3-5 デバッグログの停止 6-16
6-3-6 デバッグログの確認 6-17
6-3-7 デバッグログファイルの仕様 6-17

6-4 SQL 実行失敗ログ 6-26
6-4-1 概要 6-26
6-4-2 使用手順 6-26
6-4-3 SQL 実行失敗ログの設定 6-26
6-4-4 SQL 実行失敗ログの確認 6-27
6-4-5 SQL 実行失敗ログファイルの仕様 6-27

6-5 SD メモリカードへの保存および SD メモリカードの交換 6-32
6-5-1 各稼働ログファイルの SD メモリカードへの保存条件 6-32
6-5-2 DB 接続サービスのディレクトリ構成 6-32
6-5-3 SD メモリカード交換時の各稼働ログ機能の動作 6-32
6-5-4 SD メモリカードの交換作業の目安時間 6-33
6-5-5 SD メモリカードの交換時期 6-34

6-6 稼働ログを確認する 6-35
6-6-1 稼働ログの確認方法 6-35
6-6-2 Sysmac Studio の稼働ログ表示による確認 6-35
6-6-3 SD メモリカード経由での確認 6-37
6-6-4 FTP クライアントソフトでの取り出し確認 6-37

6-1 稼働ログとは

稼働ログとは、DB 接続サービスの CPU ユニット上の動作をトレースするためのログです。CPU ユニット装着の SD メモリカードに保存されます。

稼働ログには、以下の 3 種類があります。

稼働ログの種類	説明
実行ログ	DB 接続機能の実行履歴を確認するために、DB 接続サービスの実行を記録するログです。
デバッグログ	デバッグを支援するために、SQL 実行内容、SQL 実行結果、ユーザ指定ログを記録するログです。
SQL 実行失敗ログ	DB 側での SQL 文の実行失敗に関する情報を確認するために、送信した SQL 文とエラー情報を記録するログです。

6-2 実行ログ

ここでは、稼働ログの 1 つである、DB 接続サービスの実行をトレースする“実行ログ”について説明します。

6-2-1 概要

実行ログでは、DB 接続サービスの開始または停止、DB への接続または切断、SQL 文実行の成功または失敗の履歴により、DB 接続サービスに指示した処理が実行されていることを確認できます。

実行ログは、Sysmac Studio の DB 接続サービス設定で［実行ログ］を［記録する］に設定することで、記録可能となります。DB_PutLog（稼働ログ記録）命令を実行することで、任意のログの記録も可能です。

実行ログを記録する場合、DB 接続サービス動作中は常に、実行ログファイルが CPU ユニットに装着した SD メモリカードに保存されます。

実行ログは CPU ユニットの内部バッファ（揮発性）に一時的に記録され、順次 SD メモリカードに保存されます。SD メモリカードを交換時は、CPU ユニットの内部バッファ（揮発性）に一時的に記録されたままとなります。SD メモリカードを装着した時点で、内部バッファに一時的に記録した実行ログが SD メモリカードに自動的に保存されます。詳細は、「6-5-3 SD メモリカード交換時の各稼働ログ機能の動作 (P.6-32)」を参照してください。

実行ログの内容は、Sysmac Studio の稼働ログ表示の［実行ログ］タブで確認できます。

使用上の注意

実行ログを使用する場合、必ず CPU ユニットに SD メモリカードを装着してください。
実行ログは CPU ユニットの内部バッファに一時的に記録され、順次 SD メモリカードに保存されます。CPU ユニットの電源 OFF またはシャットダウン処理時に SD メモリカードが装着されていない場合は、内部バッファに記録された実行ログは失われます。

6-2-2 使用手順

実行ログは、以下の手順で使用します。

手順	参照
1．実行ログの設定	「6-2-3 実行ログの設定 (P.6-3)」
↓	
2．実行ログの確認	「6-6 稼働ログを確認する (P.6-35)」

6-2-3 実行ログの設定

Sysmac Studio の［構成・設定］－［上位接続設定］－［DB 接続］の下の［DB 接続サービス設定］にて、以下を設定します。

設定	説明	値
実行ログ	実行ログを記録するかを設定します。	• 記録する（デフォルト）*1 • 記録しない
ログのファイル数	実行ログの最大ファイル数を設定します。 最大ファイル数に達した場合は、一番古いファイルを削除して、新しいファイルを作成します。	2～100 （デフォルト：48）
ログのレコード数	実行ログファイルの 1 ファイルに保存可能なレコード数を設定します。 設定したレコード数に達すると、新しいファイルを作成し保存を開始します。	100～65536 （デフォルト：7200）

*1. 形 NX502-1□00 の場合、デフォルトは［記録しない］になります。

DB_PutLog（稼働ログ記録）命令により、任意のログを記録することができます。DB_PutLog（稼働ログ記録）命令で記録したログをユーザ指定ログと呼びます。
ユーザ指定ログを記録するには、DB_PutLog（稼働ログ記録）命令で、ログの種類を“実行ログ”とし、ログコード、ログ名、およびログメッセージを指定します。DB_PutLog（稼働ログ記録）命令の詳細は、「第 7 章 DB 接続命令 (P.7-1)」を参照してください。

6-2-4 実行ログの確認

実行ログの確認方法は、「6-6 稼働ログを確認する (P.6-35)」を参照してください。

6-2-5 実行ログファイルの仕様

実行ログファイルの仕様は、以下のとおりです。
- 実行ログファイルは、複数のレコードで構成されます。
- 1 レコードは 1 行で表現されます。
- 1 個の実行ログファイルに格納するレコードの最大数は、Sysmac Studio で設定します。
- 1 レコードのサイズは、DB 接続サービスのバージョンが Ver.1.04 以前では最大 256 バイト、Ver.2.00 以降では最大 58KB です。
- ファイル名とファイルの種類は、以下のとおりです。

ファイル名	ファイルの種類
DB_ExecutionLog.log	ログ中の最新のログファイル
DB_ExecutionLog_[年月日時分秒ミリ秒].log*1 例： DB_ExecutionLog_20120724220915040.log	過去のログファイル
DB_ExecutionLog.fjc	ログファイルの管理用ファイル

*1. ファイル名に含まれる時刻情報は、CPU ユニットのシステム時刻が使用されます。

- 格納先ディレクトリ（SD メモリカード内）は、以下のとおりです。
  a）ログファイルの格納先：
  /packages/DB_Connection/ExecutionLog/
  b）ログファイルの管理用ファイルの格納先：
  /packages/DB_Connection/System/
- レコードのフォーマットは、以下のとおりです。
  1 レコードは、以下の 1 行から成り、複数のパラメータで構成されます。パラメータとパラメータはタブで区切られます。

[連番]<tab>[日付]<tab>[時刻]<tab>[ミリ秒]<tab>[カテゴリ]<tab>[ログコード]<tab>[ログ名]<tab>[結果]<tab>[DB コネクション名]<tab>[シリアル ID]<tab>[詳細情報]<CR><LF>

パラメータ	サイズ	説明
連番	1～5 バイト	0～65535 65535 を超えたら、0 に戻ります。 複数のファイルにわたって、連続した数字となります（新しいファイルを作成しても、連番は 0 に戻りません）
日付	10 バイト（固定）	レコードが記録された年月日を表示します*1。 YYYY-MM-DD 例：2012-07-23
時刻	8 バイト（固定）	レコードが記録された時分秒を表示します*1 hh:mm:ss 例：15:33:45
ミリ秒	3 バイト（固定）	レコードが記録されたときのミリ秒を 10 進数 3 けたの整数（000～999）で表示します*1 例：10ms の場合：010 623ms の場合：623
カテゴリ	最大 16 バイト（可変）	カテゴリを表示します。 詳細については「カテゴリ (P.6-6)」を参照してください。
ログコード	4 バイト固定	カテゴリ内で一意となる識別コードを 10 進数 4 けたで表示します。 詳細については「ログコード (P.6-6)」を参照してください。
ログ名	最大 32 バイト（可変）	ログの内容を示す名称を表示します。 詳細については「ログ名 (P.6-7)」を参照してください。
結果	6 バイト固定	実行結果を 16 進数 4 けたのコードで表示します（例：0x1234） 0x0000：成功 0x0000 以外：失敗（DB 接続命令の ErrorID と同じコード）
DB コネクション名	最大 16 バイト（可変）	DB コネクション名（半角英数字）を表示します ※カテゴリが“DB 接続サービス”、“ユーザ指定ログ”の場合は、何も表示しません。
シリアル ID	最大 10 バイト（可変）	レコード処理命令、ストアドプロシージャ命令実行時に採番される識別コード（10 進数最大 10 けた）を表示します。 取り得る範囲：0～2147483647 2147483647 を超えた場合、または CPU ユニットの電源投入時に 0 に戻ります。 ※カテゴリが“DB 接続サービス”、“DB コネクション”、“ユーザ指定ログ”の場合は、何も表示しません。

パラメータ	サイズ	説明
詳細情報	可変	実行ログの詳細を表示します。内容はカテゴリにより異なります。 詳細情報内の各情報間はタブで区切られます。 各情報の詳細は、「各情報の詳細 (P.6-8)」を参照してください。 【カテゴリ文字列：DB_CONNECTION（カテゴリ：DB コネクション)】 〔SQL ステータス〕<tab>〔DB エラーコード〕<tab>〔エラーメッセージ〕 【カテゴリ文字列：SQL（カテゴリ：SQL)】 • INSERT/UPDATE/SELECT 時 〔テーブル名〕<tab>〔DB マップ変数名〕<tab>〔DB 応答時間〕<tab>〔DB エラーコード〕 • DELETE 時 〔テーブル名〕<tab>〔DB 応答時間〕<tab>〔DB エラーコード〕 • BATCHINSERT 時 「BATCHINSERT 時の詳細情報 (P.6-8)」を参照してください。 【カテゴリ文字列：PROCEDURE（カテゴリ：ストアドプロシージャ)】 「ストアドプロシージャの詳細情報 (P.6-10)」を参照してください。 【カテゴリ文字列：SQL_RESEND（カテゴリ：SQL 再送)】 〔DB 応答時間〕<tab>〔DB エラーコード〕 【カテゴリ文字列：USER（カテゴリ：ユーザ指定ログ)】 〔ログメッセージ〕
タブ区切り	計 10 バイト	
CR+LF	2 バイト	

*1. 時刻情報は、コントローラの電源を ON したときのタイムゾーンに従います。タイムゾーンを変更した場合、コントローラの電源を再投入してください。

● カテゴリ

カテゴリ	カテゴリ文字列
DB 接続サービス	DB_SERVICE
DB コネクション	DB_CONNECTION
SQL	SQL
ストアドプロシージャ	PROCEDURE
SQL 再送	SQL_RESEND
ユーザ指定ログ	USER

● ログコード

カテゴリ	コード (10 進)	操作	ログを記録するタイミング
DB 接続サービス	0001	DB 接続サービスの開始	DB 接続サービスの開始処理が完了（成功/失敗）したとき
	0002	DB 接続サービスの停止	DB 接続サービスの停止処理が完了（成功/失敗）したとき
	0003	DB 接続サービスのシャットダウン	DB 接続サービスの終了処理が完了（成功/失敗）したとき

カテゴリ	コード (10 進)	操作	ログを記録するタイミング
DB コネクション	0001	DB コネクションの接続	Sysmac Studio、または、命令から DB コネクションの接続を指示して、接続処理が完了（成功/失敗）したとき
	0002	DB コネクションの切断	Sysmac Studio、または、命令から DB コネクションの切断を指示して、切断処理が完了（成功/失敗）したとき
	0003	DB コネクションの異常切断	DB とのコネクションの異常切断を検出したとき
	0004	DB コネクションの再接続	DB とのコネクションの異常切断状態から接続中に復旧したとき
SQL	0001	INSERT	DB_Insert（DB レコード挿入）命令を実行し、DB 接続サービスが DB に発行した INSERT のレスポンス（成功/失敗）が返ってきたとき
	0002	UPDATE	DB_Update（DB レコード更新）命令を実行し、DB 接続サービスが DB に発行した UPDATE のレスポンス（成功/失敗）が返ってきたとき
	0003	SELECT	DB_Select（DB レコード取得）命令を実行し、DB 接続サービスが DB に発行した SELECT のレスポンス（成功/失敗）が返ってきたとき
	0004	DELETE	DB_Delete（DB レコード削除）命令を実行し、DB 接続サービスが DB に発行した DELETE のレスポンス（成功/失敗）が返ってきたとき
	0005	BATCHINSERT	DB_BatchInsert 命令を実行し、DB 接続サービスが DB に発行した BATCHINSERT のレスポンス（成功/失敗）が返ってきたとき
PROCEDURE	0001	ATTACH	DB_AttachProcedure 命令を実行したとき
	0002	EXECUTE	DB_ExecuteProcedure 命令を実行し、DB 接続サービスが DB に発行した PROCEDURE のレスポンス（成功/失敗）が返ってきたとき
	0003	DETACH	DB_DetachProcedure 命令を実行したとき
SQL 再送	0001	INSERT	スプールに保留された INSERT 文を再送し、INSERT のレスポンス（成功/失敗）が返ってきたとき
	0002	UPDATE	スプールに保留された UPDATE 文を再送し、UPDATE のレスポンス（成功/失敗）が返ってきたとき
ユーザ指定ログ	0000-9999（ユーザが指定）	DB_PutLog 命令の実行	DB_PutLog（稼働ログ記録）命令を実行したとき

● ログ名

カテゴリ	操作	ログ名
DB 接続サービス	DB 接続サービスの開始	Start
	DB 接続サービスの停止	Stop
	DB 接続サービスのシャットダウン	Shutdown
DB コネクション	DB コネクションの接続	Connect
	DB コネクションの切断	Close
	DB コネクションの異常切断	Disconnect
	DB コネクションの再接続	Reconnect

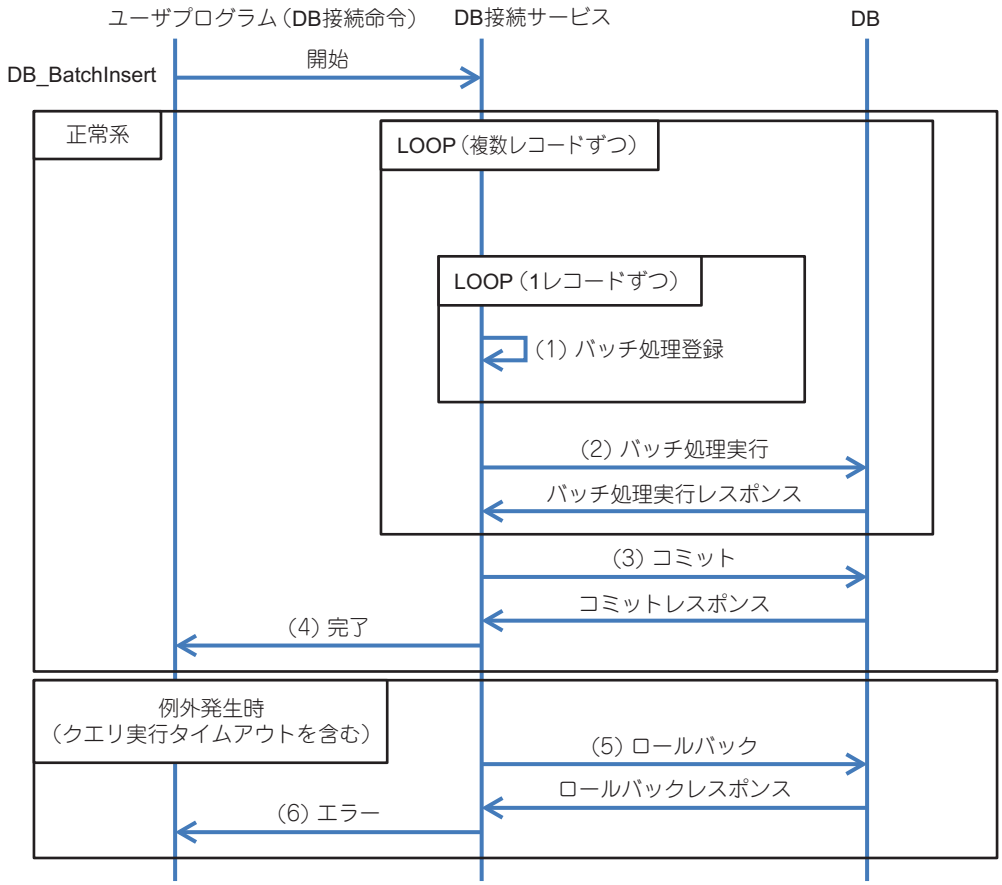
カテゴリ	操作	ログ名
SQL	INSERT	INSERT
	UPDATE	UPDATE
	SELECT	SELECT
	DELETE	DELETE
	BATCHINSERT	BATCHINSERT
PROCEDURE	EXECUTE	EXECUTE
	DETACH	DETACH
	ATTACH	ATTACH
SQL 再送	INSERT	INSERT
	UPDATE	UPDATE
ユーザ指定ログ	DB_PutLog 命令の実行	DB_PutLog 命令の入力変数「LogName」で指定した文字列

● 各情報の詳細

情報	説明
SQL ステータス	SQL 標準規格（ISO/IEC9075、JISX3005-2）で定義されている SQLSTATE 値を表示します。
DB エラーコード	接続対象の DB ベンダ固有のエラーコードを表示します。ネットワーク障害発生時は、0 と表示される場合があります。0 と表示されている場合は、SQL ステータスを確認してください。
エラーメッセージ	1 レコードのサイズ 256 バイトを超えない範囲で先頭文字から表示します。
テーブル名、DB マップ変数名	先頭から最大 60 バイトまで表示します。
DB マップ変数名	入力変数「MapVar」に指定した変数名を表示します（POU インスタンス名は表示しません。DELETE のときは何も表示しません。）
DB 応答時間	ms 単位の整数値で表示します。
DB ログメッセージ	DB_PutLog 命令の入力変数「LogMsg」で指定した文字列を表示します（最大 128 バイト）

BATCHINSERT 時の詳細情報

BATCHINSERT 時の詳細情報を以下に説明します。
BATCHINSERT 時の詳細情報は、下図のタイミング(1)～(6)によって、フォーマットが異なります。



タイミング(1)～(6)の詳細情報のフォーマットは以下のとおりです。

出力タイミング		カテゴリ	ログ名	詳細情報*1
(2)	バッチ処理実行	SQL	BATCHINSERT	[テーブル名]<tab>[DB マップ変数名]<tab>[バッチインサート処理名:REQUEST]<tab>[挿入数]<tab>[処理中配列範囲]
(3)	コミット			[テーブル名]<tab>[DB マップ変数名]<tab>[バッチインサート処理名:COMMIT]
(4)	完了			[テーブル名]<tab>[DB マップ変数名]<tab>[DB 応答時間]<tab>
(5)	ロールバック			[テーブル名]<tab>[DB マップ変数名]<tab>[バッチインサート処理名:ROLLBACK]
(6)	エラー			[テーブル名]<tab>[DB マップ変数名]<tab>[DB 応答時間]<tab>[DB エラーコード]

*1. 各情報の詳細については「各情報の詳細（BATCHINSERT 時）（P.6-9）」を参照してください。

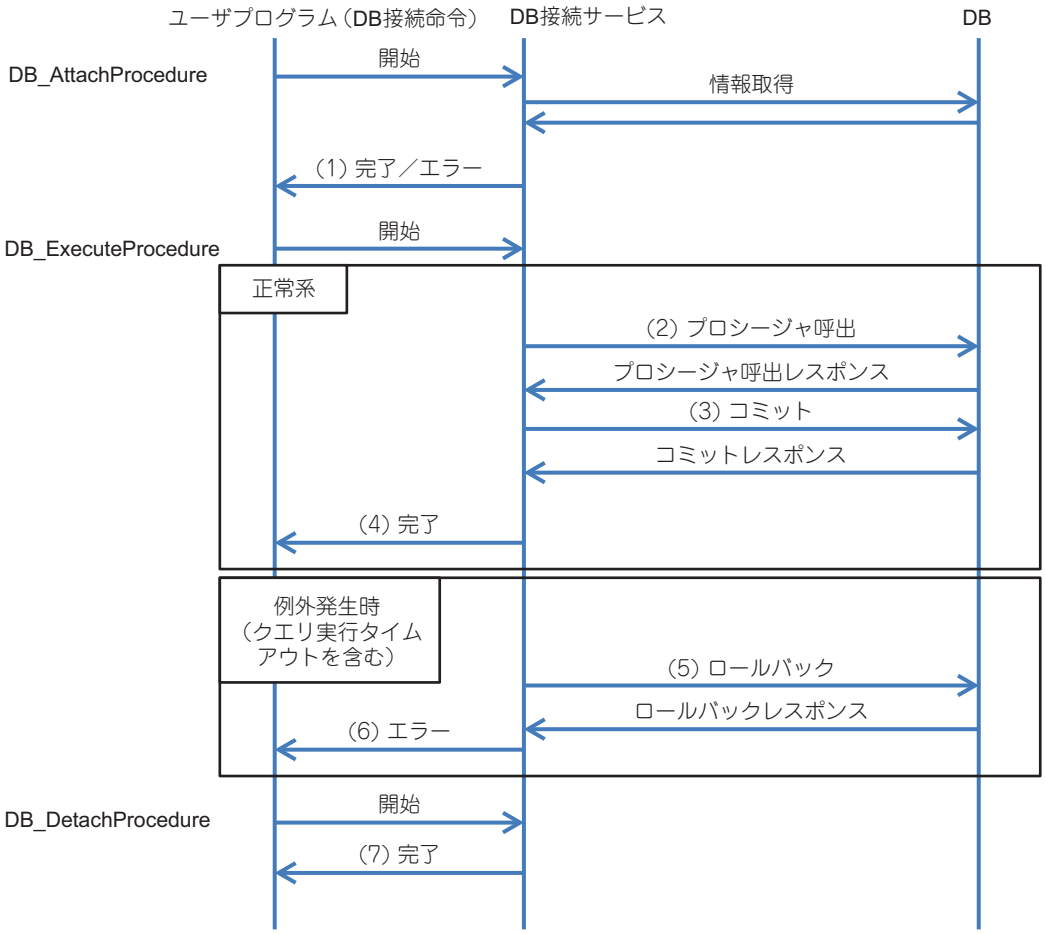
● 各情報の詳細（BATCHINSERT 時）

項目	説明
テーブル名、DB マップ変数名	先頭から最大 60 バイトまで表示します。
バッチインサート処理名	バッチインサートの内部処理を表します。 上図のタイミング(1)、(2)、(3)、(5)によって変わります。 (1)：APPEND：バッチ処理登録 (2)：REQUEST：バッチ処理実行 (3)：COMMIT：コミット (5)：ROLLBACK：ロールバック
挿入数	入力変数「InsertCnt」の値を表示します。DB マップ変数の配列要素数 ≦ InsertCnt または InsertCnt = 0 の場合は、DB マップ変数の配列要素数を表示します。（1～5 バイト）

項目	説明
処理中配列範囲	DB への挿入を要求した配列要素番号の、開始番号と終了番号の差を表示します。 例：配列要素[0..299]の場合は、0..99、100..199、200..299 のように表示します。
DB 応答時間	ms 単位の整数値で表示します。
DB エラーコード	接続対象の DB ベンダ固有のエラーコードを表示します。ネットワーク障害発生時は、0 と表示される場合があります。0 と表示されている場合は、SQL ステータスを確認してください。

ストアドプロシージャの詳細情報

ストアドプロシージャの詳細情報を以下に説明します。
ストアドプロシージャの詳細情報は、下図のタイミング(1)～(7)によって、フォーマットが異なります。



タイミング(1)～(7)における詳細情報のフォーマットは以下のとおりです。

出力タイミング		カテゴリ	ログ名	詳細情報*1
(1)	完了/エラー	PROCEDURE	ATTACH	正常完了： [プロシージャ名]<tab>[プロシージャハンドル]<tab>[Attach エラーコード]<tab>[プロシージャ IF]<tab>[IN 引数マップ変数名]<tab>[OUT 引数マップ変数名]<tab>[INOUT 引数マップ変数名]<tab>[戻り値マップ変数名]<tab>[結果セットマップ変数名] エラー完了： [プロシージャ名]<tab><tab>[Attach エラーコード]<tab>[プロシージャ IF]<tab>[IN 引数マップ変数名]<tab>[OUT 引数マップ変数名]<tab>[INOUT 引数マップ変数名]<tab>[戻り値マップ変数名]<tab>[結果セットマップ変数名]<tab>[Attach エラー箇所]<tab>[Attach エラーメンバ名] CPU ユニット内部バッファ使用時の正常完了及びエラー完了： [プロシージャ名]<tab>[プロシージャハンドル]<tab>[Attach エラーコード]<tab>...
(2)	プロシージャ呼出	PROCEDURE	EXECUTE	[プロシージャ名]<tab>[プロシージャハンドル]<tab>[プロシージャ処理名:CALL]
(3)	コミット			[プロシージャ名]<tab>[プロシージャハンドル]<tab>[プロシージャ処理名:COMMIT]
(4)	完了			[プロシージャ名]<tab>[プロシージャハンドル]<tab>[戻り値]<tab>[DB 応答時間]<tab>
(5)	ロールバック			[プロシージャ名]<tab>[プロシージャハンドル]<tab>[プロシージャ処理名: ROLLBACK]
(6)	エラー			[プロシージャ名]<tab>[プロシージャハンドル]<tab>[DB 応答時間]<tab>[DB エラーコード]
(7)	完了	PROCEDURE	DETACH	[プロシージャ名]<tab>[プロシージャハンドル]

*1. 各情報の詳細については「各情報の詳細（ストアドプロシージャ）（P.6-11）」を参照してください。

● 各情報の詳細（ストアドプロシージャ）

項目	説明
プロシージャ名	先頭から最大 60 バイトまで表示します。 マルチバイト文字使用時は、文字の区切りまで表示します。
プロシージャハンドル	DB_AttachProcedure 命令の出力変数「ProcHandle」に出力した値を表示します。
Attach エラーコード	「[Attach エラーコード]、[Attach エラー箇所]、[Attach エラーメンバ名]の詳細(P.6-12)」を参照してください。
プロシージャ IF	DB から取得したプロシージャ IF を表示します。 フォーマットは以下で出力します。 "戻り値データ型_プロシージャ名(引数方向_引数 1 データ型_引数 1 名,_引数 2 データ型_引数 2 名)" ※_は半角スペース。 引数方向は、IN/OUT/INOUT を出力。 データ型は、すべて DB 側のデータ型で出力。
IN 引数マップ変数名	入力変数「ArgIn」に指定した変数名を表示します。
OUT 引数マップ変数名	入力変数「ArgOut」に指定した変数名を表示します。
INOUT 引数マップ変数名	入力変数「ArgInOut」に指定した変数名を表示します。
戻り値マップ変数名	入力変数「ReturnVal」に指定した変数名を表示します。
結果セットマップ変数名	入力変数「ResultSet」に指定した変数名を表示します。

項目	説明
Attach エラー箇所	「[Attach エラーコード]、[Attach エラー箇所]、[Attach エラーメンバ名]の詳細(P.6-12)」を参照してください。
Attach エラーメンバ名	「[Attach エラーコード]、[Attach エラー箇所]、[Attach エラーメンバ名]の詳細(P.6-12)」を参照してください。
プロシージャ処理名	ストアドプロシージャの内部処理を表します。 上図のタイミング(2)、(3)、(5)によって変わります。 (2)：CALL：プロシージャ呼出 (3)：COMMIT：コミット (5)：ROLLBACK：ロールバック
戻り値	戻り値を表示します。
DB 応答時間	ms 単位の整数値で表示します。
DB エラーコード	接続対象の DB ベンダ固有のエラーコードを表示します。ネットワーク障害発生時は、0 と表示される場合があります。0 と表示されている場合は、SQL ステータスを確認してください。

● [Attach エラーコード]、[Attach エラー箇所]、[Attach エラーメンバ名]の詳細

Attach エラーコード	Attach エラー箇所	Attach エラーメンバ名	説明
Success	-	-	成功
ExistOnlyOneSide*1	ArgIn	-	ストアドプロシージャに IN 引数が存在するが、コントローラの ArgIn に_DBC_Unused が設定されている。
	ArgOut	-	ストアドプロシージャに OUT 引数が存在するが、コントローラの ArgOut に_DBC_Unused が設定されている。
	ArgInOut	-	ストアドプロシージャに INOUT 引数が存在するが、コントローラの ArgInOut に_DBC_Unused が設定されている。
	ReturnVal	-	ストアドプロシージャに戻り値が存在しないのに、コントローラの ReturnVal に_DBC_Unused 以外が設定されている。
TypeNotMatch*2	ArgIn	メンバ名	ストアドプロシージャの IN 引数の型と、コントローラの ArgIn のメンバ変数の型が一致しない。
	ArgOut	メンバ名	ストアドプロシージャの OUT 引数の型と、コントローラの ArgOut のメンバ変数の型が一致しない。
	ArgInOut	メンバ名	ストアドプロシージャの INOUT 引数の型と、コントローラの ArgInOut のメンバ変数の型が一致しない。
	ReturnVal	-	ストアドプロシージャの戻り値の型と、コントローラ側の ReturnVal の型が一致しない。
CountNotMatch	ArgIn	-	ストアドプロシージャの IN 引数の数と、コントローラの ArgIn のメンバ変数の数が一致しない。
	ArgOut	-	ストアドプロシージャの OUT 引数の数と、コントローラの ArgOut のメンバ変数の数が一致しない。
	ArgInOut	-	ストアドプロシージャの INOUT 引数の数と、コントローラの ArgInOut のメンバ変数の数が一致しない。
NameNotMatch	ArgIn	メンバ名	コントローラの ArgIn のメンバ変数の中に、ストアドプロシージャの IN 引数に存在しない名前のものがある。
	ArgOut	メンバ名	コントローラの ArgOut のメンバ変数の中に、ストアドプロシージャの OUT 引数に存在しない名前のものがある。
	ArgInOut	メンバ名	コントローラの ArgInOut のメンバ変数の中に、ストアドプロシージャの INOUT 引数に存在しない名前のものがある。
DBMSSpecific	-	-	ストアドプロシージャの OUT 引数の中にカーソル型が 2 個以上ある(Oracle, PostgreSQL のみ)

*1. SQL Server では、プロシージャの場合、戻り値の処理を記載しない場合も戻り値が int 型と定義されているため、ExistOnlyOneSide ではなく、TypeNotMatch が記録されます。
PostgreSQL では、戻り値を返す必要がない場合 void 型で宣言できます。

void 型で宣言した場合、void 型が返ると判断されるため、ExistOnlyOneSide ではなく、TypeNotMatch が記録されます。

*2. SQL Server では、InOut 型と Out 型の区別がないため、TypeNotMatch エラー時、Attach エラー箇所の ArgOut と ArgInOut が入れ替わることがあります。

レコードの例

- DB 接続サービスの開始：

```
1       2012-07-24 21:29:45          267       DB_SERVICE      0001      Start     0x0000
```

- INSERT（失敗）：

```
1       2012-07-24 21:29:45.267      SQL       0001      INSERT 0x1234   DBConnection1
        45          TableX    VarY   100       17026
```

- ユーザ指定ログ：

```
1       2012-07-24 21:29:45          267       USER      9876      ライン A10x0000
        ”生産開始”
```

ログファイルの例

```
0       2012-07-24          08:29:45 267       DB_SERVICE        0001       Start
        0x0000
1       2012-07-24          08:31:52 002       DB_CONNECTION                0001
        Connect 0x0000      MyDatabase1
2       2012-07-24          08:31:53 959       DB_CONNECTION                0001
        Connect 0x0000      MyDatabase2
3       2012-07-24          09:00:00 052       USER      0001      ライン A10x0000
        ″生産開始″
4       2012-07-24          09:00:00 150       SQL       0001      INSERT 0x0000
        MyDatabase1         0         TABLE_Production            Production      100
        0
5       2012-07-24          09:10:00 150       SQL       0001      INSERT 0x0000
        MyDatabase1         1         TABLE_Production            Production      100
        0
6       2012-07-24          09:20:00 151       SQL       0001      INSERT 0x0000
        MyDatabase1         2         TABLE_Production            Production      100
        0
7       2012-07-24          09:30:00 150       SQL       0001      INSERT 0x0000
        MyDatabase1         3         TABLE_Production            Production      100
        0
8       2012-07-24          09:55:23 422       USER      0002      ライン A10x0000
                ″生産終了″
9       2012-07-24          10:15:00 549       SQL       0003      SELECT 0x0000
        MyDatabase2         4         TABLE_MPS                   ProductionSchedule      200
        0
```

使用上の注意

SD メモリカード内の、ログ中の最新のログファイル（DB_ExecutionLog.log）とログファイルの管理用ファイル（DB_ExecutionLog.fjc）は削除しないでください。削除した場合、実行ログの内容が消失するなど、ログファイルが正しく保存されません。

6-3 デバッグログ

ここでは、稼働ログの 1 つである、DB 接続サービスのデバッグを支援する“デバッグログ”について説明します。

6-3-1 概要

デバッグログでは、どの SQL 文を実行したか、SQL 文ごとのパラメータ、実行結果を確認できます。

デバッグログは、Sysmac Studio の［オンライン設定］の、［デバッグログ］－［開始・停止］で［開始］ボタンをクリックすることで、記録可能となります。DB_PutLog（稼働ログ記録）命令を実行することで、任意のログの記録も可能です。

デバッグログは、デバッグログファイルとして CPU ユニット装着の SD メモリカードに保存されます。
CPU ユニットに SD メモリカードを装着していない状態では、デバッグログは記録できません。

デバッグログの内容は、Sysmac Studio の稼働ログ表示の［デバッグログ］タブで確認できます。

参考

デバッグログは、DB 接続命令により実行した SQL 文のパラメータ、実行結果を確認するログです。スプール再送時、デバッグログには記録されません。スプールに保留された SQL 文を再送した時刻や実行結果は、実行ログで同じシリアル ID のログを確認してください。また、その際の SQL 文のパラメータを確認したい場合、デバッグログで対象の SQL 文がスプールに保留されたときのログを確認してください。

6-3-2 使用手順

デバッグログは、以下の手順で使用します。

手順	参照
1．デバッグログの設定	「6-3-3 デバッグログの設定 (P.6-14)」
↓	
2．デバッグログの開始	「6-3-4 デバッグログの開始 (P.6-15)」
↓	
3．デバッグログの確認	「6-6 稼働ログを確認する (P.6-35)」

6-3-3 デバッグログの設定

Sysmac Studio の［構成・設定］－［上位接続設定］－［DB 接続］の下の［DB 接続サービス設定］で、以下を設定します。

項目	説明	値
ログのファイル数	デバッグログの最大ファイル数を設定します。	1～100 （デフォルト：1）

項目	説明	値
ログのファイルサイズ	1 ファイルの最大サイズを設定します。 設定したファイルサイズを超えるか、1 ファイルのレコード数が 65536 レコードを超えたとき、次のファイルを作成し保存を開始します。	1～100MB （デフォルト：10MB）
ログが上限に達した時の動作	デバッグログファイル数が最大ファイル数に達した場合の動作を設定します。	• ログを停止（デフォルト） • ログを継続（最も古いファイルを削除）
記録開始時ログ削除	記録開始時に SD メモリカード内のデバッグログファイルを削除するかを設定します。	• 削除する（デフォルト） • 削除しない

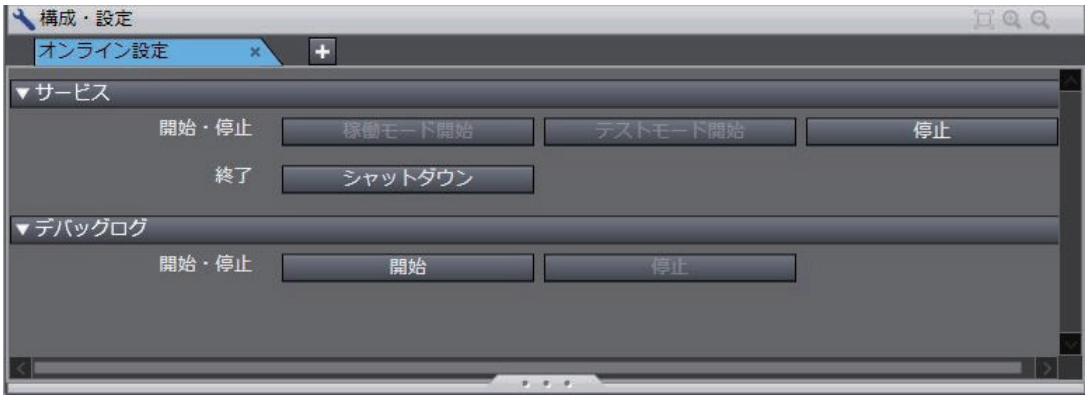
DB_PutLog（稼働ログ記録）命令により、任意のログを記録することができます。DB_PutLog（稼働ログ記録）命令で記録したログをユーザ指定ログと呼びます。
ユーザ指定ログを記録するには、DB_PutLog（稼働ログ記録）命令で、ログの種類を“デバッグログ”とし、ログコード、ログ名、およびログメッセージを指定します。DB_PutLog（稼働ログ記録）命令の詳細は、「第 7 章 DB 接続命令 (P.7-1)」を参照してください。

6-3-4　デバッグログの開始

以下のいずれかにより、デバッグログの記録を開始します。
- Sysmac Studio からのオンライン操作
- DB_ControlService（DB 接続サービスの制御）命令実行

Sysmac Studio からのオンライン操作による開始

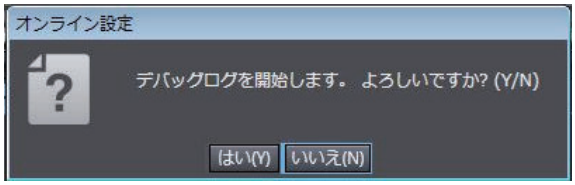
1　Sysmac Studio の［構成・設定］－［上位接続設定］－［DB 接続］－［DB 接続サービス設定］を右クリックし、［オンライン設定］を選択します。
以下の画面が表示されます。



以下のボタン操作により、デバッグログの開始または停止を行うことができます。

分類	項目	ボタン	動作
デバッグログ	開始・停止	［開始］	デバッグログの記録を開始します。
		［停止］	デバッグログの記録を停止します。

2　［開始］ボタンをクリックします。
下記の確認メッセージが表示されます。



3 **［はい（Y)］ボタンをクリックします。**

DB_ControlService 命令実行による開始

DB_ControlService（DB 接続サービスの制御）命令で、入力変数 Cmd にデバッグログ記録開始を指定して実行します。命令の詳細は、「第 7 章 DB 接続命令 (P.7-1)」を参照してください。

6-3-5　デバッグログの停止

以下のいずれかにより、デバッグログの記録を停止します。
- Sysmac Studio からのオンライン操作
- DB_ControlService（DB 接続サービスの制御）命令実行
- 自動的に停止する条件に一致

Sysmac Studio からのオンライン操作による停止

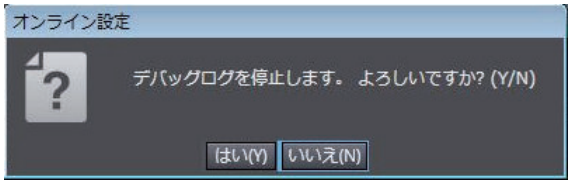
1 **Sysmac Studio の［構成・設定］－［上位接続設定］－［DB 接続］－［DB 接続サービス設定］を右クリックし、［オンライン設定］を選択します。**
以下の画面が表示されます。



以下のボタン操作により、デバッグログの開始または停止を行うことができます。

分類	項目	ボタン	動作
デバッグログ	開始・停止	［開始］	デバッグログの記録を開始します。
		［停止］	デバッグログの記録を停止します。

2 **［停止］ボタンをクリックします。**
下記の確認メッセージが表示されます。



3 **［はい（Y）］ボタンをクリックします。**

DB_ControlService 命令実行による停止

DB_ControlService（DB 接続サービスの制御）命令で、入力変数 Cmd にデバッグログ記録終了を指定して実行します。命令の詳細は、「第 7 章 DB 接続命令 (P.7-1)」を参照してください。

自動的に停止する条件

以下の場合に、自動的にデバッグログの記録を停止します。

- SD メモリカードの給電停止ボタンを押したとき
  ただし、形 NJ501-R□20 は、給電停止ボタンを押してもデバッグログの記録を停止しません。
- Sysmac Studio で同期（ダウンロード）を実行したとき
- メモリオールクリアを実行したとき
- SD メモリカードバックアップ機能、および Sysmac Studio コントローラバックアップ機能のリストアを実行したとき

6-3-6 デバッグログの確認

デバッグログの確認方法は、「6-6 稼働ログを確認する (P.6-35)」を参照してください。

6-3-7 デバッグログファイルの仕様

デバッグログファイルの仕様は、以下のとおりです。

- デバッグログファイルは、複数のレコードで構成されます。
- 1 個のデバッグログファイルに格納する最大ファイルサイズは、Sysmac Studio で設定します。
- 1 レコードのサイズは、最大 58KB です。
- ファイル名とファイルの種類は、以下のとおりです。

ファイル名	ファイルの種類
DB_DebugLog.log	ログ中の最新のログファイル
DB_DebugLog_[年月日時分秒ミリ秒].log*1 例： DB_DebugLog_20120724220915040.log	過去のログファイル
DB_DebugLog.fjc	ログファイルの管理用ファイル

*1. ファイル名に含まれる時刻情報は、CPU ユニットのシステム時刻が使用されます。

- 格納先ディレクトリ（SD メモリカード内）は、以下のとおりです。
  a）ログファイルの格納先：
  /packages/DB_Connection/DebugLog/
  b）ログファイルの管理用ファイルの格納先：
  /packages/DB_Connection/System/
- レコードのフォーマットは、以下のとおりです。
  1 レコードは、以下の 1 行から成り、複数のパラメータで構成されます。パラメータとパラメータはタブで区切られます。

[連番]<tab>[日付]<tab>[時刻]<tab>[ミリ秒]<tab>[カテゴリ]<tab>[ログコード]<tab>[ログ名]<tab>[結果]<tab>[DB コネクション名]<tab>[シリアル ID]<tab>[詳細情報]<CR><LF>

パラメータ	サイズ	説明
連番	1～5 バイト	0～65535 65535 を超えたら、0 に戻ります。 複数のファイルにわたって、連続した数字となります（新しいファイルを作成しても、連番は 0 に戻りません）
日付	10 バイト（固定）	レコードが記録された年月日を表示します[*1]。 YYYY-MM-DD 例：2012-07-23
時刻	8 バイト（固定）	レコードが記録された時分秒を表示します[*1]。 hh:mm:ss 例：15:33:45
ミリ秒	3 バイト（固定）	レコードが記録されたときのミリ秒を 10 進数 3 けたの整数（000～999）で表示します[*1] 例：10ms の場合：010 623ms の場合：623
カテゴリ	最大 16 バイト（可変）	カテゴリを表示します。 詳細については「カテゴリ (P.6-19)」を参照してください。
ログコード	4 バイト固定	カテゴリ内で一意となる識別コードを 10 進数 4 けたで表示します。 詳細については「ログコード (P.6-19)」を参照してください。
ログ名	最大 32 バイト（可変）	ログの内容を示す名称を表示します。 詳細については「ログ名 (P.6-20)」を参照してください。
結果	6 バイト固定	実行結果を 16 進数 4 けたのコードで表示します（例：0x1234） 0x 0000：成功 0x0000 以外：失敗（DB 接続命令の ErrorID と同じコード）
DB コネクション名	最大 16 バイト（可変）	DB コネクション名（半角英数字）を表示します。 ※カテゴリが「DB 接続サービス」、「ユーザ指定ログ」の場合は、何も表示しません。
シリアル ID	最大 10 バイト（可変）	レコード処理命令、ストアドプロシージャ命令実行時に採番する識別コードを表示します。（実行ログ内、カテゴリ“SQL”のレコードで表示されるシリアル ID と同じ ID を表示します） 10 進数最大 10 けた。 取り得る範囲：0～2147483647 2147483647 を超えた場合、または CPU ユニットの電源投入時に 0 に戻ります。 ※カテゴリが“DB 接続サービス”、“DB コネクション”、“ユーザ指定ログ”の場合は、何も表示しません。

パラメータ	サイズ	説明
詳細情報	可変	デバッグログの詳細を表示します。内容はカテゴリにより異なります。 詳細情報内の各情報間はタブで区切られます。 各情報の詳細は、「各情報の詳細 (P.6-21)」を参照してください。 【カテゴリ文字列：SQL（カテゴリ：SQL)】 • INSERT/UPDATE/SELECT/DELETE 時 [テーブル名]<tab>[DB マップ変数名]<tab>[SQL 文] • BATCHINSERT 時 「BATCHINSERT 時の詳細情報 (P.6-21)」を参照してください。 【カテゴリ文字列：PROCEDURE（カテゴリ：PROCEDURE)】 「ストアドプロシージャの詳細情報 (P.6-22)」を参照してください。 【カテゴリ文字列：SQL_RESULT（カテゴリ：SQL 実行結果)】 • INSERT/UPDATE/SELECT/BATCHINSERT 時 [テーブル名]<tab>[DB マップ変数名]<tab>[DB 応答時間]<tab>[DB エラーコード]<tab>[エラーメッセージ] • DELETE 時 [テーブル名]<tab>[DB 応答時間]<tab>[DB エラーコード]<tab>[エラーメッセージ] 【カテゴリ文字列：PROCEDURE_RESULT（カテゴリ：PROCEDURE 実行結果)】 • EXECUTE 時 [プロシージャ名]<tab>[プロシージャハンドル]<tab>[プロシージャの戻り値]<tab>[DB 応答時間]<tab>[DB エラーコード]<tab>[エラーメッセージ] 【カテゴリ文字列：USER（カテゴリ：ユーザ指定ログ)】 [ログメッセージ]
タブ区切り	計 10 バイト	
CR+LF	2 バイト	

*1. 時刻情報は、コントローラの電源を ON したときのタイムゾーンに従います。タイムゾーンを変更した場合、コントローラの電源を再投入してください。

● カテゴリ

カテゴリ	カテゴリ文字列
DB コネクション	DB_CONNECTION
SQL	SQL
ストアドプロシージャ	PROCEDURE
SQL 実行結果	SQL_RESULT
ストアドプロシージャ実行結果	PROCEDURE_RESULT
ユーザ指定ログ	USER

● ログコード

カテゴリ	コード (10 進)	操作	ログを記録するタイミング
DB コネクション	0001	DB コネクションの接続	命令から DB コネクションの接続を指示して、接続処理が完了（成功/失敗）したとき

カテゴリ	コード (10 進)	操作	ログを記録するタイミング
SQL	0001	INSERT	• DB_Insert（DB レコード挿入）命令を実行し、DB 接続サービスが SQL 文を送信する前 • SQL 文がスプールに保留されるとき
	0002	UPDATE	• DB_Update（DB レコード更新）命令を実行し、DB 接続サービスが SQL 文を送信する前 • SQL 文がスプールに保留されるとき
	0003	SELECT	DB_Select（DB レコード取得）命令を実行し、DB 接続サービスが SQL 文を送信する前
	0004	DELETE	DB_Delete（DB レコード削除）命令を実行し、DB 接続サービスが SQL 文を送信する前
	0005	BATCHINSERT	DB_BatchInsert 命令を実行し、DB 接続サービスが SQL 文を送信する前
PROCEDURE	0001	ATTACH	DB_AttachProcedure 命令を実行する前
	0002	EXECUTE	DB_ExecuteProcedure 命令を実行し、DB 接続サービスが SQL 文を送信する前
	0003	DETACH	DB_DetachProcedure 命令を実行する前
SQL 実行結果	0001	INSERT	DB 接続サービスが DB に発行した INSERT のレスポンス（成功/失敗）が返ってきたとき
	0002	UPDATE	DB 接続サービスが DB に発行した UPDATE のレスポンス（成功/失敗）が返ってきたとき
	0003	SELECT	DB 接続サービスが DB に発行した SELECT のレスポンス（成功/失敗）が返ってきたとき
	0004	DELETE	DB 接続サービスが DB に発行した DELETE のレスポンス（成功/失敗）が返ってきたとき
	0005	BATCHINSERT	DB 接続サービスが DB に発行した BATCHINSERT のレスポンス（成功/失敗）が返ってきたとき
PROCEDURE 実行結果	0001	EXECUTE	DB 接続サービスが DB に発行した PROCEDURE のレスポンス（成功/失敗）が返ってきたとき
ユーザ指定ログ	0000-9999（ユーザが指定）	DB_PutLog 命令の実行	DB_PutLog（稼働ログ記録）命令を実行したとき

● ログ名

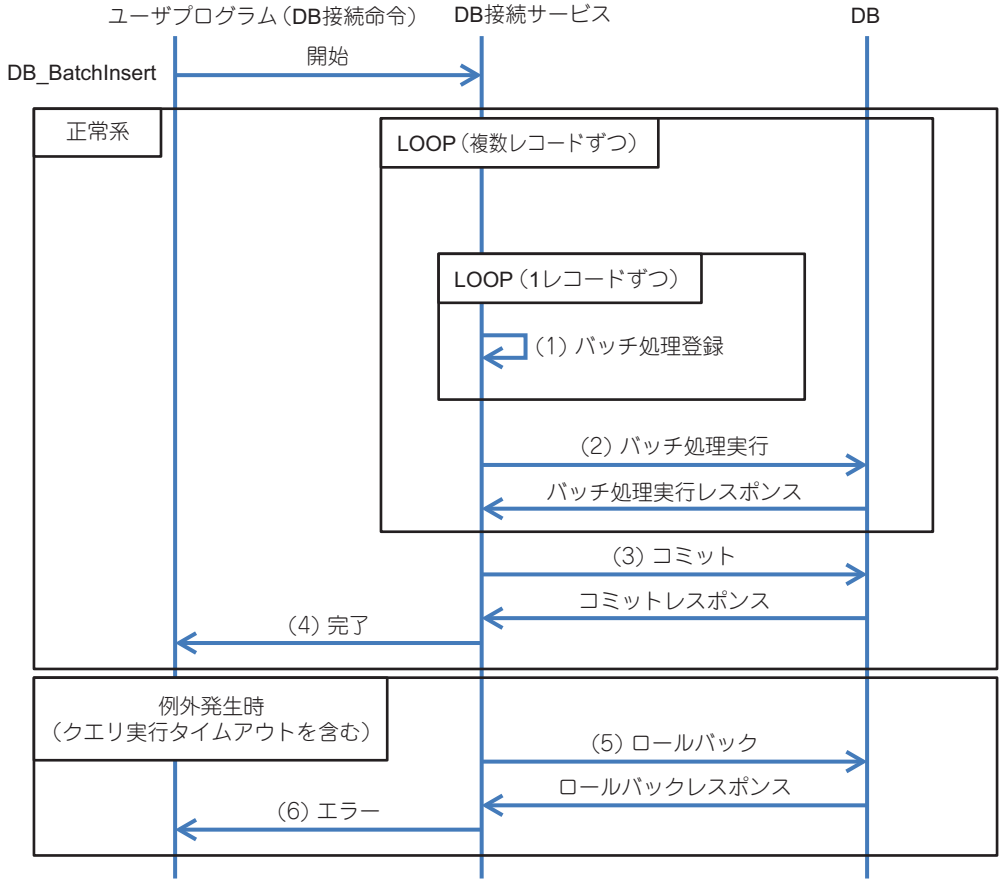
カテゴリ	操作	ログ名
DB コネクション	DB コネクションの接続	Connect
SQL	INSERT	INSERT
	UPDATE	UPDATE
	SELECT	SELECT
	DELETE	DELETE
	BATCHINSERT	BATCHINSERT
PROCEDURE	ATTACH	ATTACH
	EXECUTE	EXECUTE
	DETACH	DETACH
SQL 実行結果	INSERT	INSERT
	UPDATE	UPDATE
	SELECT	SELECT
	DELETE	DELETE
	BATCHINSERT	BATCHINSERT
PROCEDURE 実行結果	EXECUTE	EXECUTE
ユーザ指定ログ	DB_PutLog 命令の実行	DB_PutLog 命令の入力変数「LogName」で指定した文字列

● 各情報の詳細

情報	説明
テーブル名、DB マップ変数名	テーブル名、DB マップ変数名を表示します。
DB マップ変数名	入力変数「MapVar」に指定した変数名を表示します（POU インスタンス名は表示しません。DELETE のときは何も表示しません。）
SQL 文	SQL 文を表示します。
DB 応答時間	ms 単位の整数値で表示します。
DB エラーコード	接続対象の DB ベンダ固有のエラーコードを表示します。ネットワーク障害発生時は、0 と表示される場合があります。0 と表示されている場合は、SQL ステータスを確認してください。
エラーメッセージ	エラーメッセージを表示します。
DB ログメッセージ	DB_PutLog 命令の入力変数「LogMsg」で指定した文字列を表示します（最大 128 バイト）

BATCHINSERT 時の詳細情報

BATCHINSERT 時の詳細情報を以下に説明します。
BATCHINSERT 時の詳細情報は、下図のタイミング(1)～(6)によって、フォーマットが異なります。



タイミング(1)～(6)の詳細情報のフォーマットは以下のとおりです。

出力タイミング		カテゴリ	ログ名	詳細情報*1
(1)	バッチ処理登録	SQL	BATCHINSERT	[テーブル名]<tab>[DB マップ変数名]<tab>[バッチインサート処理名:APPEND]<tab>[要素番号]<tab>[SQL 文]
(2)	バッチ処理実行			[テーブル名]<tab>[DB マップ変数名]<tab>[バッチインサート処理名:REQUEST]<tab>[挿入数]<tab>[処理中配列範囲]
(3)	コミット			[テーブル名]<tab>[DB マップ変数名]<tab>[バッチインサート処理名:COMMIT]
(4)	完了	SQL_RESULT	BATCHINSERT	[テーブル名]<tab>[DB マップ変数名]<tab>[DB 応答時間]<tab><tab>
(5)	ロールバック	SQL	BATCHINSERT	[テーブル名]<tab>[DB マップ変数名]<tab>[バッチインサート処理名:ROLLBACK]
(6)	エラー	SQL_RESULT	BATCHINSERT	[テーブル名]<tab>[DB マップ変数名]<tab>[DB 応答時間]<tab>[DB エラーコード]<tab>[エラーメッセージ]

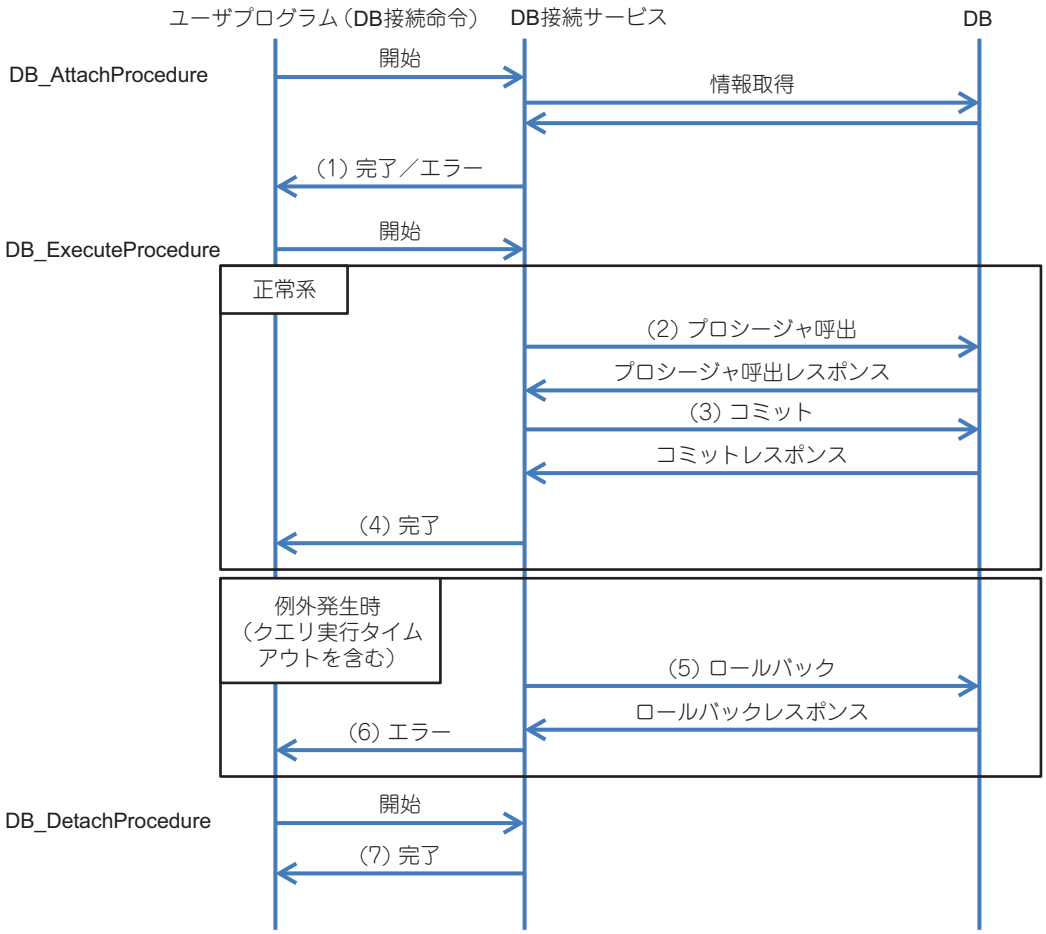
*1. 各情報の詳細については「各情報の詳細（BATCHINSERT 時）（P.6-22)」を参照してください。

● 各情報の詳細（BATCHINSERT 時）

項目	説明
テーブル名、DB マップ変数名	テーブル名、DB マップ変数名を表示します。
バッチインサート処理名	バッチインサートの内部処理を表します。 上図のタイミング(1)、(2)、(3)、(5)によって変わります。 (1)：APPEND：バッチ処理登録 (2)：REQUEST：バッチ処理実行 (3)：COMMIT：コミット (5)：ROLLBACK：ロールバック
要素番号	処理中の配列要素番号を表示します。
SQL 文	SQL 文を出力します。
挿入数	入力変数「InsertCnt」の値を表示します。DB マップ変数の配列要素数 ≦ InsertCnt または InsertCnt = 0 の場合は、DB マップ変数の配列要素数を表示します。(1～5 バイト)
処理中配列範囲	DB への挿入を要求した配列要素番号の、開始番号と終了番号の差を表示します。 例：配列要素[0..299]の場合は、0..99、100..199、200..299 のように表示します。
DB 応答時間	ms 単位の整数値で表示します。
DB エラーコード	接続対象の DB ベンダ固有のエラーコードを表示します。ネットワーク障害発生時は、0 と表示される場合があります。0 と表示されている場合は、SQL ステータスを確認してください。
エラーメッセージ	エラーメッセージを表示します。

ストアドプロシージャの詳細情報

ストアドプロシージャの詳細情報を以下に説明します。
ストアドプロシージャの詳細情報は、下図のタイミング(1)～(7)によって、フォーマットが異なります。



タイミング(1)～(7)における詳細情報のフォーマットは以下のとおりです。

出力タイミング		カテゴリ	ログ名	詳細情報*1
(1)	完了/エラー	PROCEDURE	ATTACH	正常完了： [プロシージャ名]<tab>[プロシージャハンドル]<tab>[Attach エラーコード]<tab>[プロシージャ IF]<tab>[IN 引数マップ変数名]<tab>[OUT 引数マップ変数名]<tab>[INOUT 引数マップ変数名]<tab>[戻り値マップ変数名]<tab>[結果セットマップ変数名] エラー完了： [プロシージャ名]<tab><tab>[Attach エラーコード]<tab>[プロシージャ IF]<tab>[IN 引数マップ変数名]<tab>[OUT 引数マップ変数名]<tab>[INOUT 引数マップ変数名]<tab>[戻り値マップ変数名]<tab>[結果セットマップ変数名]<tab>[Attach エラー箇所]<tab>[Attach エラーメンバ名]
(2)	プロシージャ呼出	PROCEDURE	EXECUTE	[プロシージャ名]<tab>[プロシージャハンドル]<tab>[プロシージャ処理名:CALL]<tab>[プロシージャコール文]
(3)	コミット			[プロシージャ名]<tab>[プロシージャハンドル]<tab>[プロシージャ処理名:COMMIT]
(4)	完了	PROCEDURE_RESULT	EXECUTE	[プロシージャ名]<tab>[プロシージャハンドル]<tab>[戻り値]<tab>[DB 応答時間]<tab><tab>
(5)	ロールバック	PROCEDURE	EXECUTE	[プロシージャ名]<tab>[プロシージャハンドル]<tab>[プロシージャ処理名: ROLLBACK]
(6)	エラー	PROCEDURE_RESULT	EXECUTE	[プロシージャ名]<tab>[プロシージャハンドル]<tab>[DB 応答時間]<tab>[DB エラーコード]<tab>[エラーメッセージ]

出力タイミング		カテゴリ	ログ名	詳細情報*1
(7)	完了	PROCEDURE	DETACH	[プロシージャ名]<tab>[プロシージャハンドル]

*1. 各情報の詳細については「各情報の詳細（ストアドプロシージャ）（P.6-24)」を参照してください。

● 各情報の詳細（ストアドプロシージャ）

項目	説明
プロシージャ名	先頭から最大 60 バイトまで表示します。 マルチバイト文字使用時は、文字の区切りまで表示します。
プロシージャハンドル	DB_AttachProcedure 命令の出力変数「ProcHandle」に出力した値を表示します。
Attach エラーコード	「[Attach エラーコード]、[Attach エラー箇所]、[Attach エラーメンバ名]の詳細(P.6-25)」を参照してください。
プロシージャ IF	DB から取得したプロシージャ IF を表示します。 フォーマットは以下で出力します。 "戻り値データ型_プロシージャ名(引数方向_引数 1 データ型_引数 1 名,_引数 2 データ型_引数 2 名)" ※_は半角スペース。 引数方向は、IN/OUT/INOUT を出力。 データ型は、すべて DB 側のデータ型で出力。
IN 引数マップ変数名	入力変数「ArgIn」に指定した変数名を表示します。
OUT 引数マップ変数名	入力変数「ArgOut」に指定した変数名を表示します。
INOUT 引数マップ変数名	入力変数「ArgInOut」に指定した変数名を表示します。
戻り値マップ変数名	入力変数「ReturnVal」に指定した変数名を表示します。
結果セットマップ変数名	入力変数「ResultSet」に指定した変数名を表示します。
Attach エラー箇所	「[Attach エラーコード]、[Attach エラー箇所]、[Attach エラーメンバ名]の詳細(P.6-25)」を参照してください。
Attach エラーメンバ名	「[Attach エラーコード]、[Attach エラー箇所]、[Attach エラーメンバ名]の詳細(P.6-25)」を参照してください。
プロシージャ処理名	ストアドプロシージャの内部処理を表します。 上図のタイミング(2)、(3)、(5)によって変わります。 (2)：CALL：プロシージャ呼出 (3)：COMMIT：コミット (5)：ROLLBACK：ロールバック
プロシージャコール文	データベース種別ごとのフォーマットで表示します。 プロシージャの IN 引数、プロシージャの OUT 引数は値を出力します。 プロシージャの OUT 引数はプロシージャの OUT 引数名だけを出力します。 例；StoredProc01 という名前で INT 型の IN 引数 param1（値は 10)、INT 型の OUT 引数 param2 という 2 つの引数を持つストアドプロシージャ • SQLServer の場合：EXEC StoredProc01 10, @param2 OUT; • Oracle の場合：StoredProc01(10, :param2); • MySQL の場合：CALL StoredProc01(10, @param2); • PostgreSQL の場合：SELECT StoredProc01(10);
戻り値	戻り値を表示します。
DB 応答時間	ms 単位の整数値で表示します。
DB エラーコード	接続対象の DB ベンダ固有のエラーコードを表示します。ネットワーク障害発生時は、0 と表示される場合があります。0 と表示されている場合は、SQL ステータスを確認してください。
エラーメッセージ	エラーメッセージを表示します。

● [Attach エラーコード]、[Attach エラー箇所]、[Attach エラーメンバ名]の詳細

Attach エラーコード	Attach エラー箇所	Attach エラーメンバ名	説明
Success	-	-	成功
ExistOnlyOneSide	ArgIn	-	ストアドプロシージャに IN 引数が存在するが、コントローラの ArgIn に_DBC_Unused が設定されている。
	ArgOut	-	ストアドプロシージャに OUT 引数が存在するが、コントローラの ArgOut に_DBC_Unused が設定されている。
	ArgInOut	-	ストアドプロシージャに INOUT 引数が存在するが、コントローラの ArgInOut に_DBC_Unused が設定されている。
	ReturnVal	-	ストアドプロシージャに戻り値が存在しないのに、コントローラの ReturnVal に_DBC_Unused 以外が設定されている。
TypeNotMatch	ArgIn	メンバ名	ストアドプロシージャの IN 引数の型と、コントローラの ArgIn のメンバ変数の型が一致しない。
	ArgOut	メンバ名	ストアドプロシージャの OUT 引数の型と、コントローラの ArgOut のメンバ変数の型が一致しない。
	ArgInOut	メンバ名	ストアドプロシージャの INOUT 引数の型と、コントローラの ArgInOut のメンバ変数の型が一致しない。
	ReturnVal	-	ストアドプロシージャの戻り値の型とコントローラ側の ReturnVal の型が一致しない。
CountNotMatch	ArgIn	-	ストアドプロシージャの IN 引数の数と、コントローラの ArgIn のメンバ変数の数が一致しない。
	ArgOut	-	ストアドプロシージャの OUT 引数の数と、コントローラの ArgOut のメンバ変数の数が一致しない。
	ArgInOut	-	ストアドプロシージャの INOUT 引数の数と、コントローラの ArgInOut のメンバ変数の数が一致しない。
NameNotMatch	ArgIn	メンバ名	コントローラの ArgIn のメンバ変数の中に、ストアドプロシージャの IN 引数に存在しない名前のものがある。
	ArgOut	メンバ名	コントローラの ArgOut のメンバ変数の中に、ストアドプロシージャの OUT 引数に存在しない名前のものがある。
	ArgInOut	メンバ名	コントローラの ArgInOut のメンバ変数の中に、ストアドプロシージャの INOUT 引数に存在しない名前のものがある。
DBMSSpecific	-	-	ストアドプロシージャの OUT 引数の中にカーソル型が 2 個以上ある(Oracle, PostgreSQL のみ)

ログファイルの例

```
1 2012-07-24    09:00:00 150    SQL    0001    INSERT 0x0000  MyDatabase1   45
        TABLE_Production        Production     INSERT                          INTO
  TABLE_Production("Column1") VALUES('1000')
2       2012-07-24       09:00:00 200   SQL_RESULT     0001    INSERT 0x300B
        MyDatabase1      46       17072  ORA-17072: Inserted value too large for column
```

使用上の注意

SD メモリカード内の、ログ中の最新のログファイル（DB_DebugLog.log）とログファイルの管理用ファイル（DB_DebugLog.fjc）は削除しないでください。削除した場合、デバッグログの内容が消失するなど、ログファイルが正しく保存されません。

6-4 SQL 実行失敗ログ

ここでは、稼働ログの 1 つである、DB 側が原因で発生した、DB 接続サービスの失敗をトレースする“SQL 実行失敗ログ”について説明します。

6-4-1 概要

SQL 実行失敗ログでは、サーバ側の DB 自体の問題[*1] により、SQL 文の送信に失敗したときに、送信した SQL 文とエラー情報を確認できます。

*1. 例えば、以下のときです。

a）DB 側のテーブルのカラム名を変更したことで、DB 接続サービスから送信した SQL 文のカラム名と一致せず、実行に失敗したとき

b）INSERT する値が、DB 側のテーブルのカラムの型の範囲を超えたとき

SQL 実行失敗ログは、DB 接続サービス設定で“SQL 実行失敗ログ”を“記録する”に設定することで、記録可能となります。

SQL 実行失敗ログは、SQL 実行失敗ログファイルとして CPU ユニット装着の SD メモリカードに保存されます。

CPU ユニットに SD メモリカードを装着していない状態では、SQL 実行失敗ログは記録できません。

SQL 実行失敗ログの内容は、Sysmac Studio の稼働ログ表示の［SQL 実行失敗ログ］タブで確認できます。

6-4-2 使用手順

実行失敗ログは、以下の手順で使用します。

手順	参照
1．実行失敗ログの設定	「6-4-3 SQL 実行失敗ログの設定 (P.6-26)」
2．実行失敗ログの確認	「6-6 稼働ログを確認する (P.6-35)」

6-4-3 SQL 実行失敗ログの設定

Sysmac Studio の［構成・設定］－［上位接続設定］－［DB 接続］の下の［DB 接続サービス設定］で、以下を設定します。

項目	説明	値
SQL 実行失敗ログ	SQL 実行失敗ログを記録するかを設定します。	• 記録する • 記録しない（デフォルト）
ログのファイル数	SQL 実行失敗ログの最大ファイル数を設定します。 最大ファイル数に達した場合は、一番古いファイルを削除して、新しいファイルを作成します。	2～100 （デフォルト：50）
ログのファイルサイズ	1 ファイルの最大サイズを設定します。 設定したファイルサイズを超えるか、1 ファイルのレコード数が 65536 レコードを超えたとき、次のファイルを作成し保存を開始します。	1～100MB （デフォルト:10MB）

6-4-4 SQL 実行失敗ログの確認

SQL 実行失敗ログの確認方法は、「6-6 稼働ログを確認する (P.6-35)」を参照してください。

6-4-5 SQL 実行失敗ログファイルの仕様

SQL 実行失敗ログファイルの仕様は、以下のとおりです。

- SQL 実行失敗ログファイルは、複数のレコードで構成されます。
- 1 レコードは 1 行で表現されます。
- 1 個の SQL 実行失敗ログファイルに格納する最大ファイルサイズは、Sysmac Studio で設定します。
- 1 レコードのサイズは、最大 58KB です。
- ファイル名とファイルの種類は、以下のとおりです。

ファイル名	ファイルの種類
DB_SQLFailedLog.log	ログ中の最新のログファイル
DB_SQLFailedLog_[年月日時分秒ミリ秒].log*1 例： DB_SQLFailedLog_20120724220915040.log	過去のログファイル
DB_SQLFailedLog.fjc	ログファイルの管理用ファイル

*1. ファイル名に含まれる時刻情報は、CPU ユニットのシステム時刻が使用されます。

- 格納先ディレクトリ（SD メモリカード内）は、以下のとおりです。
  a）ログファイルの格納先：
  /packages/DB_Connection/SQLFailedLog/
  b）ログファイルの管理用ファイルの格納先：
  /packages/DB_Connection/System/
- レコードのフォーマットは、以下のとおりです。
  1 レコードは、以下の 1 行から成り、複数のパラメータで構成されます。パラメータとパラメータはタブで区切られます。

[連番]<tab>[日付]<tab>[時刻]<tab>[ミリ秒]<tab>[カテゴリ]<tab>[ログコード]<tab>[ログ名]<tab>[結果]<tab>[DB コネクション名]<tab>[シリアル ID]<tab>[詳細情報]<CR><LF>

パラメータ	サイズ	説明
連番	1～5 バイト	0～65535 65535 を超えたら、0 に戻ります。 複数のファイルにわたって、連続した数字となります（新しいファイルを作成しても、連番は 0 に戻りません）
日付	10 バイト（固定）	レコードが記録された年月日を表示します*1。 YYYY-MM-DD 例：2012-07-23
時刻	8 バイト（固定）	レコードが記録された時分秒を表示します*1。 hh:mm:ss 例：15:33:45
ミリ秒	3 バイト（固定）	レコードが記録されたときのミリ秒を 10 進数 3 けたの整数（000～999）で表示します*1 例：10ms の場合：010 623ms の場合：623
カテゴリ	最大 16 バイト（可変）	カテゴリを表示します。 詳細については「カテゴリ (P.6-28)」を参照してください。
ログコード	4 バイト固定	カテゴリ内で一意となる識別コードを 10 進数 4 けたで表示します。 詳細については、「ログコード (P.6-29)」を参照してください。
ログ名	最大 32 バイト（可変）	ログの内容を示す名称を表示します。 詳細については、「ログ名 (P.6-29)」を参照してください。

パラメータ	サイズ	説明
結果	6 バイト固定	実行結果を 16 進数 4 けたのコードで表示します（例：0x1234） 0x0000：成功 0x0000 以外：失敗（DB 接続命令の ErrorID と同じコード）
DB コネクション名	最大 16 バイト（可変）	DB コネクション名（半角英数字）を表示します。
シリアル ID	最大 10 バイト（可変）	レコード処理命令、ストアドプロシージャ命令実行時に採番する識別コードを表示します。（実行ログ内、カテゴリ“SQL”または“SQL 再送”のレコードで表示されるシリアル ID と同じ ID を表示します）
詳細情報	可変	SQL 実行失敗ログの詳細を表示します。内容はカテゴリにより異なります。 詳細情報内の各情報間はタブで区切られます。 詳細は、「各情報の詳細 (P.6-30)」を参照してください。 【カテゴリ文字列：SQL_FAIL（カテゴリ：SQL 実行失敗）】 • INSERT/UPDATE/SELECT 時 [テーブル名]<tab>[DB マップ変数名]<tab>[DB エラーコード]<tab>[エラーメッセージ]<tab>[SQL 文] • DELETE 時 [テーブル名]<tab>[DB エラーコード]<tab>[エラーメッセージ]<tab>[SQL 文] • BATCHINSERT 時 「BATCHINSERT 時の詳細情報 (P.6-30)」を参照してください。 【カテゴリ文字列：PROCEDURE_FAIL（カテゴリ：PROCEDURE 実行失敗）】 「ストアドプロシージャの詳細情報 (P.6-31)」を参照してください。 【カテゴリ文字列：SPOOL（カテゴリ：スプールに保留）】 [テーブル名]<tab>[DB マップ変数名]<tab>[SQL 文] 【カテゴリ文字列：STATUS_ERROR（カテゴリ：状態異常）】 • INSERT/UPDATE/SELECT/DELETE 時 [テーブル名]<tab>[DB マップ変数名]<tab>[SQL 文] • BATCHINSERT 時 「BATCHINSERT 時の詳細情報 (P.6-30)」を参照してください。 • EXECUTE 時 「ストアドプロシージャの詳細情報 (P.6-31)」を参照してください。
タブ区切り	計 10 バイト	
CR+LF	2 バイト	

*1. 時刻情報は、コントローラの電源を ON したときのタイムゾーンに従います。タイムゾーンを変更した場合、コントローラの電源を再投入してください。

● カテゴリ

カテゴリ	カテゴリ文字列
SQL 実行失敗	SQL_FAIL
ストアドプロシージャ実行失敗	PROCEDURE_FAIL
スプールに保留	SPOOL
状態異常	STATUS_ERROR

● ログコード

カテゴリ	コード（10進）	操作	ログを記録するタイミング
SQL 実行失敗	0001	INSERT	DB 接続サービスが DB に発行した SQL 文が DB 起因で実行失敗したとき
	0002	UPDATE	
	0003	SELECT	
	0004	DELETE	
	0005	BATCHINSERT	
PROCEDURE 実行失敗	0001	EXECUTE	DB 接続サービスが DB に発行した SQL 文が DB 起因で実行失敗したとき
スプールに保留	0001	INSERT	• DB 接続サービスと DB との間の情報交換に障害が発生し、スプールに SQL 文が保留されたとき
	0002	UPDATE	
状態異常	0001	INSERT	• DB 接続サービスがエラーを検出して SQL 文を送信できなかったとき • DB 接続サービスとサーバ側 DB との間の情報交換に障害が発生したとき（スプール無効時） • DB 接続サービスとサーバ側 DB との間の情報交換に障害が発生し、スプールに SQL 文を保留しようとしたが、スプールの空き容量がなく保留できなかったとき
	0002	UPDATE	
	0003	SELECT	• DB 接続サービスがエラーを検出して SQL 文を送信できなかったとき • DB 接続サービスと DB との間の情報交換に障害が発生したとき • スプールに保留された SQL 文が存在するため実行できなかったとき
	0004	DELETE	
	0005	BATCHINSERT	• DB 接続サービスがエラーを検出して SQL 文を送信できなかったとき • DB 接続サービスと DB との間の情報交換に障害が発生したとき • スプールに保留した SQL 文が存在するため実行できなかったとき
	0010	PROCEDURE	

● ログ名

カテゴリ	操作	ログ名
DB 接続サービス	DB 接続サービスの開始	Start
	DB 接続サービスの停止	Stop
	DB 接続サービスのシャットダウン	Shutdown
DB コネクション	DB コネクションの接続	Connect
	DB コネクションの切断	Close
	DB コネクションの異常切断	Disconnect
	DB コネクションの再接続	Reconnect
SQL	INSERT	INSERT
	UPDATE	UPDATE
	SELECT	SELECT
	DELETE	DELETE
	BATCHINSERT	BATCHINSERT
PROCEDURE	EXECUTE	EXECUTE
	DETACH	DETACH
	ATTACH	ATTACH
SQL 再送	INSERT	INSERT
	UPDATE	UPDATE

カテゴリ	操作	ログ名
ユーザ指定ログ	DB_PutLog 命令の実行	DB_PutLog 命令の入力変数「LogName」で指定した文字列

● 各情報の詳細

情報	説明
テーブル名、DB マップ変数名	テーブル名、DB マップ変数名を表示します。
DB マップ変数名	入力変数「MapVar」に指定した変数名を表示します（POU インスタンス名は表示しません。DELETE のときは何も表示しません。）
SQL 文	SQL 文を表示します。
DB エラーコード	接続対象の DB ベンダ固有のエラーコードを表示します。ネットワーク障害発生時は、0 と表示される場合があります。0 と表示されている場合は、SQL ステータスを確認してください。
エラーメッセージ	エラーメッセージを表示します。

BATCHINSERT 時の詳細情報

BATCHINSERT 時の詳細情報を以下に説明します。
SQL 実行失敗ログは、エラー時に出力されます。エラー時に出力される詳細情報は、1 行目はエラー全体についての情報を表し、以降の行は各要素番号についての情報を表します。

出力タイミング	カテゴリ	詳細情報*1
エラー	SQL_FAIL	1 行目：[テーブル名]<tab>[DB マップ変数名]<tab>[ログレコード種別:INFO]<tab>[挿入数]<tab>[DB エラーコード]<tab>[エラーメッセージ] 2 行目：[テーブル名]<tab>[DB マップ変数名]<tab>[ログレコード種別:SQL]<tab>[要素番号]<tab>[SQL 文] : 要素番号+1 行目：[テーブル名]<tab>[DB マップ変数名]<tab>[ログレコード種別:SQL]<tab>[要素番号]<tab>[SQL 文]
	STATUS_ERROR	1 行目：[テーブル名]<tab>[DB マップ変数名]<tab>[ログレコード種別:INFO]<tab>[挿入数] 2 行目：[テーブル名]<tab>[DB マップ変数名]<tab>[ログレコード種別:SQL]<tab>[要素番号]<tab>[SQL 文] : 要素番号+1 行目：[テーブル名]<tab>[DB マップ変数名]<tab>[ログレコード種別:SQL]<tab>[要素番号]<tab>[SQL 文]

*1. 各情報の詳細については「各情報の詳細（BATCHINSERT 時）（P.6-30）」を参照してください。

● 各情報の詳細（BATCHINSERT 時）

項目	説明
テーブル名、DB マップ変数名	テーブル名、DB マップ変数名を表示します。
ログレコード種別	ログレコードの種別を表示します。 • INFO：挿入レコード数/DB エラーコード/エラーメッセージを表示 • SQL：各レコードの SQL 文を表示
挿入数	入力変数「InsertCnt」の値を表示します。（1～5 バイト）
DB エラーコード	接続対象の DB ベンダ固有のエラーコードを表示します。ネットワーク障害発生時は、0 と表示される場合があります。0 と表示されている場合は、SQL ステータスを確認してください。
エラーメッセージ	エラーメッセージを表示します。
要素番号	処理中の配列要素番号を表示します。

項目	説明
SQL 文	SQL 文を表示します。

ストアドプロシージャの詳細情報

ストアドプロシージャの詳細情報を以下に説明します。
SQL 実行失敗ログは、エラー時に出力されます。

出力タイミング	カテゴリ	詳細情報*1
エラー	PROCEDURE_FAIL	[プロシージャ名]<tab>[プロシージャハンドル]<tab>[DB エラーコード]<tab>[エラーメッセージ]<tab>[プロシージャコール文]
	STATUS_ERROR	[プロシージャ名]<tab>[プロシージャハンドル]<tab>[プロシージャコール文]

*1. 各情報の詳細については「各情報の詳細（ストアドプロシージャ）（P.6-31）」を参照してください。

● 各情報の詳細（ストアドプロシージャ）

項目	説明
プロシージャ名	先頭から最大 60 バイトまで表示します。 マルチバイト文字使用時は、文字の区切りまで表示します。
プロシージャハンドル	DB_AttachProcedure 命令の出力変数「ProcHandle」に出力した値を表示します。
DB エラーコード	接続対象の DB ベンダ固有のエラーコードを表示します。ネットワーク障害発生時は、0 と表示される場合があります。0 と表示されている場合は、SQL ステータスを確認してください。
エラーメッセージ	エラーメッセージを表示します。
プロシージャコール文	データベース種別ごとのフォーマットで表示します。 プロシージャの IN 引数、プロシージャの OUT 引数は値を出力します。 プロシージャの OUT 引数はプロシージャの OUT 引数名だけを出力します。 例；StoredProc01 という名前で INT 型の IN 引数 param1（値は 10）、INT 型の OUT 引数 param2 という 2 つの引数を持つストアドプロシージャ • SQLServer の場合：EXEC StoredProc01 10, @param2 OUT; • Oracle の場合：StoredProc01(10, :param2); • MySQL の場合：CALL StoredProc01(10, @param2); • PostgreSQL の場合：SELECT StoredProc01(10);

ログファイルの例

```
1       2012-07-24      09:00:00  200       SQL_FAIL          0001      INSERT    0x300B
        MyDatabase1     0         17072     ORA-17072: Inserted value too large for column
INSERT INTO TABLE_Production(Column1)VALUES('1000')
2       2012-07-24      09:01:13  550       SPOOL     0001      INSERT    0x3012    MyDatabase1
        15      INSERT INTO TABLE_Production(Column2)VALUES('200')
3       2012-07-24      09:01:14  050       SPOOL     0001      INSERT    0x3014    MyDatabase1
        18      INSERT INTO TABLE_Production(Column2)VALUES('300')
4       2012-07-24      09:01:14  550       STATUS_ERROR      0001      INSERT    0x300C
        MyDatabase1     19        INSERT INTO TABLE_Production(Column2) VALUES('400')
```

使用上の注意

SD メモリカード内の、ログ中の最新のログファイル（DB_SQLFailedLog.log）とログファイルの管理用ファイル（DB_SQLFailedLog.fjc）は削除しないでください。削除した場合、SQL 実行失敗ログの内容が消失するなど、ログファイルが正しく保存されません。

6-5 SD メモリカードへの保存および SD メモリカードの交換

DB 接続サービスでは、CPU ユニット装着の SD メモリカードを、稼働ログ機能で使用します。
SD メモリカードには、実行ログファイル、デバッグログファイル、SQL 実行失敗ログファイルが格納されます。
ここでは、稼働ログ機能における SD メモリカードへの保存および SD メモリカードの交換について説明をします。

SD メモリカード機能の詳細は、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル ソフトウェア編（SBCA-467)』を参照してください。

6-5-1 各稼働ログファイルの SD メモリカードへの保存条件

各稼働ログファイルは、以下のときに SD メモリカードに保存されます。

稼働ログ	機能を使用するための操作	SD メモリカードへのログファイルの保存条件
実行ログ	Sysmac Studio の［DB 接続サービス設定］で［実行ログ］を［記録する］に設定する。	DB 接続サービス動作中は常に保存されます。*1
デバッグログ	Sysmac Studio の［DB 接続サービス設定］－［オンライン設定］－［デバッグログ］－［開始・停止］の［開始］ボタンをクリック または DB_ControlService（DB 接続サービスの制御）命令によりデバッグログ記録を開始	デバッグログ記録中は常に保存されます。
SQL 実行失敗ログ	Sysmac Studio の［DB 接続サービス設定］で［SQL 実行失敗ログ］を［記録する］に設定する。	サーバ側 DB の要因による SQL 文の送信失敗時に保存されます。*2

*1. CPU ユニットに SD メモリカードが装着されていない状態で CPU ユニットの電源を投入した場合、実行ログを保存するときにイベントログに“実行ログ保存失敗”が登録されます。SD メモリカードを装着した時点で、実行ログの記録が開始されます。
*2. CPU ユニットに SD メモリカードが装着されていない状態で CPU ユニットの電源を投入した場合、SQL 実行失敗ログを保存するときにイベントログに“SQL 実行失敗ログ保存失敗”が登録されます。SD メモリカードを装着した時点で、SQL 実行失敗ログの記録が開始されます。

6-5-2 DB 接続サービスのディレクトリ構成

DB 接続サービスは、SD メモリカード内の“packages/DB_Connection”ディレクトリ以下を使用します。以下のとおりです。

packages/DB_Connection/System　：ログファイルの管理用ファイルを格納します。
packages/DB_Connection/ExecutionLog　：実行ログファイルを格納します。
packages/DB_Connection/DebugLog　：デバッグログファイルを格納します。
packages/DB_Connection/SQLFailedLog　：SQL 実行失敗ログファイルを格納します。

6-5-3 SD メモリカード交換時の各稼働ログ機能の動作

DB 接続サービス動作中に SD メモリカードを交換する場合、各稼働ログ機能は、以下のように動作します。

稼働ログ機能	SD メモリカード交換中の状態		
	SD メモリカードの給電停止ボタンを押した時点	SD メモリカードが装着されていない間	SD メモリカードを装着した時点
実行ログ	継続します。 CPU ユニットの内部バッファに実行ログがあれば、SD メモリカードに記録されます。	CPU ユニットの内部バッファに一時的に記録されます。	内部バッファに一時的に記録されていたログが、自動的に SD メモリカードに記録されます。
デバッグログ	停止します。*1 CPU ユニットの内部バッファにデバッグログがあれば、SD メモリカードに記録されます。	デバッグログは記録されません。	デバッグログは停止したままです。 Sysmac Studio からのオンライン操作、または DB_ControlService（DB 接続サービスの制御）命令実行により、記録が開始されます。
SQL 実行失敗ログ	停止します。*1 CPU ユニットの内部バッファに SQL 実行失敗ログがあれば、SD メモリカードに記録されます。	SQL 実行失敗ログは記録されません。	SQL 実行失敗ログが自動的に開始されます。

*1. 形 NJ501-R□20 は、給電停止ボタンを押しても停止しません。

使用上の注意

SD メモリカードの交換時は、以下の点に注意してください。

- 交換する SD メモリカードは、あらかじめフォーマットしたものを使用してください。
- 実行ログを記録中は、SD メモリカード給電停止ボタンを押し、SD PWR LED が消えてから 5 分以内に新しい SD メモリカードと交換してください。
  5 分を超える場合、内部バッファに記録した実行ログが失われることがあります。
  SD メモリカードが装着される前に内部バッファの空き容量がなくなった場合は、“実行ログ保存失敗”がイベントログに登録されます。
- 形 NJ501-R□20 は、稼働中の SD メモリカード交換ができません。
  詳細は、『NJ シリーズ ロボット統合 CPU ユニット ユーザーズマニュアル (SBCE-436)』を参照してください。

6-5-4 SD メモリカードの交換作業の目安時間

DB 接続サービス動作中に SD メモリカードを交換する場合、以下の時間内に SD メモリカードを交換してください。SD メモリカードの交換作業の目安時間は、CPU ユニット形式、および DB 接続命令の実行間隔によって異なります。

CPU ユニット形式	DB 接続命令の実行間隔		
	50ms	100ms	500ms
形 NJ501-□□20*1	30 秒	60 秒	300 秒（5 分）
形 NJ101-□□20			
形 NX701-□□20	300 秒（5 分）	600 秒（10 分）	3,000 秒（50 分）
形 NX502-1□00			
形 NX102-□□20	30 秒	60 秒	300 秒（5 分）

*1. 形 NJ501-R□20 は、稼働中の SD メモリカード交換ができません。
詳細は、『NJ シリーズ ロボット統合 CPU ユニット ユーザーズマニュアル (SBCE-436)』を参照してください。

使用上の注意

SD メモリカードの交換時は、以下の点にご注意ください。
- 交換する SD メモリカードは、あらかじめフォーマットしたものを使用してください。
- 実行ログを記録中は、SD メモリカード給電停止ボタンを押し、SD PWR LED が消えてから上記の交換作業の目安時間以内に新しい SD メモリカードと交換してください。
  交換作業の目安時間を超える場合、内部バッファに記録した実行ログが失われることがあります。
  なお、SD メモリカードが装着される前に内部バッファの空き容量がなくなった場合は、“実行ログ保存失敗”がイベントログに登録されます。
- 交換作業の目安時間を超えてしまう場合は、いったん設備を停止するか、DB 接続サービス設定の［実行ログ］を［記録しない］に設定して実行ログが記録されていない状態にしてから、SD メモリカードを交換してください。詳細は、「2-2-1 DB 接続サービス設定 (P.2-5)」を参照してください。

6-5-5 SD メモリカードの交換時期

以下の場合に SD メモリカードを交換してください。
- “SD メモリカード寿命”イベントが発生した。
- システム定義変数「_Card1Deteriorated」（SD メモリカード寿命警告フラグ）の値が TRUE になった。

バージョン関連情報

SD メモリカードの寿命検知機能は CPU ユニットのユニットバージョンと SD メモリカードの組み合わせにより使用可能かどうかが決まります。『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル ソフトウェア編 （SBCA-467)』の「サポートする SD メモリカードおよびフォルダ／ファイル仕様」の説明を参照してください。

6-6 稼働ログを確認する

ここでは、CPU ユニット装着の SD メモリカードに保存された稼働ログを確認する方法について、説明します。

6-6-1 稼働ログの確認方法

稼働ログ（実行ログ、デバッグログ、SQL 実行失敗ログ）は、以下のいずれかの方法で確認できます。

- Sysmac Studio の稼働ログ表示による確認
- SD メモリカード経由での確認
- FTP クライアントソフトでの取り出し確認

使用上の注意

各稼働ログファイルは、UTF-8 の文字コードでエンコードされています。

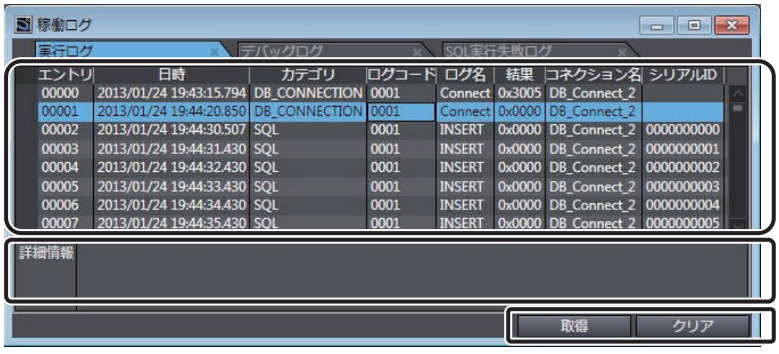
6-6-2 Sysmac Studio の稼働ログ表示による確認

SD メモリカードに保存された稼働ログ（実行ログ、デバッグログ、SQL 実行失敗ログ）は、オンラインで Sysmac Studio の稼働ログ表示から、以下の方法で確認できます。

1 CPU ユニットにオンライン接続している状態で、［構成・設定］－［上位接続設定］－［DB 接続］を右クリックし、［稼働ログ表示］を選択します。

2 実行ログ、デバッグログ、SQL 実行失敗ログは、タブごとに表示されます。

［実行ログタブ］、［デバッグログ］タブ、または［SQL 実行失敗ログ］タブを選択します。
以下の内容が表示されます。



- 一覧表示

項目	内容
エントリ	連番が表示されます。
日時	日時が表示されます。
カテゴリ	カテゴリが表示されます。
ログコード	ログコードが表示されます。
ログ名	ログ名が表示されます。
結果	結果が表示されます。
コネクション名	コネクション名が表示されます。
シリアル ID	シリアル ID が表示されます。

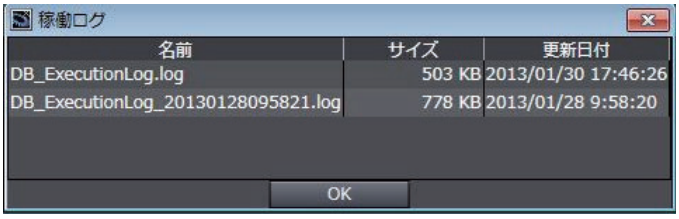
- 詳細情報

ログの［詳細情報］パラメータが表示されます。

- ボタン

［取得］ボタン：

コントローラからログファイルを取得し、表示します。以下の［稼働ログ］ダイアログボックスに、ログファイルの一覧が表示されます。



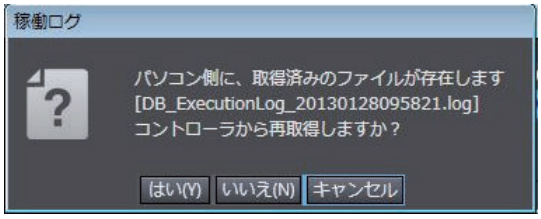
表示したいログファイルを 1 つ選択して［OK］ボタンをクリックします。ログファイルが取得されます。

a)［実行ログ］タブの場合：コントローラの実行ログを取得します。

b)［デバッグログ］タブの場合：コントローラのデバッグログを取得します。

c)［SQL 実行失敗ログ］タブの場合：コントローラの SQL 実行失敗ログを取得します。

（注）1. パソコンに取得済みの同名ログファイルが存在する場合には、以下のメッセージが表示されます。



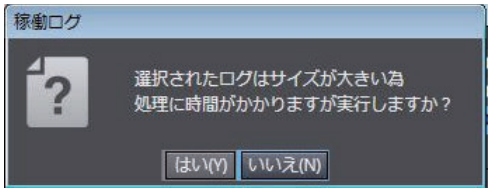
以下のいずれかを選択します。

［はい］ ：コントローラから指定ファイルを取得し表示します

［いいえ］ ：コントローラからファイルを取得せず、パソコンに存在する取得済みファイルの内容を表示します。

［キャンセル］ ：ファイル一覧画面に戻ります。

（注）2. 選択されたログファイルのサイズが 10MB より大きい場合には以下のメッセージが表示されます。



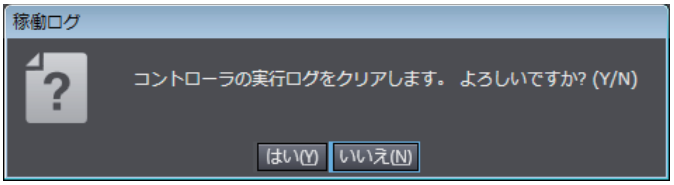
以下のいずれかを選択します。

［はい］ ：コントローラから指定ファイルを取得し表示します

［いいえ］ ：ファイル一覧画面に戻ります。

［クリア］ボタン：

コントローラの稼働ログをクリアします。以下のような確認メッセージが表示されます。



［はい］ボタンをクリックすると、以下の動作となります。

a)［実行ログ］タブの場合：コントローラの実行ログをクリアします。

b)［デバッグログ］タブの場合：コントローラのデバッグログをクリアします。

c)［SQL 実行失敗ログ］タブの場合：コントローラの SQL 実行失敗ログをクリアします。

6-6-3 SD メモリカード経由での確認

CPU ユニット装着の SD メモリカードを抜いて、パソコンに装着し Microsoft Excel やテキストエディタなどで確認できます。

6-6-4 FTP クライアントソフトでの取り出し確認

Ethernet ネットワーク経由で FTP サーバ機能によってログファイルを取り出し、Microsoft Excel やテキストエディタなどで確認できます。
以下の方法によります。
内蔵 EtherNet/IP ポートの FTP サーバ機能を使用します。

1 **Sysmac Studio の［構成・設定］－［内蔵 EtherNet/IP ポート設定］の［FTP 設定］で、［FTP サーバ］を［使用する］に設定します。**

2 **FTP クライアントソフトを使用して、CPU ユニットにログインします。**

3 **稼働ログのファイルを取得します。**
Mget コマンドでワイルドカード指定することで、ログ書込みが完了した複数のログファイルを取得することができます。
例：mget DB_ExecutionLog_*.log

4 **FTP クライアントソフトと CPU ユニットとの接続を解除します。**

5 **取得した稼働ログのファイルを、Microsoft Excel やテキストエディタなどで開き確認します。**

7

DB 接続命令

DB 接続命令と変数 7-2
DB_Connect（DB コネクション接続） 7-6
DB_Close（DB コネクション切断） 7-9
DB_CreateMapping（DB マップ生成） 7-11
DB_Insert（DB レコード挿入） 7-15
DB_Update（DB レコード更新） 7-19
DB_Select（DB レコード取得） 7-35
DB_Delete（DB レコード削除） 7-40
DB_ControlService（DB 接続サービスの制御） 7-54
DB_GetServiceStatus（DB 接続サービスの状態取得） 7-60
DB_GetConnectionStatus（DB コネクションの状態取得） 7-65
DB_ControlSpool（スプールの再送・クリア） 7-71
DB_PutLog（稼働ログ記録） 7-78
DB_Shutdown（DB 接続サービスのシャットダウン） 7-83
DB_BatchInsert（DB レコード一括挿入） 7-87
DB_AttachProcedure（DB ストアドプロシージャ ハンドル生成） 7-99
DB_ExecuteProcedure（DB ストアドプロシージャ 実行） 7-103
DB_DetachProcedure（DB ストアドプロシージャ ハンドル解放） 7-115

DB 接続命令と変数

DB 接続命令の一覧

DB 接続命令の一覧を以下に示します。

命令	名称	対応する DB 接続サービスのバージョン	参照ページ
DB_Connect	DB コネクション接続	Ver.1.00 以降	P.7-6
DB_Close	DB コネクション切断	Ver.1.00 以降	P.7-9
DB_CreateMapping	DB マップ生成	Ver.1.00 以降	P.7-11
DB_Insert	DB レコード挿入	Ver.1.00 以降	P.7-15
DB_Update	DB レコード更新	Ver.1.00 以降	P.7-19
DB_Select	DB レコード取得	Ver.1.00 以降	P.7-35
DB_Delete	DB レコード削除	Ver.1.00 以降	P.7-40
DB_ControlService	DB 接続サービスの制御	Ver.1.00 以降	P.7-54
DB_GetServiceStatus	DB 接続サービスの状態取得	Ver.1.00 以降	P.7-60
DB_GetConnectionStatus	DB コネクションの状態取得	Ver.1.00 以降	P.7-65
DB_ControlSpool	スプールの再送・クリア	Ver.1.00 以降	P.7-71
DB_PutLog	稼働ログ記録	Ver.1.00 以降	P.7-78
DB_Shutdown	DB 接続サービスのシャットダウン	Ver.1.00 以降	P.7-83
DB_BatchInsert	DB レコード一括挿入	Ver.2.00 以降	P.7-87
DB_AttachProcedure	DB ストアドプロシージャ ハンドル生成	Ver.2.00 以降	P.7-99
DB_ExecuteProcedure	DB ストアドプロシージャ 実行	Ver.2.00 以降	P.7-103
DB_DetachProcedure	DB ストアドプロシージャ ハンドル解放	Ver.2.00 以降	P.7-115

DB 接続命令の変数

DB 接続命令で使用する変数の詳細を示します。

DB 接続命令の共通入力変数・出力変数

- **DBConnection**

入力変数	名称	データ型	内容
DBConnection	DB コネクション	DWORD	DB_Connect により出力された DB コネクション。DB 接続機能は DB コネクションを使って命令を実行します。

● ServiceStatus

出力変数名 メンバ名	名称	データ型	内容
ServiceStatus	DB 接続サービスステータス	_sDBC_SERVICE_STATUS	DB 接続サービスのステータスを表す構造体
Status	サービスの状態	_eDBC_STATUS	サービスの状態を表す列挙型 _DBC_STATUS_IDLE(0) ：アイドル _DBC_STATUS_RUNNING(1) ：稼働モードで動作中 _DBC_STATUS_TEST(2) ：テストモードで動作中
DebugLog	デバッグログフラグ	BOOL	デバッグログを記録中の場合 TRUE、停止中の場合 FALSE。
OperatingTime	稼働時間	TIME	サービスが開始してからの経過時間。
ExecCnt	正常実行回数	DINT	すべてのコネクションで正常に SQL 文を実行した回数。
FailedCnt	異常実行回数	DINT	すべてのコネクションで SQL 文の実行に失敗した回数。
SpoolDataCnt	スプールデータ数	DINT	すべてのコネクションでスプールに保留されている SQL 文の数。

● ConnectionStatus

出力変数名 / メンバ名	名称	データ型	内容
ConnectionStatus	DB コネクションステータス	_sDBC_CONNECTION_STATUS	DB コネクションのステータスを表す構造体
Status	接続の状態	_eDBC_CONNECTION_STATUS	DB コネクションの接続状態を表す列挙型 _DBC_CONNECTION_STATUS_CLOSED(0)：未接続 _DBC_CONNECTION_STATUS_CONNECTED(1)：接続中 _DBC_CONNECTION_STATUS_DISCONNECTED(2)：異常切断（接続中にネットワーク障害などにより切断）
ConnectedTime	接続時間	TIME	接続の状態が接続中の時間
DisconnectedTime	異常切断時間	TIME	接続の状態が異常切断の時間
ExecCnt	正常実行回数	DINT	DB コネクションごとで正常に SQL 文を実行した回数。
FailedCnt	異常実行回数	DINT	DB コネクションごとで SQL 文の実行に失敗した回数。
DBRespTime	DB 応答時間	TIME	SQL 文実行時に、CPU ユニットから SQL 文を送信してから、CPU ユニットに SQL 実行結果が返ってくるまでの時間。 DB から正常応答が返ったときのみ格納します。ただし、命令実行タイムアウト発生時は、その命令の実行完了（出力変数 Error が FALSE→TRUE）時点では、DB 応答時間は格納しません（前回の DB 応答時間を保持します）。命令実行タイムアウト発生後、DB から正常応答が返った時点で、DB 応答時間を格納します。
SpoolDataCnt	スプールデータ数	INT	DB コネクションごとでスプールに保留されている SQL 文の数。
SpoolUsageRate	スプール使用率	SINT	DB コネクションあたりのスプールの容量に対する、スプールの使用率。単位は%。
ErrorDateTime	異常切断発生日時	DATE_AND_TIME	最後に異常切断が発生した日時
SQLSTATE	SQL ステータス	STRING(8)	異常発生時[*1] の SQL 標準規格（ISO/IEC9075、JISX3005-2）で定義されたエラーコード[*2]
ErrorCode	エラーコード	DINT	異常発生時[*1] の DB ベンダ固有のエラーコード[*2]
ErrorMsg	エラーメッセージ	STRING(128)	異常発生時[*1] の DB ベンダ固有のエラーメッセージ[*2]

*1. ネットワーク障害発生時、および、SQL 実行エラー発生時
*2. CPU ユニットのユニットバージョンによって、値が異なる場合があります。CPU ユニット Ver.1.08 で、SQL Server の接続エラーの値が変更されています。

● **SendStatus**

出力変数	名称	データ型	内容
SendStatus	送信状態	_eDBC_SEND_STATUS	DB への SQL 文の送信状態を表す列挙型 _DBC_SEND_INIT(0)：初期状態 _DBC_SEND_UNSENT(1)：SQL 文未送信 _DBC_SEND_SENDING(2)：SQL 文送信中 _DBC_SEND_SPOOLED(3)：SQL 文をスプールに保留した _DBC_SEND_COMPLETE(4)：SQL 文送信完了

NJ/NX シリーズ命令共通の変数

入力変数	名称	データ型	内容
Execute	起動	BOOL	信号の立ち上がりで命令を実行する

出力変数	名称	データ型	内容
Done	完了	BOOL	命令が正常終了したことを示す TRUE : 正常終了 FALSE: 異常終了、実行中、または実行条件非成立
Busy	命令実行中	BOOL	命令が実行中であることを示す TRUE : 実行中 FALSE: 実行中ではない
Error	エラー	BOOL	命令が異常終了したことを示す TRUE : 異常終了 FALSE: 正常終了、実行中、または実行条件非成立
ErrorID	エラーコード	WORD	異常終了時の異常 ID 正常終了時は、WORD#16#0

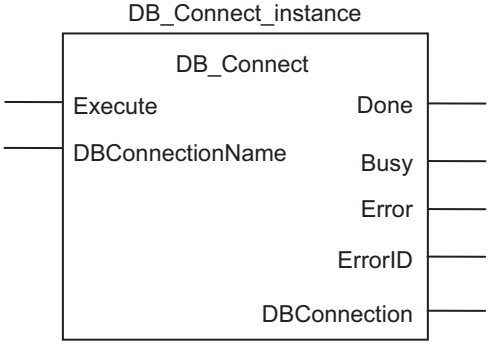
DB 接続サービス関連のシステム定義変数

変数	名称	データ型	内容
_DBC_Status	DB 接続サービスステータス	_sDBC_STATUS	DB 接続サービスのステータスを表すシステム定義変数
_DBC_Unused	DB 接続入力変数省略	BOOL	ストアドプロシージャの引数、戻り値、または、結果セットのいずれかが存在しない場合に、DB_AttachProcedure 命令の入力変数を省略するためのシステム定義変数

システム定義変数の詳細は、「3-5-4 システム定義変数 (P.3-23)」を参照してください。

DB_Connect（DB コネクション接続）

DB に接続します。

命令	名称	FB/FUN	グラフィック表現	ST 表現
DB_Connect	DB コネクション接続	FB		DB_Connect_instance (Execute, DBConnectionName, Done, Busy, Error, ErrorID, DBConnection);

（注）DB_Connect 命令のインスタンスとして、インスタンス名 DB_Connect_instance で変数宣言されている場合の例です。

変数

入力変数

入力変数	名称	データ型	有効範囲	単位	初期値	内容
Execute	起動	BOOL	FALSE, TRUE	-	FALSE	実行条件を指定します
DBConnectionName	DB コネクション名	STRING	最大 17 バイト（終端 NULL 含む）	-	''	Sysmac Studio で設定した DB コネクション名を指定します。

出力変数

出力変数	名称	データ型	有効範囲	単位	内容
Done	完了	BOOL	FALSE, TRUE	-	正常終了を表します
Busy	実行中	BOOL	FALSE, TRUE	-	処理中を表します
Error	エラー	BOOL	FALSE, TRUE	-	異常終了を表します
ErrorID	エラーコード	WORD	16#0000 ～ 16#FFFF	-	エラーコードを表します
DBConnection	DB コネクション	DWORD	16#00000000 ～ FFFFFFFF	-	DB コネクションを出力します。DB_CreateMapping、DB_Insert、DB_Update、DB_Select、DB_Delete、DB_Close 命令はこの DB コネクションを指定します。

関連するシステム定義変数

システム定義変数	名称	データ型	内容
_EIP_EtnOnlineSta	オンライン	BOOL	内蔵 EtherNet/IP ポートの通信機能の状態 TRUE:利用可能 FALSE:利用不可

関連するデバイスパラメータ

NX シリーズに接続した X バスユニットの Ethernet ポートを使用する場合は、X バスユニットのデバイスパラメータを参照してください。

関連するエラーコード一覧

エラーコード	名称	内容
0406Hex	領域範囲外指定	「DBConnectionName」が NULL 文字（16#00）だけの文字列である場合
0410Hex	文字列フォーマット異常	「DBConnectionName」にスペース文字を含む文字列を指定した場合 「DBConnectionName」が終端 NULL で終わらない場合
041DHex	命令同時実行リソースオーバ	DB 接続命令を同時に 32 個を超えて実行した場合
3000Hex	DB 接続サービス未開始	DB 接続サービスが動作中でないときに、本命令を実行場合
3002Hex	DB 接続サービス終了／終了処理中	DB 接続サービスが終了後／終了処理中に、本命令を実行した場合
3003Hex	DB コネクション名不正	「DBConnectionName」が DB コネクション設定に設定されていない場合
3004Hex	DB コネクション接続拒否	DB コネクション設定で設定した DB が接続を拒否した場合
3005Hex	DB コネクション接続失敗	ネットワーク障害などにより DB と通信できない場合 DB コネクション設定で設定したアドレスが間違っている場合
3006Hex	DB コネクション既接続	同一の DB コネクション名で既に DB コネクション接続をしている場合
3007Hex	DB コネクション接続数超過	同時に接続可能な DB コネクションの最大数を超えている場合
3008Hex	DB コネクション不正	同一の DB コネクションに、同時に本命令を実行した場合
3013Hex	DB 接続サービス異常停止中	DB 接続サービスが異常停止中に、本命令を実行した場合
3015Hex	DB 接続サービス初期処理中	DB 接続サービスが初期処理中に、本命令を実行した場合
301EHex	DB 接続サービス未起動	DB 接続サービスを［使用しない］に設定した状態で、本命令を実行した場合

機能説明

DB コネクション名「DBConnectionName」で指定した DB に接続します。

「DBConnectionName」は、Sysmac Studio の DB コネクション設定で設定します。

命令が正常終了(「Done」の値が TRUE に変化) した時点で、DB コネクションが確立しています。命令が正常終了すると DB コネクションに出力します。以降の DB 接続命令の一部は、この DB コネクションを指定します。

使用上の注意

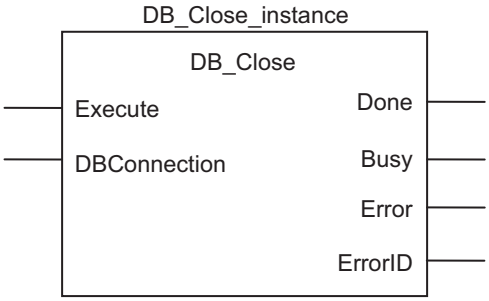
- 本命令は、「Execute」の値が FALSE になっても、また実行時間がタスク周期を超えても、処理を最後まで続けます。処理が正常終了したかどうかは、「Done」の値が TRUE になることで確認してください。
- 「Execute」、「Done」、「Busy」、「Error」のタイミングチャートは、『NJ/NX シリーズコマンドリファレンスマニュアル　基本編』（SBCA-468）の“本章の見方”を参照してください。
- 本命令は、イベントタスク上で使うことができません。コンパイル時にエラーになります。
- 本命令は、NJ/NX シリーズ CPU ユニットの内蔵 EtherNet/IP ポート、および NX シリーズに接続した X バスユニットの Ethernet ポートで使用可能です。NJ/NX シリーズ CPU ユニットに接続した EtherNet/IP ユニット経由で DB に接続することはできません。
- 本命令で生成した DB コネクションは、以下を実行したときに切断されます。
  a）DB_Close 命令、DB_Shutdown 命令を実行したとき
  b）コントローラを運転モードからプログラムモードに変更したとき
  c）DB 接続サービスを停止したとき
- 同時に接続できる DB コネクションの数については、「1-2-1 DB 接続サービスの仕様 (P.1-4)」を参照してください。
- DB 接続サービスをテストモードで開始している場合、本命令は実際には DB への接続を行わず正常終了します。
- 同一の DB コネクション名で既に DB コネクション接続をしている場合は、「DBConnection」に、接続中の DB コネクションを出力します。
- 本命令は、以下の場合、異常となります。「Error」は TRUE になります。
  a）DB 接続サービスが動作中でないときに、本命令を実行した場合
  b）DB 接続サービスが初期処理中に、本命令を実行した場合
  c）DB 接続サービスが異常停止中に、本命令を実行した場合
  d）DB 接続サービスが終了後／終了処理中に、本命令を実行した場合
  e）「DBConnectionName」が DB コネクション設定に設定されていない場合
  f）「DBConnectionName」が NULL 文字（16#00）だけの文字列である場合
  g）「DBConnectionName」にスペース文字を含む文字列を指定した場合
  h）「DBConnectionName」が終端 NULL で終わらない場合
  i） DB コネクション設定で設定したアドレスが間違っていて、接続に失敗した場合
  j） DB コネクション設定で設定した DB が接続を拒否した場合
  k）ネットワーク障害などにより DB と通信できない場合
  l） 同一の DB コネクションに、同時に本命令を実行した場合
  m）同一の DB コネクション名で既に DB コネクション接続をしている場合
  n）同時に接続可能な DB コネクションの最大数を超えている場合
  o）DB 接続命令を同時に 32 個を超えて実行した場合

サンプルプログラム

サンプルプログラムは、DB_Update 命令のサンプルプログラム「サンプルプログラム (P.7-23)」を参照してください。

DB_Close（DB コネクション切断）

DB_Connect 命令で接続した DB との DB コネクションを切断します。

命令	名称	FB/FUN	グラフィック表現	ST 表現
DB_Close	DB コネクション切断	FB		DB_Close_instance (Execute, DBConnection, Done, Busy, Error, ErrorID);

（注）DB_Close 命令のインスタンスとして、インスタンス名 DB_Close_instance で変数宣言されている場合の例です。

変数

入力変数

入力変数	名称	データ型	有効範囲	単位	初期値	内容
Execute	起動	BOOL	FALSE, TRUE	-	FALSE	実行条件を指定します
DBConnection	DB コネクション	DWORD	16#00000000 ～ FFFFFFFF	-	16#00000000	DB_Connect で取得した DB コネクション

出力変数

出力変数	名称	データ型	有効範囲	単位	内容
Done	完了	BOOL	FALSE, TRUE	-	正常終了を表します
Busy	実行中	BOOL	FALSE, TRUE	-	処理中を表します
Error	エラー	BOOL	FALSE, TRUE	-	異常終了を表します
ErrorID	エラーコード	WORD	16#0000 ～ 16#FFFF	-	エラーコードを表します

関連するシステム定義変数

システム定義変数	名称	データ型	内容
_EIP_EtnOnlineSta	オンライン	BOOL	内蔵 EtherNet/IP ポートの通信機能の状態 TRUE:利用可能 FALSE:利用不可

関連するデバイスパラメータ

NX シリーズに接続した X バスユニットの Ethernet ポートを使用する場合は、X バスユニットのデバイスパラメータを参照してください。

関連するエラーコード一覧

エラーコード	名称	内容
041DHex	命令同時実行リソースオーバ	DB 接続命令を同時に 32 個を超えて実行した場合
3000Hex	DB 接続サービス未開始	DB 接続サービスが動作中でないときに、本命令を実行した場合
3002Hex	DB 接続サービス終了／終了処理中	DB 接続サービスが終了後／終了処理中に、本命令を実行した場合
3008Hex	DB コネクション不正	「DBConnection」が不正または既に切断している場合
3013Hex	DB 接続サービス異常停止中	DB 接続サービスが異常停止中に、本命令を実行した場合
3015Hex	DB 接続サービス初期処理中	DB 接続サービスが初期処理中に、本命令を実行した場合
301EHex	DB 接続サービス未起動	DB 接続サービスを［使用しない］に設定した状態で、本命令を実行した場合

機能説明

「DBConnection」で指定した DB コネクションを切断します。

使用上の注意

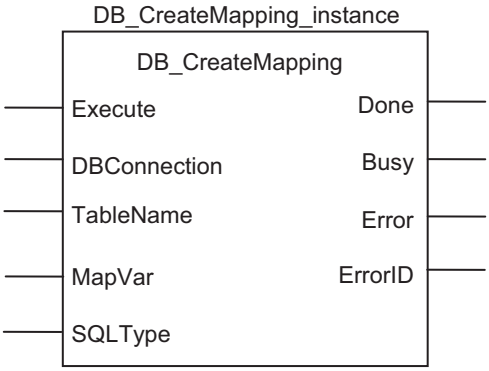
- 本命令は、「Execute」の値が FALSE になっても、また実行時間がタスク周期を超えても、処理を最後まで続けます。処理が正常終了したかどうかは、「Done」の値が TRUE になることで確認してください。
- 「Execute」、「Done」、「Busy」、「Error」のタイミングチャートは、『NJ/NX シリーズコマンドリファレンスマニュアル　基本編』（SBCA-468）の“本章の見方”を参照してください。
- 本命令は、イベントタスク上で使うことができません。コンパイル時にエラーになります。
- DB 接続サービスをテストモードで開始している場合、実際には DB への接続をしていませんが、本命令は正常終了します。
- 本命令は、以下の場合、異常となります。「Error」は TRUE となります。
  a）DB 接続サービスが動作中でないときに、本命令を実行した場合
  b）DB 接続サービスが初期処理中に、本命令を実行した場合
  c）DB 接続サービスが異常停止中に、本命令を実行した場合
  d）DB 接続サービスが終了後／終了処理中に、本命令を実行した場合
  e）「DBConnection」が不正、または既に切断している場合
  f）DB 接続命令を同時に 32 個を超えて実行した場合

サンプルプログラム

サンプルプログラムは、DB_Update 命令のサンプルプログラム「サンプルプログラム (P.7-23)」を参照してください。

DB_CreateMapping（DB マップ生成）

DB のテーブルと DB マップ変数を関連付けします。

命令	名称	FB/FUN	グラフィック表現	ST 表現
DB_CreateMapping	DB マップ生成	FB		DB_CreateMapping_instance (Execute, DBConnection, TableName, MapVar, SQLType, Done, Busy, Error, ErrorID);

（注）DB_CreateMapping 命令のインスタンスとして、インスタンス名 DB_CreateMapping_instance で変数宣言されている場合の例です。

変数

入力変数

入力変数	名称	データ型	有効範囲	単位	初期値	内容
Execute	起動	BOOL	FALSE, TRUE	-	FALSE	実行条件を指定します
DBConnection	DB コネクション	DWORD	16#00000000 ～ FFFFFFFF	-	16#00000000	DB_Connect で取得した DB コネクション
TableName	テーブル名	STRING	データ型に従う*1	-	''	DB のテーブル名
MapVar	DB マップ変数	構造体型、構造体型の配列（全体配列）	データ型に従う	-	-	DB にアクセスするために定義した構造体変数
SQLType	SQL 種別	_eDBC_SQLTYPE	_DBC_SQLTYPE_INSERT(1)：INSERT _DBC_SQLTYPE_UPDATE(2)：UPDATE _DBC_SQLTYPE_SELECT(3)：SELECT _DBC_SQLTYPE_BATCHINSERT(4)：BatchInsert	-	_DBC_SQLTYPE_INSERT	マッピングする変数のレコード処理の種別。

*1. DB 側が大文字／小文字を区別する場合、テーブル名は以下のように指定してください。
MySQL と接続するとき、半角バッククォートで囲む
例）`TableName1`

MySQL 以外と接続するとき、半角ダブルクォートで囲む
例）"TableName1"

出力変数

出力変数	名称	データ型	有効範囲	単位	内容
Done	完了	BOOL	FALSE, TRUE	-	正常終了を表します
Busy	実行中	BOOL	FALSE, TRUE	-	処理中を表します
Error	エラー	BOOL	FALSE, TRUE	-	異常終了を表します
ErrorID	エラーコード	WORD	16#0000 ～ 16#FFFF	-	エラーコードを表します

関連するシステム定義変数

システム定義変数	名称	データ型	内容
_EIP_EtnOnlineSta	オンライン	BOOL	内蔵 EtherNet/IP ポートの通信機能の状態 TRUE:利用可能 FALSE:利用不可

関連するデバイスパラメータ

NX シリーズに接続した X バスユニットの Ethernet ポートを使用する場合は、X バスユニットのデバイスパラメータを参照してください。

関連するエラーコード一覧

エラーコード	名称	内容
0400Hex	入力値範囲外	「SQLType」に列挙子にない値を指定した場合
0406Hex	領域範囲外指定	「TableName」が NULL 文字（16#00）だけの文字列である場合
0410Hex	文字列フォーマット異常	「TableName」にスペース文字を含む文字列を指定した場合
041BHex	データ容量超過	1 つの DB コネクションに、DB マップ変数の数が上限値を超える DB マップ変数を関連付けようとした場合
041DHex	命令同時実行リソースオーバ	DB 接続命令を同時に 32 個を超えて実行した場合
3000Hex	DB 接続サービス未開始	DB 接続サービスが動作中でないときに、本命令を実行した場合
3002Hex	DB 接続サービス終了／終了処理中	DB 接続サービスが終了後／終了処理中に、本命令を実行した場合
3008Hex	DB コネクション不正	「DBConnection」が不正または既に切断している場合
3009Hex	DB マップ変数不正	「MapVar」で指定した変数のデータ型が構造体でない場合 「MapVar」で指定した変数の構造体のメンバに派生データ型が含まれている場合 構造体型の配列指定変数を INSERT 用、UPDATE 用の DB マップ変数「MapVar」に指定した場合 構造体配列でない変数を BATCHINSERT 用の DB マップ変数「MapVar」に指定した場合 1 次元配列でない変数を BATCHINSERT 用の DB マップ変数「MapVar」に指定した場合
300BHex	SQL 実行エラー	実行した SQL 文が DB でエラーとなった場合
3011Hex	DB コネクション異常切断発生中	ネットワーク障害などで DB と通信できない場合

エラーコード	名称	内容
3013Hex	DB 接続サービス異常停止中	DB 接続サービスが異常停止中に、本命令を実行した場合
3015Hex	DB 接続サービス初期処理中	DB 接続サービスが初期処理中に、本命令を実行した場合
3019Hex	対象外データベース種別への命令実行	本命令の対象となっていないデータベース種別を指定して、命令を実行した場合
301EHex	DB 接続サービス未起動	DB 接続サービスを［使用しない］に設定した状態で、本命令を実行した場合

機能説明

「TableName」で指定した DB のテーブルと「MapVar」で指定した DB マップ変数を関連付けします。
DB_Insert 命令、DB_Update 命令、DB_Select 命令、DB_BatchInsert 命令を実行する前に、あらかじめ、この命令を実行する必要があります。
マッピングする変数の SQL コマンド種別を「SQLType」で指定します。例えば「MapVar」で指定した DB マップ変数の値を DB_Insert 命令で DB へ挿入する場合には、_DBC_SQLTYPE_INSERT を指定します。

使用上の注意

- 本命令は、「Execute」の値が FALSE になっても、また実行時間がタスク周期を超えても、処理を最後まで続けます。処理が正常終了したかどうかは、「Done」の値が TRUE になることで確認してください。
- 「Execute」、「Done」、「Busy」、「Error」のタイミングチャートは、『NJ/NX シリーズコマンドリファレンスマニュアル　基本編』（SBCA-468）の“本章の見方”を参照してください。
- 本命令は、イベントタスク上で使うことができません。コンパイル時にエラーになります。
- DB 接続サービスをテストモードで開始している場合、実際には DB への接続をしていませんが、本命令は正常終了します。
- 関連付けできる DB マップ変数の数は、「1-2-1 DB 接続サービスの仕様 (P.1-4)」を参照してください。ただし、関連付けする DB マップ変数の数が上限値に満たない場合でも、以下の条件に該当する場合は、命令エラー（データ容量超過）が発生します。
  a）すべての DB コネクションで、DB マップ変数のデータ型として使用する構造体定義のメンバ総数が 10000 個を超える場合
- 本命令は、以下の場合、異常となります。「Error」は TRUE となります。
  a）DB 接続サービスが動作中でないときに、本命令を実行した場合
  b）DB 接続サービスが初期処理中に、本命令を実行した場合
  c）DB 接続サービスが異常停止中に、本命令を実行した場合
  d）DB 接続サービスが終了後／終了処理中に、本命令を実行した場合
  e）「DBConnection」が不正、または既に切断している場合
  f）「TableName」が NULL 文字（16#00）だけの文字列である場合
  g）「TableName」にスペース文字を含む文字列を指定した場合
  h）「MapVar」で指定した変数のデータ型が構造体でない場合
  i）「MapVar」で指定した変数の構造体のメンバに派生データ型が含まれている場合
  j）構造体型の配列指定変数を INSERT 用、UPDATE 用の DB マップ変数「MapVar」に指定した場合
  k）「SQLType」に列挙子にない値を指定した場合
  l）実行した SQL 文が DB でエラーとなった場合
  m）ネットワーク障害などで DB と通信できない場合

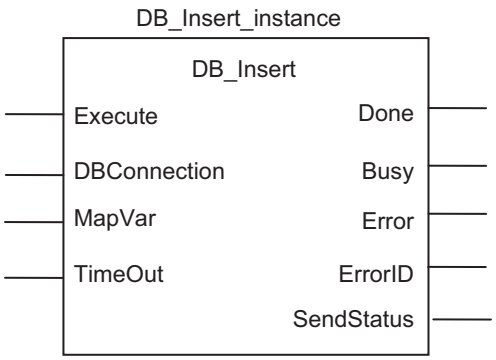
n）マッピング可能な DB マップ変数の最大数を超える DB マップ変数を関連付けようとした場合
o）DB 接続命令を同時に 32 個を超えて実行した場合
p）BATCHINSERT 用の DB マップ変数に構造体変数を指定した場合

サンプルプログラム

サンプルプログラムは、DB_Update 命令のサンプルプログラム「サンプルプログラム (P.7-23)」を参照してください。

DB_Insert（DB レコード挿入）

DB のテーブルへ DB マップ変数の値を 1 つのレコードとして挿入します。

命令	名称	FB/FUN	グラフィック表現	ST 表現
DB_Insert	DB レコード挿入	FB		DB_Insert_instance (Execute, DBConnection, MapVar, TimeOut, Done, Busy, Error, ErrorID, SendStatus);

（注）DB_Insert 命令のインスタンスとして、インスタンス名 DB_Insert_instance で変数宣言されている場合の例です。

変数

入力変数

入力変数	名称	データ型	有効範囲	単位	初期値	内容
Execute	起動	BOOL	FALSE, TRUE	-	FALSE	実行条件を指定します
DBConnection	DB コネクション	DWORD	16#00000000 ～ FFFFFFFF	-	16#00000000	DB_Connect で取得した DB コネクション
MapVar	DB マップ変数	構造体型	データ型に従う	-	-	DB_CreateMapping 命令で関連付けた DB マップ変数
TimeOut	命令実行タイムアウト時間	TIME	T#0s, T#0.05s ～T#180s	-	T#0s	命令実行タイムアウト時間を指定します T#0s を指定した場合は、タイムアウト監視なしとなります

出力変数

出力変数	名称	データ型	有効範囲	単位	内容
Done	完了	BOOL	FALSE, TRUE	-	正常終了を表します
Busy	実行中	BOOL	FALSE, TRUE	-	処理中を表します
Error	エラー	BOOL	FALSE, TRUE	-	異常終了を表します
ErrorID	エラーコード	WORD	16#0000 ～ 16#FFFF	-	エラーコードを表します

出力変数	名称	データ型	有効範囲	単位	内容
SendStatus	送信状態	_eDBC_SEND_STATUS	データ型に従う	-	SQL 文の送信の進捗状況を出力します

関連するシステム定義変数

システム定義変数	名称	データ型	内容
_EIP_EtnOnlineSta	オンライン	BOOL	内蔵 EtherNet/IP ポートの通信機能の状態 TRUE:利用可能 FALSE:利用不可

関連するデバイスパラメータ

NX シリーズに接続した X バスユニットの Ethernet ポートを使用する場合は、X バスユニットのデバイスパラメータを参照してください。

関連するエラーコード一覧

エラーコード	名称	内容
0400Hex	入力値範囲外	「TimeOut」に有効範囲外の値を指定した場合
041DHex	命令同時実行リソースオーバ	DB 接続命令を同時に 32 個を超えて実行した場合
3000Hex	DB 接続サービス未開始	DB 接続サービスが動作中でないときに、本命令を実行した場合
3002Hex	DB 接続サービス終了／終了処理中	DB 接続サービスが終了後／終了処理中に、本命令を実行した場合
3008Hex	DB コネクション不正	「DBConnection」が不正または既に切断している場合
300AHex	DB マップ変数未登録	「MapVar」で指定した変数を DB_CreateMapping でマッピングしていない場合
300BHex	SQL 実行エラー	実行した SQL 文が DB でエラーとなった場合 NJ/NX シリーズ側のデータ型と DB 側のデータ型の対応表に記載していないデータ型の組み合わせで、型変換できない場合
300CHex	スプール容量超過	スプールの容量を超えたため保留できない場合
3011Hex	DB コネクション異常切断発生中	ネットワーク障害などで DB と通信できない場合
3012Hex	DB 接続命令実行タイムアウト	指定した命令実行タイムアウト時間内に命令が終了しなかった場合
3013Hex	DB 接続サービス異常停止中	DB 接続サービスが異常停止中に、本命令を実行した場合
3014Hex	スプールデータあり	既にスプールに保留した SQL 文が存在するため SQL 文をスプールに保留した
3015Hex	DB 接続サービス初期処理中	DB 接続サービスが初期処理中に、本命令を実行した場合
3016Hex	DB 処理中	前回実行した DB_Insert／DB_Update／DB_Select／DB_Delete 命令が命令実行タイムアウトし DB 側の処理がまだ完了していない状態で命令を実行した場合
301EHex	DB 接続サービス未起動	DB 接続サービスを［使用しない］に設定した状態で、本命令を実行した場合

機能説明

DB_CreateMapping 命令で関連付けたテーブルへ、DB マップ変数「MapVar」の値を 1 レコードとして挿入します。

スプール機能を有効にしている場合、ネットワーク障害などにより DB へ挿入できないときには、スプールへ SQL 文を保留します。スプールへ保留した場合、「SendStatus」に「_DBC_SEND_SPOOLED」をセットし異常終了（DB コネクション異常切断発生中）します。
スプール機能を有効にしている場合、「TimeOut」で指定した命令実行タイムアウト時間内に DB へ挿入できないときは、スプールへ SQL 文を保留します。スプールへ保留した場合、「SendStatus」に「_DBC_SEND_SPOOLED」をセットし異常終了（DB 接続命令実行タイムアウト）します。

スプール機能を有効にしている場合、既にスプールに保留した SQL 文が存在する場合は、スプールへ SQL 文を保留します。スプールへ保留した場合、「Send Status」に「_DBC_SEND_SPOOLED」をセットし異常終了（スプールデータあり）します。

スプール機能を有効にしていても、命令エラー（SQL 実行エラー）が発生した場合、送信した SQL 文自体が SQL 実行エラーの原因である可能性があります。SQL 文を再送しても、再度 SQL 実行エラーが発生する可能性があるため、SQL 文はスプールに保留されません。

保留した SQL 文が DB コネクションあたりのスプールの容量に達している場合、本命令は異常終了（スプール容量超過）します。

使用上の注意

- 本命令は、「Execute」の値が FALSE になっても、また実行時間がタスク周期を超えても、処理を最後まで続けます。処理が正常終了したかどうかは、「Done」の値が TRUE になることで確認してください。
- 「Execute」、「Done」、「Busy」、「Error」のタイミングチャートは、『NJ/NX シリーズコマンドリファレンスマニュアル　基本編』（SBCA-468）の“本章の見方”を参照してください。
- 本命令は、イベントタスク上で使うことができません。コンパイル時にエラーになります。
- 本命令は、SQL 文の不正などに起因し DB へ登録できない場合、SQL 文をスプールに保留せず異常終了します。
- DB 接続サービスをテストモードで開始している場合、本命令は実際には DB へ INSERT を実行せず正常終了します。
- エラーコードが 300BHex（SQL 実行エラー）の場合は、DB_GetConnectionStatus 命令で、SQL 実行エラーの詳細情報を取得することができます。
- 命令実行タイムアウト時間の測定誤差の目安は、タスク実行時間の割合が 50%で 100 カラムのレコードの場合、+50ms です。ただし、タスク実行時間の割合およびカラム数によって、測定誤差が変動します。
- 同一の DB コネクションに対して DB 接続命令を同時に複数実行した場合、DB 接続サービスは命令を 1 個ずつ順番に実行します。2 個目以降に実行される命令の命令実行タイムアウト時間は、「Execute」の値が TRUE になった時点ではなく、その命令を DB 接続サービスが実行した時点から計測を始めます。そのため、「Execute」の値が TRUE になった時点から、その命令がタイムアウトするまでの時間は、命令実行タイムアウト時間で設定した値よりも長くなります。
- DB 接続命令が実際に実行される前に、DB マップ変数の値が変更された場合、変更後の値で DB 接続命令が実行されることがあります。DB マップ変数の値を変更する場合は、DB 接続命令が終了したことを確認した上で、実施するようにユーザプログラムを作成してください。
- 本命令は、以下の場合、異常となります。「Error」は TRUE になります。
  a）DB 接続サービスが動作中でないときに、本命令を実行した場合
  b）DB 接続サービスが初期処理中に、本命令を実行した場合

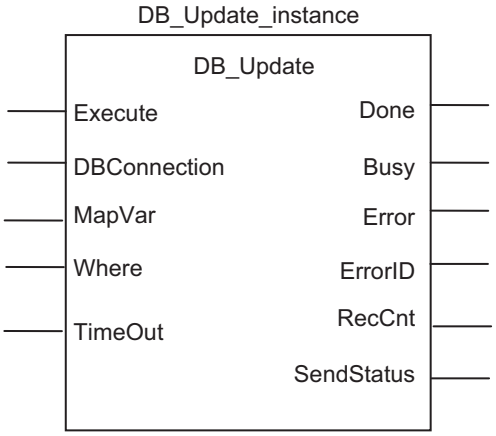
c）DB 接続サービスが異常停止中に、本命令を実行した場合
d）DB 接続サービスが終了後／終了処理中に、本命令を実行した場合
e）「DBConnection」が不正、または既に切断している場合
f）「MapVar」で指定した変数を DB_CreateMapping でマッピングしていない場合
g）「TimeOut」に有効範囲外の値を指定した場合
h）実行した SQL 文が DB でエラーとなった場合
i）NJ/NX シリーズ側のデータ型と DB 側のデータ型の対応表に記載していないデータ型の組み合わせで、型変換できない場合
j）ネットワーク障害などで DB と通信できない場合
k）既にスプールに保留した SQL 文が存在する場合
l）スプールの容量を超えたため保留できない場合
m）指定した命令実行タイムアウト時間内に命令が終了しなかった場合
n）前回実行した DB_Insert/DB_Update/DB_Select/DB_Delete 命令が命令実行タイムアウトし、サーバ側 DB の処理がまだ完了していない状態で命令を実行した場合
o）DB 接続命令を同時に 32 個を超えて実行した場合

サンプルプログラム

サンプルプログラムは、DB_Update 命令のサンプルプログラム「サンプルプログラム (P.7-23)」を参照してください。

DB_Update（DB レコード更新）

テーブルのレコードの値を、DB マップ変数の値で更新します。

命令	名称	FB/FUN	グラフィック表現	ST 表現
DB_Update	DB レコード更新	FB		DB_Update_instance (Execute, DBConnection, MapVar, Where, TimeOut, Done, Busy, Error, ErrorID, RecCnt, SendStatus);

（注）DB_Update 命令のインスタンスとして、インスタンス名 DB_Update_instance で変数宣言されている場合の例です。

変数

入力変数

入力変数	名称	データ型	有効範囲	単位	初期値	内容
Execute	起動	BOOL	FALSE, TRUE	-	FALSE	実行条件を指定します
DBConnection	DB コネクション	DWORD	16#00000000 ～ FFFFFFFF	-	16#00000000	DB_Connect で取得した DB コネクション
MapVar	DB マップ変数	構造体型	データ型に従う	-	-	DB_CreateMapping 命令で関連付けた DB マップ変数
Where	抽出条件	STRING	最大 1986 バイト（終端 NULL 含む）*1	-	''	抽出条件(WHERE 句)を表す文字列を指定します（'WHERE' は含みません）
TimeOut	命令実行タイムアウト時間	TIME	T#0s, T#0.05s ～T#180s	-	T#0s	命令実行タイムアウト時間を指定します T#0s を指定した場合は、タイムアウト監視なしとなります

*1. DB 側が大文字／小文字を区別する場合、テーブル名は以下のように指定してください。
MySQL と接続するとき、半角バッククォートで囲む
例）`ColumnA`
MySQL 以外と接続するとき、半角ダブルクォートで囲む
例）"ColumnA"

出力変数

出力変数	名称	データ型	有効範囲	単位	内容
Done	完了	BOOL	FALSE, TRUE	-	正常終了を表します
Busy	実行中	BOOL	FALSE, TRUE	-	処理中を表します
Error	エラー	BOOL	FALSE, TRUE	-	異常終了を表します
ErrorID	エラーコード	WORD	16#0000 ～ 16#FFFF	-	エラーコードを表します
RecCnt	レコード数	DINT	0 ～ 2147483647	-	更新したレコード数
SendStatus	送信状態	_eDBC_SEND_STATUS	データ型に従う	-	SQL 文の送信の進捗状況を出力します

関連するシステム定義変数

システム定義変数	名称	データ型	内容
_EIP_EtnOnlineSta	オンライン	BOOL	内蔵 EtherNet/IP ポートの通信機能の状態 TRUE:利用可能 FALSE:利用不可

関連するデバイスパラメータ

NX シリーズに接続した X バスユニットの Ethernet ポートを使用する場合は、X バスユニットのデバイスパラメータを参照してください。

関連するエラーコード一覧

エラーコード	名称	内容
0400Hex	入力値範囲外	「TimeOut」に有効範囲外の値を指定した場合
041DHex	命令同時実行リソースオーバ	DB 接続命令を同時に 32 個を超えて実行した場合
3000Hex	DB 接続サービス未開始	DB 接続サービスが動作中でないときに、本命令を実行した場合
3002Hex	DB 接続サービス終了／終了処理中	DB 接続サービスが終了後／終了処理中に、本命令を実行した場合
3008Hex	DB コネクション不正	「DBConnection」が不正、または既に切断している場合
300AHex	DB マップ変数未登録	「MapVar」で指定した変数を DB_CreateMapping でマッピングしていない場合
300BHex	SQL 実行エラー	実行した SQL 文が DB でエラーとなった場合 NJ/NX シリーズ側のデータ型と DB 側のデータ型の対応表に記載していないデータ型の組み合わせで、型変換できない場合
300CHex	スプール容量超過	スプールの容量を超えたため保留できない場合
300EHex	抽出条件不正	「Where」が NULL 文字（16#00）だけの文字列である場合
3011Hex	DB コネクション異常切断発生中	ネットワーク障害などで DB と通信できない場合
3012Hex	DB 接続命令実行タイムアウト	指定した命令実行タイムアウト時間内に命令が終了しなかった場合
3013Hex	DB 接続サービス異常停止中	DB 接続サービスが異常停止中に、本命令を実行した場合
3014Hex	スプールデータあり	既にスプールに保留した SQL 文が存在するため SQL 文をスプールに保留した
3015Hex	DB 接続サービス初期処理中	DB 接続サービスが初期処理中に、本命令を実行した場合

エラーコード	名称	内容
3016Hex	DB 処理中	前回実行した DB_Insert／DB_Update／DB_Select／DB_Delete 命令が命令実行タイムアウトし DB 側の処理がまだ完了していない状態で命令を実行した場合
301EHex	DB 接続サービス未起動	DB 接続サービスを［使用しない］に設定した状態で、本命令を実行した場合

機能説明

DB_CreateMapping 命令で関連付けたテーブルから Where 句で抽出したレコードの値を、DB マップ変数「MapVar」の値で更新します。

更新するレコードは抽出条件「Where」(WHERE 句)で抽出します。「Where」は文字列で表現します。
抽出条件「Where」は、NULL 文字（16#00）だけを指定することはできません。命令が異常終了します。

WHERE 句でシングルクォートを使用する場合は、エスケープ文字（$' ）で指定します。
エスケープ文字については、『NJ/NX シリーズ CPU ユニット　ユーザーズマニュアル　ソフトウェア編（SBCA-467)』を参照してください。
WHERE 句の記述形式については、接続先の DB のマニュアルを参照してください。
抽出条件「Where」は以下のような値を指定します。

例 1）　特定のカラムの値が指定した値以上のレコードの値を更新する
「ColumnA」（符号なし整数）の値が、1234 以上のレコードの値を更新する
'" ColumnA" >= 1234'
生成する SQL 文：　UPDATE TableProduct SET “ColumnA” =<値>, “ColumnB” =<値> Where “ColumnA” >= 1234
例 2）　特定のカラムの値が、指定した文字列で始まるレコードの値を更新する
「ColumnB」（文字列）の値が、' ABC' で始まるレコードの値を更新する
'" ColumnB" LIKE $'ABC%$"'
生成する SQL 文：　UPDATE TableProduct SET “ColumnA” =<値>, “ColumnB” =<値> Where “ColumnB” LIKE 'ABC%'
例 3）　特定のカラムの値が、指定した変数の値以上のレコードの値を更新する
「ColumnA」（符号なし整数）の値が、指定した変数の値以上の値を更新する
指定値： UINTVar := 1234;
抽出条件「Where」の入力パラメータ： WhereCond_Update := CONCAT('$" ColumnA$" >= ', UINT_TO_STRING(UINTVar));
生成する SQL 文：　UPDATE TableProduct SET “ColumnA” =<値>, “ColumnB” =<値> Where “ColumnA” >= 1234

スプール機能を有効にしている場合、ネットワーク障害などにより DB のレコードを更新できないときには、スプールへ SQL 文を保留します。スプールへ保留した場合、「SendStatus」に「_DBC_SEND_SPOOLED」をセットし異常終了します。

スプール機能を有効にしている場合、「TimeOut」で指定した命令実行タイムアウト時間内に DB のレコードを更新できないときは、スプールへ SQL 文を保留します。スプールへ保留した場合、「SendStatus」に「_DBC_SEND_SPOOLED」をセットし異常終了（DB 接続命令実行タイムアウト）します。

スプール機能を有効にしていても、命令エラー（SQL 実行エラー）が発生する場合、抽出条件の設定誤りなど、送信した SQL 文自体が SQL 実行エラーの原因である可能性があります。SQL 文を再送しても、再度 SQL 実行エラーが発生する可能性があるため、SQL 文はスプールに保留されません。

保留した SQL 文がスプールの最大容量に達している場合、本命令は異常終了（スプール容量超過）します。

使用上の注意

- 本命令は、「Execute」の値が FALSE になっても、また実行時間がタスク周期を超えても、処理を最後まで続けます。処理が正常終了したかどうかは、「Done」の値が TRUE になることで確認してください。
- 「Execute」、「Done」、「Busy」、「Error」のタイミングチャートは、『NJ/NX シリーズコマンドリファレンスマニュアル　基本編』（SBCA-468）の“本章の見方”を参照してください。
- 本命令は、イベントタスク上で使うことができません。コンパイル時にエラーになります。
- 本命令は、抽出条件を指定せずに実行することはできません。
- 本命令は、SQL 文の不正などに起因し DB へ登録できない場合、SQL 文をスプールに保留せず異常終了します。
- DB 接続サービスをテストモードで開始している場合、本命令は実際には DB へ UPDATE を実行せず正常終了します。
- エラーコードが 300BHex（SQL 実行エラー）の場合は、DB_GetConnectionStatus 命令で、SQL 実行エラーの詳細情報を取得することができます。
- 命令実行タイムアウト時間の測定誤差の目安は、タスク実行時間の割合が 50%で 100 カラムのレコードの場合、+50ms です。ただし、タスク実行時間の割合およびカラム数によって、測定誤差が変動します。
- 同一の DB コネクションに対して DB 接続命令を同時に複数実行した場合、DB 接続サービスは命令を 1 個ずつ順番に実行します。2 個目以降に実行される命令の命令実行タイムアウト時間は、「Execute」の値が TRUE になった時点ではなく、その命令を DB 接続サービスが実行した時点から計測を始めます。そのため、「Execute」の値が TRUE になった時点から、その命令がタイムアウトするまでの時間は、命令実行タイムアウト時間で設定した値よりも長くなります。
- DB 接続命令が実際に実行される前に、DB マップ変数の値が変更された場合、変更後の値で DB 接続命令が実行されることがあります。DB マップ変数の値を変更する場合は、DB 接続命令が終了したことを確認した上で、実施するようにユーザプログラムを作成してください。
- 本命令は、以下の場合、異常となります。「Error」は TRUE となります。
  a）DB 接続サービスが動作中でないときに、本命令を実行した場合
  b）DB 接続サービスが初期処理中に、本命令を実行した場合
  c）DB 接続サービスが異常停止中に、本命令を実行した場合
  d）DB 接続サービスが終了後／終了処理中に、本命令を実行した場合
  e）「DBConnection」が不正または既に切断している場合
  f）「MapVar」で指定した変数を DB_CreateMapping でマッピングしていない場合
  g）「Where」が NULL 文字（16#00）だけの文字列である場合
  h）「TimeOut」に有効範囲外の値を指定した場合
  i）実行した SQL 文が DB でエラーとなった場合
  j）NJ/NX シリーズ側のデータ型と DB 側のデータ型の対応表に記載していないデータ型の組み合わせで、型変換できない場合
  k）ネットワーク障害などで DB と通信できない場合
  l）スプールの容量を超えたため保留できない場合
  m）既にスプールに保留した SQL 文が存在する場合
  n）指定した命令実行タイムアウト時間内に命令が終了しなかった場合
  o）前回実行した DB_Insert/DB_Update/DB_Select/DB_Delete 命令が命令実行タイムアウトし、サーバ側 DB の処理がまだ完了していない状態で命令を実行した場合
  p）DB 接続命令を同時に 32 個を超えて実行した場合

サンプルプログラム

ここでは、以下のサンプルプログラムを掲載します。
- DB レコード挿入用トリガが TRUE になった時に DB に生産データを挿入
- DB レコード更新用トリガが TRUE になった時に DB の生産データを更新

DB コネクションと構造体型の設定

このサンプルプログラムで使用する必要最低限の設定を以下に示します。

● DB コネクションの設定

DB コネクション名：MyDatabase1

● 構造体型の設定

名称		データ型
PRODUCTION_INSERT		STRUCT
	Name	STRING[256]
	LotNo	STRING[32]
	Status	STRING[8]
	ProductionDate	DATE

名称		データ型
PRODUCTION_UPDATE		STRUCT
	Status	STRING[8]
	FinishTime	DATE_AND_TIME

ラダー図

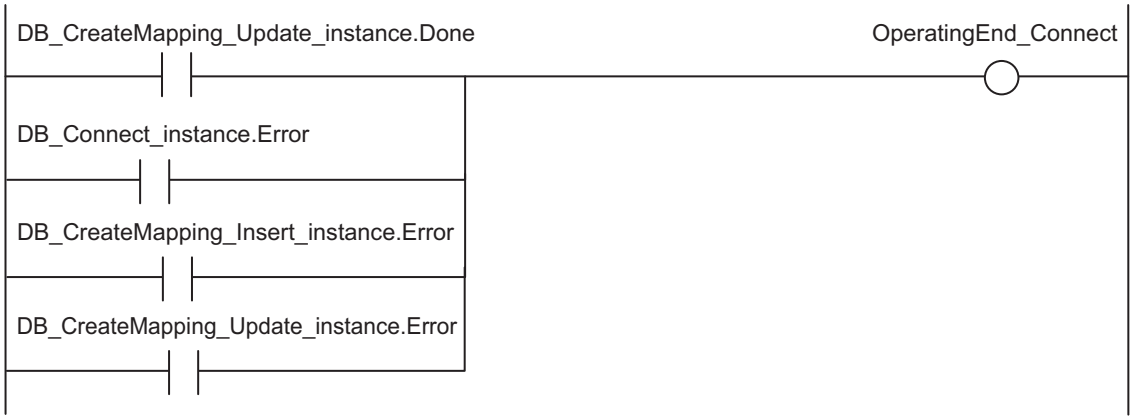
● 主な変数

名称	データ型	初期値	コメント
_DBC_Status	_sDBC_STATUS	-	DB 接続サービスのステータスを表すシステム定義変数です。
DB_Connect_instance	DB_Connect	-	DB_Connect 命令のインスタンスです。
MyDB1	DWORD	-	DB_Connect 命令のインスタンス DB_Connect_instance の出力 DBConnection に割り付けた変数です。
Trigger_Connect	BOOL	FALSE	DB コネクション接続のトリガとなる変数です。
RS_Connect_instance	RS	-	RS 命令のインスタンスです。
Operating_Connect	BOOL	FALSE	この変数が TRUE の時に、DB_Connect 命令を実行します。
OperatingEnd_Connect	BOOL	FALSE	DB_Connect 命令が終了すると、TRUE になります。
DB_CreateMapping _Insert_instance	DB_CreateMapping	-	DB_CreateMapping 命令のインスタンスです。
MapVar_Insert	PRODUCTION _INSERT		DB_CreateMapping 命令のインスタンス DB_CreateMapping_Insert_instance の入力 MapVar に割り付けた変数です。
DB_Insert_instance	DB_Insert	-	DB_Insert 命令のインスタンスです。

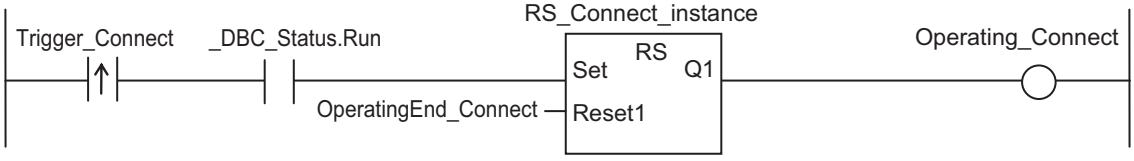
名称	データ型	初期値	コメント
Name	STRING[256]	'WORK001'	生産情報：製品名
LotNo	UINT	1234	生産情報：ロット番号
Trigger_Insert	BOOL	FALSE	DB レコード挿入のトリガとなる変数です。
RS_Insert_instance	RS	-	RS 命令のインスタンスです。
Operating_Insert	BOOL	FALSE	この変数が TRUE の時に、DB_Insert 命令を実行します。
OperatingEnd_Insert	BOOL	FALSE	DB_Insert 命令が終了すると、TRUE になります。
DB_CreateMapping_Update_instance	DB_CreateMapping	-	DB_CreateMapping 命令のインスタンスです。
MapVar_Update	PRODUCTION_UPDATE		DB_CreateMapping 命令のインスタンス DB_CreateMapping_Update_instance の入力 MapVar に割り付けた変数です。
WhereCond	STRING[256]	-	DB_Update 命令のインスタンス DB_Update_instance の入力 Where に割り付けた変数です。
DB_Update_instance	DB_Update	-	DB_Update 命令のインスタンスです。
Trigger_Update	BOOL	FALSE	DB レコード更新のトリガとなる変数です。
RS_Update_instance	RS	-	RS 命令のインスタンスです。
Operating_Update	BOOL	FALSE	この変数が TRUE の時に、DB_Update 命令を実行します。
OperatingEnd_Update	BOOL	FALSE	DB_Update 命令が終了すると、TRUE になります。
DB_Close_instance	DB_Close	-	DB_Close 命令のインスタンスです。
Trigger_Close	BOOL	FALSE	DB コネクション切断のトリガとなる変数です。
RS_Close_instance	RS	-	RS 命令のインスタンスです。
Operating_Close	BOOL	FALSE	この変数が TRUE の時に、DB_Close 命令を実行します。
OperatingEnd_Close	BOOL	FALSE	DB_Close 命令が終了すると、TRUE になります。

● サンプルプログラム

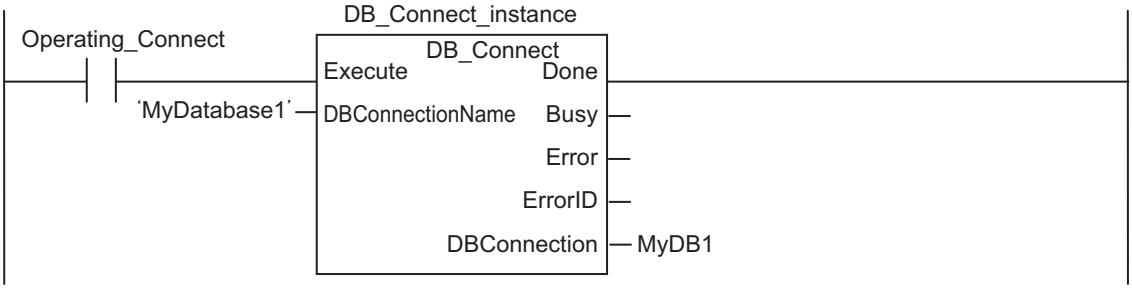
◆DB コネクション名「MyDatabase1」に接続し、DB のテーブルと変数を関連付けます
DB_Connect 命令、DB_CreateMapping 命令実行完了を判断する



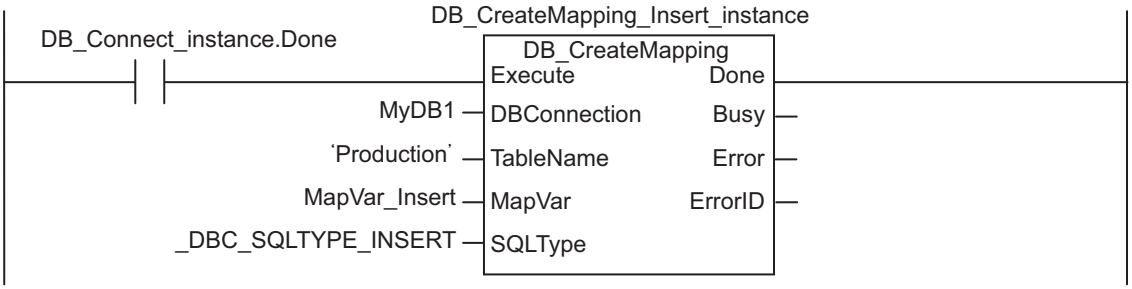
DB コネクション接続用トリガを受け付ける



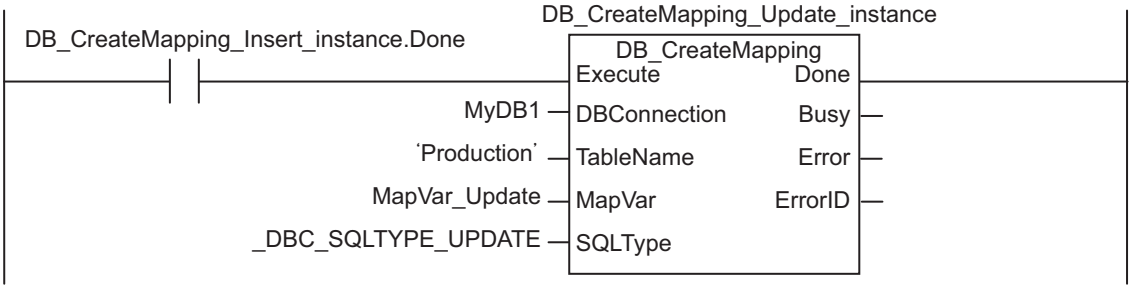
DB コネクション名「MyDatabase1」に接続する



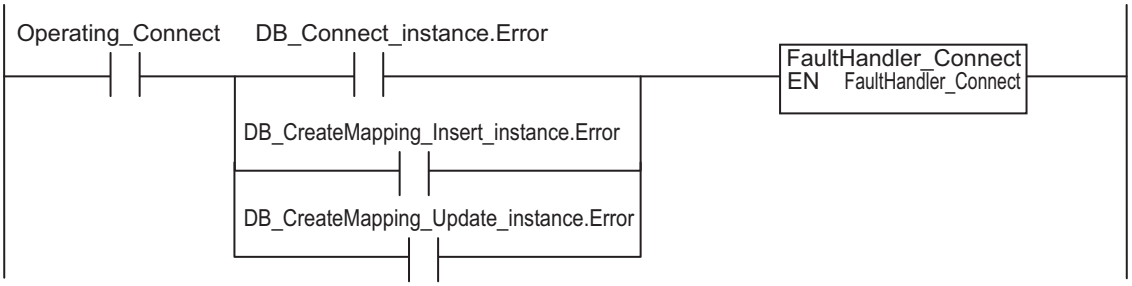
DB コネクション「MyDB1」の「Production」テーブルに、変数 MapVar_Insert を INSERT 操作用に関連付ける



DB コネクション「MyDB1」の「Production」テーブルに、変数 MapVar_Update を UPDATE 操作用に関連付ける

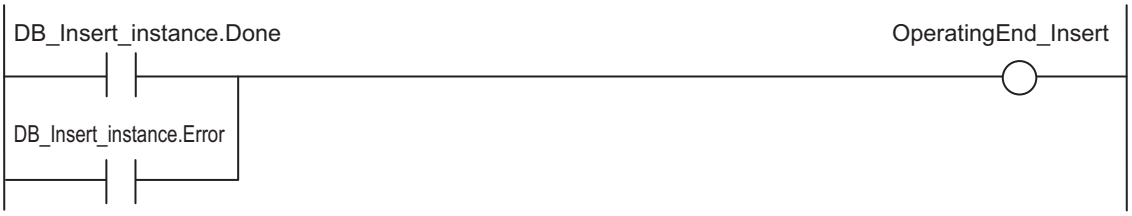


異常終了した場合、装置における異常終了時の手順（FaultHandler_Connect）を実行する。
異常終了時の手順については別途、設計してください

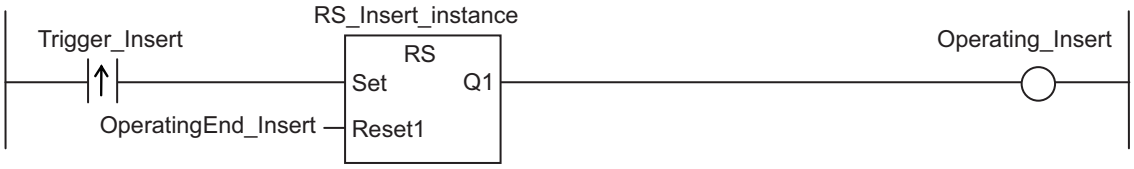


◆変数 Trigger_Insert が TRUE になった場合に、DB コネクション「MyDB1」に生産データを挿入します

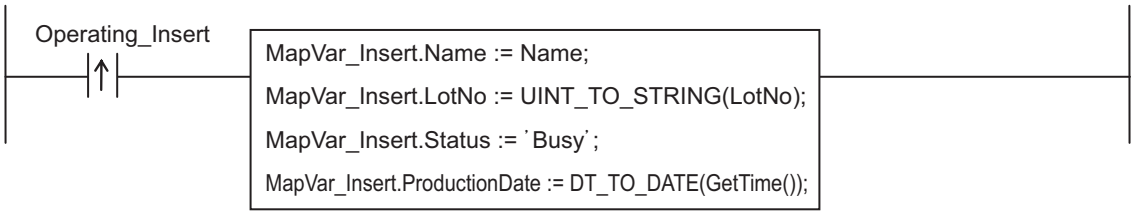
DB_Insert 命令実行完了を判断する



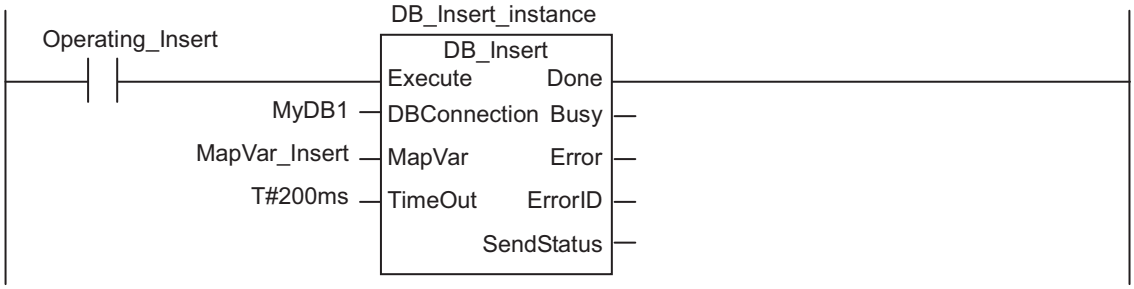
DB レコード挿入用トリガを受け付ける



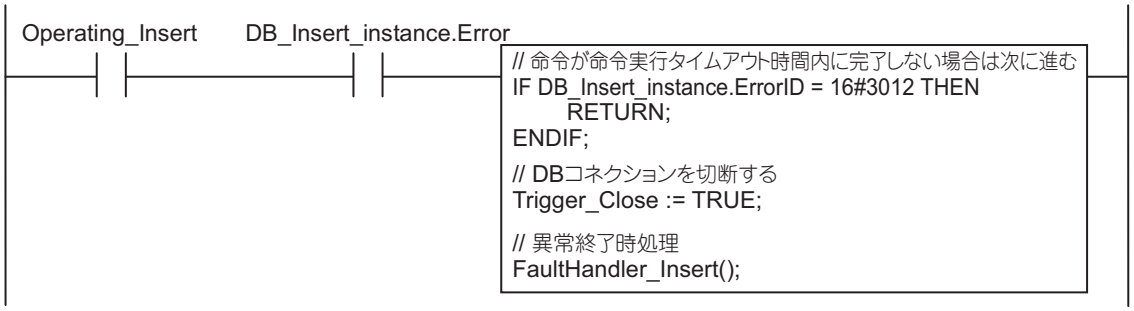
挿入する生産データを生成する



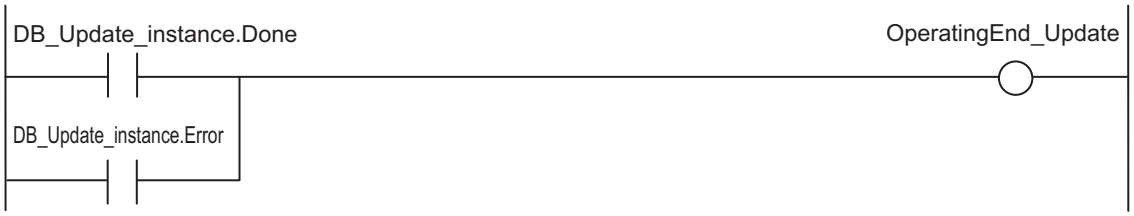
DB コネクション「MyDB1」に生産データを挿入する。
命令実行のタイムアウト時間は 200ms とする。



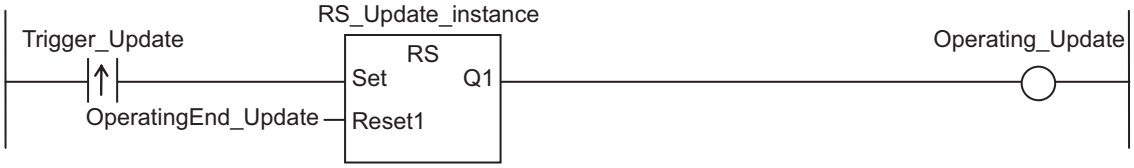
異常終了した場合、装置における異常終了時の手順（FaultHandler_Insert）を実行する。
異常終了時の手順については別途、設計してください



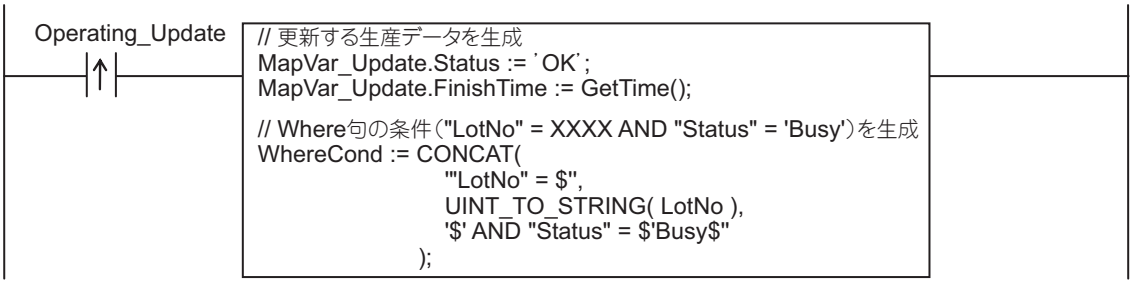
◆変数 Trigger_Update が TRUE になった場合に、DB コネクション「MyDB1」のレコードを更新します
DB_Update 命令実行完了を判断する



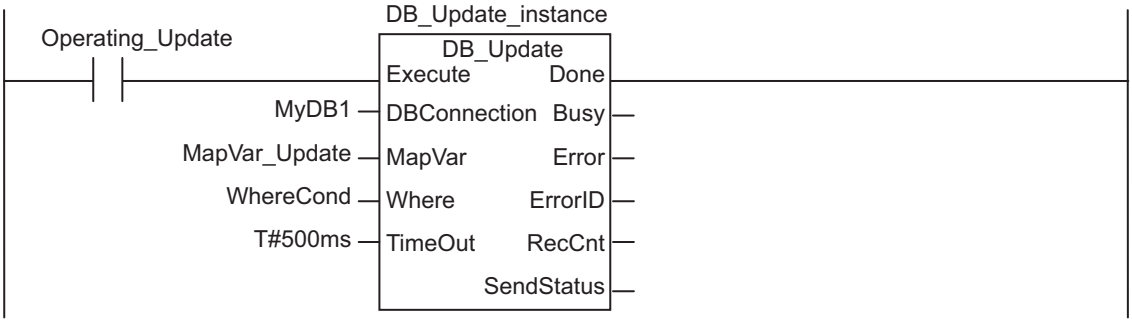
DB レコード更新用トリガを受け付ける



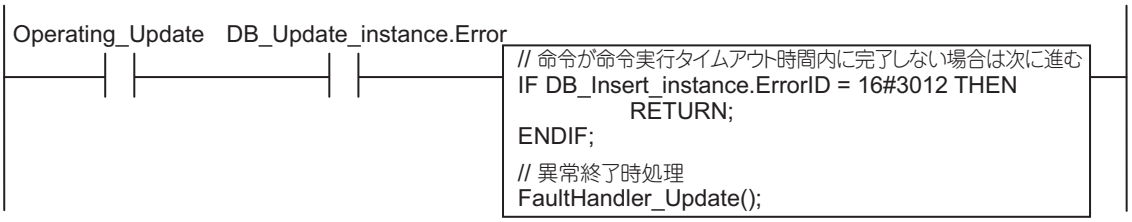
更新する生産データを生成する。
Where 句の条件を生成する



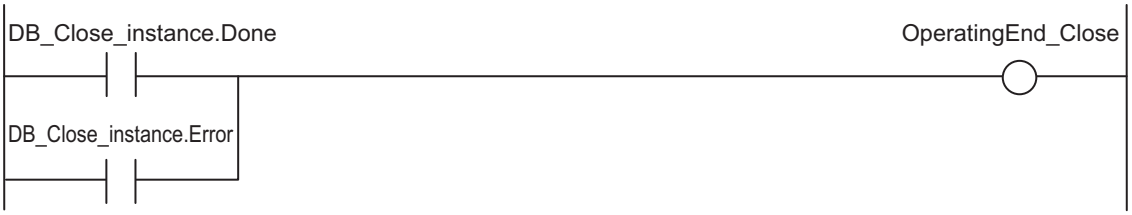
DB コネクション「MyDB1」の生産データを更新する。命令実行のタイムアウト時間は 500ms とする。



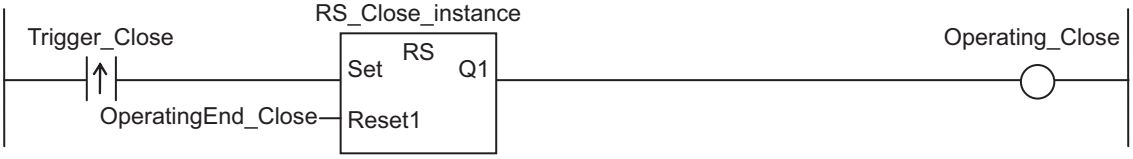
異常終了した場合、装置における異常終了時の手順（FaultHandler_Update）を実行する。
異常終了時の手順については別途、設計してください



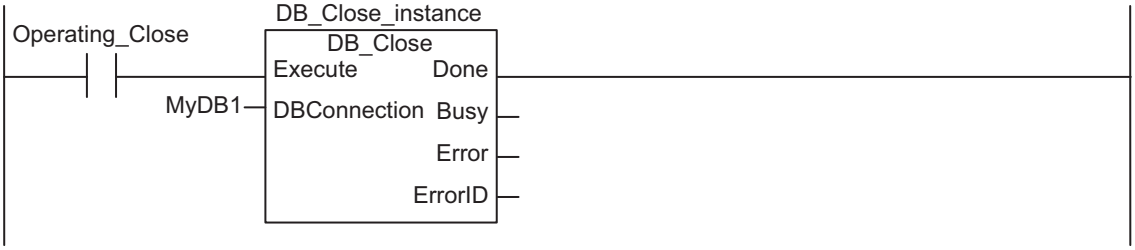
DB コネクション「MyDB1」との DB コネクションを切断する。
DB_Close 命令実行完了を判断する



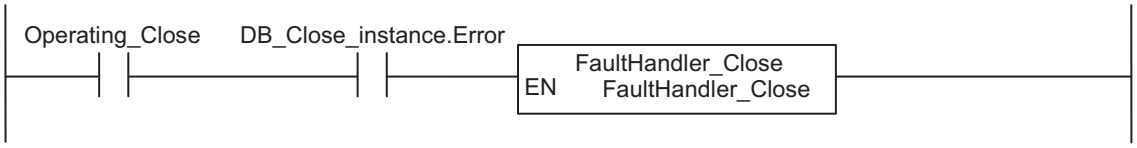
DB コネクション切断用トリガを受け付ける



DB コネクション「MyDB1」との DB コネクションを切断する



異常終了した場合、装置における異常終了時の手順（FaultHandler_Close）を実行する。
異常終了時の手順については別途、設計してください


Operating_Close
DB_Close_instance.Error
FaultHandler_Close
EN
FaultHandler_Close

ストラクチャードテキスト（ST）

● 主な変数

名称	データ型	初期値	コメント
_DBC_Status	_sDBC_STATUS	-	DB 接続サービスのステータスを表すシステム定義変数です。
DB_Connect_instance	DB_Connect	-	DB_Connect 命令のインスタンスです。
MyDB1	DWORD	-	DB_Connect のインスタンス DB_Connect_instance の出力 DBConnection に割り付けた変数です。
Trigger_Connect	BOOL	FALSE	DB コネクション接続のトリガとなる変数です。
LastTrigger_Connect	BOOL	FALSE	前回実行時のトリガ状態を保持する変数です。
Operating_Connect	BOOL	FALSE	この変数が TRUE の時に、DB_Connect 命令を実行します。
OperatingStart_Connect	BOOL	FALSE	この変数が TRUE の時に、DB コネクション接続開始時の処理を実行します。
DB_CreateMapping _Insert_instance	DB_CreateMapping	-	DB_CreateMapping 命令のインスタンスです。
MapVar_Insert	PRODUCTION_INSERT		DB_CreateMapping のインスタンス DB_CreateMapping_Insert_instance の入力 MapVar に割り付けた変数です。
DB_Insert_instance	DB_Insert	-	DB_Insert 命令のインスタンスです。
Name	STRING[256]	'WORK001'	生産情報：製品名
LotNo	UINT	1234	生産情報：ロット番号
Trigger_Insert	BOOL	FALSE	DB レコード挿入のトリガとなる変数です。
LastTrigger_Insert	BOOL	FALSE	前回実行時のトリガ状態を保持する変数です。
Operating_Insert	BOOL	FALSE	この変数が TRUE の時に、DB_Insert 命令を実行します。
OperatingStart_Insert	BOOL	FALSE	この変数が TRUE の時に、DB レコード挿入開始時の処理を実行します。
DB_CreateMapping _Update_instance	DB_CreateMapping	-	DB_CreateMapping 命令のインスタンスです。
MapVar_Update	PRODUCTION_UPDATE	-	DB_CreateMapping のインスタンス DB_CreateMapping_Update_instance の入力 MapVar に割り付けた変数です。
DB_Update_instance	DB_Update	-	DB_Update 命令のインスタンスです。
WhereCond	STRING[256]	-	DB_Update のインスタンス DB_Update_instance の入力 Where に割り付けた変数です。
Trigger_Update	BOOL	FALSE	DB レコード更新のトリガとなる変数です。
LastTrigger_Update	BOOL	FALSE	前回実行時のトリガ状態を保持する変数です。
Operating_Update	BOOL	FALSE	この変数が TRUE の時に、DB_Update 命令を実行します。
OperatingStart_Update	BOOL	FALSE	この変数が TRUE の時に、DB レコード更新開始時の処理を実行します。
DB_Close_instance	DB_Close	-	DB_Close 命令のインスタンスです。
Trigger_Close	BOOL	FALSE	DB コネクション切断のトリガとなる変数です。
LastTrigger_Close	BOOL	FALSE	前回実行時のトリガ状態を保持する変数です。
Operating_Close	BOOL	FALSE	この変数が TRUE の時に、DB_Close 命令を実行します。
OperatingStart_Close	BOOL	FALSE	この変数が TRUE の時に、DB コネクション切断開始時の処理を実行します。

名称	データ型	初期値	コメント
Stage	INT	-	DB コネクション接続の状態を表す変数です。

● サンプルプログラム

```
// ◆DB コネクション名「MyDatabese1」に接続し、DB のテーブルと変数を関連付けます

// 変数 Trigger_Connect の立ち上がりでシーケンス開始
   IF ( (Trigger_Connect=TRUE)
      AND (LastTrigger_Connect=FALSE)
      AND (_DBC_Status.Run=TRUE) ) THEN
         OperatingStart_Connect := TRUE;
         Operating_Connect := TRUE;
   END_IF;
   LastTrigger_Connect:=Trigger_Connect;

   // シーケンス開始時処理
   IF (OperatingStart_Connect=TRUE) THEN
      // DB 接続命令インスタンス初期化
      DB_Connect_instance( Execute:=FALSE );

      DB_CreateMapping_Insert_instance(
         Execute  := FALSE,
         MapVar   := MapVar_Insert,
         SQLType  := _DBC_SQLTYPE_INSERT
      );

      DB_CreateMapping_Update_instance(
         Execute  := FALSE,
         MapVar   := MapVar_Update,
         SQLType  := _DBC_SQLTYPE_UPDATE
      );

      Stage := INT#1;
      OperatingStart_Connect := FALSE;
   END_IF;

   // DB コネクション名「MyDatabese1」に接続します。
   // DB コネクション「MyDB1」の「Production」テーブルに、変数 MapVar_Insert を INSERT 操作用に関連付けます。
   // DB コネクション「MyDB1」の「Production」テーブルに、変数 MapVar_Update を UPDATE 操作用に関連付けます。
   IF (Operating_Connect=TRUE) THEN
      CASE Stage OF
      1 : // DB コネクション接続
         DB_Connect_instance(
            Execute          := TRUE,
            DBConnectionName := 'MyDatabase1',
            DBConnection     => MyDB1
         );
```

```
            IF (DB_Connect_instance.Done=TRUE) THEN
               Stage := INT#2; // 正常終了
            END_IF;
            IF (DB_Connect_instance.Error=TRUE) THEN
               Stage := INT#99; // 異常終了
            END_IF;

         2 : // DB のテーブルと変数を関連付けます
            DB_CreateMapping_Insert_instance(
               Execute       := TRUE,
               DBConnection  := MyDB1,
               TableName     := 'Production',
               MapVar        := MapVar_Insert,
               SQLType       := _DBC_SQLTYPE_INSERT
            );

            DB_CreateMapping_Update_instance(
               Execute       := TRUE,
               DBConnection  := MyDB1,
               TableName     := 'Production',
               MapVar        := MapVar_Update,
               SQLType       := _DBC_SQLTYPE_UPDATE
            );

            IF ( (DB_CreateMapping_Insert_instance.Done=TRUE)
AND (DB_CreateMapping_Update_instance.Done=TRUE) ) THEN
                  Operating_Connect:=FALSE; // 正常終了
            END_IF;
            IF ( (DB_CreateMapping_Insert_instance.Error=TRUE)
OR (DB_CreateMapping_Update_instance.Error = TRUE) ) THEN
                  Stage := INT#99; // 異常終了
            END_IF;

         99 :
            // 異常終了時処理を実行します。
            // 装置の異常発生時の手順（FaultHandler_Connect）については、実装してください。
            FaultHandler_Connect();
            Operating_Connect := FALSE;
         END_CASE;
      END_IF;

      // ------------------------------------------------------------------------------
      // ◆変数 Trigger_Insert が TRUE になった場合に、DB コネクション「MyDB1」に生産データを挿入します。

      // 変数 Trigger_Insert の立ち上がりでシーケンス開始
      IF ( (Trigger_Insert=TRUE) AND (LastTrigger_Insert=FALSE) ) THEN
         OperatingStart_Insert := TRUE;
         Operating_Insert := TRUE;
      END_IF;
      LastTrigger_Insert := Trigger_Insert;
```

```
    // シーケンス開始時処理
    IF (OperatingStart_Insert=TRUE) THEN
        // DB 接続命令インスタンス初期化
        DB_Insert_instance( Execute:=FALSE, MapVar:=MapVar_Insert );

        // 挿入する生産データを生成します。
        MapVar_Insert.Name            := Name;
        MapVar_Insert.LotNo           := UINT_TO_STRING(LotNo);
        MapVar_Insert.Status          := 'Busy';
        MapVar_Insert.ProductionDate  := DT_TO_DATE(GetTime( ));

        OperatingStart_Insert := FALSE;
    END_IF;

    // DB コネクション「MyDB1」に生産データを挿入します。命令実行のタイムアウト時間は 200ms とします。
    IF (Operating_Insert=TRUE) THEN
        // レコード挿入
        DB_Insert_instance(
            Execute       := TRUE,
            DBConnection  := MyDB1,
            MapVar        := MapVar_Insert,
            TimeOut       := T#200ms
        );

    IF (DB_Insert_instance.Done=TRUE) THEN
        Operating_Insert:=FALSE; // 正常終了
    END_IF;
    IF (DB_Insert_instance.Error=TRUE) THEN
        // 命令が命令実行タイムアウト時間内に完了しない場合は次に進む
        IF (DB_Insert_instance.ErrorID = 16#3012) THEN
            Operating_Insert:=FALSE; // 正常終了
        ELSE
            // 異常終了時処理を実行します。
            // 装置の異常発生時の手順（FaultHandler_Insert）については、実装してください。
            FaultHandler_Insert();
            Operating_Insert := FALSE;
        END_IF;
    END_IF;
END_IF;

    // ------------------------------------------------------------------------------
    // ◆変数 Trigger_Update が TRUE になった場合に、DB コネクション「MyDB1」のレコードを更新します。

    // 変数 Trigger_Update の立ち上がりでシーケンス開始
    IF ( (Trigger_Update=TRUE) AND (LastTrigger_Update=FALSE) ) THEN
        OperatingStart_Update := TRUE;
        Operating_Update := TRUE;
    END_IF;
    LastTrigger_Update := Trigger_Update;

    // シーケンス開始時処理
```

```
    IF (OperatingStart_Update=TRUE) THEN
        // DB 接続命令インスタンス初期化
        DB_Update_instance( Execute:=FALSE, MapVar:=MapVar_Update );

        // 更新する生産データを生成します。
        MapVar_Update.Status := 'OK';
        MapVar_Update.FinishTime := GetTime();

        // Where 句の条件 ("LotNo" = XXXX AND "Status" = 'Busy') を生成
        WhereCond := CONCAT(
                              '"LotNo" = $'',
                              UINT_TO_STRING( LotNo ),
                              '$' AND "Status" = $'Busy$''
                            );

        OperatingStart_Update := FALSE;
    END_IF;

    // DB コネクション「MyDB1」のレコードを更新します。命令実行のタイムアウト時間は 200ms とします。
    IF (Operating_Update=TRUE) THEN
        // レコード更新
        DB_Update_instance(
            Execute       := TRUE,
            DBConnection  := MyDB1,
            MapVar        := MapVar_Update,
            Where         := WhereCond,
            TimeOut       := T#200ms );

    IF (DB_Update_instance.Done=TRUE) THEN
        Operating_Update:=FALSE; // 正常終了
    END_IF;
    IF (DB_Update_instance.Error=TRUE) THEN
        // 命令が命令実行タイムアウト時間内に完了しない場合は次に進む
        IF (DB_Update_instance.ErrorID = 16#3012) THEN
            Operating_Update:=FALSE; // 正常終了
        ELSE
            // 異常終了時処理を実行します。
            // 装置の異常発生時の手順 (FaultHandler_Update) については、実装してください。
            FaultHandler_Update();
            Operating_Update := FALSE;
        END_IF;
    END_IF;
END_IF;

// ---------------------------------------------------------------------------------------
// ◆DB コネクション「MyDB1」との DB コネクションを切断します。

// 変数 Trigger_Close の立ち上がりでシーケンス開始
IF ( (Trigger_Close=TRUE) AND (LastTrigger_Close=FALSE) ) THEN
    OperatingStart_Close := TRUE;
    Operating_Close := TRUE;
```

```
END_IF;
LastTrigger_Close := Trigger_Close;

// シーケンス開始時処理
IF (OperatingStart_Close=TRUE) THEN
   // DB 接続命令インスタンス初期化
   DB_Close_instance( Execute:=FALSE );

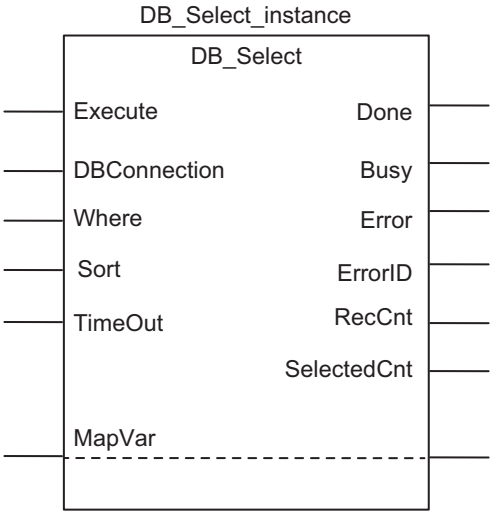
   OperatingStart_Close := FALSE;
END_IF;

// DB コネクション「MyDB1」との DB コネクションを切断します。
IF (Operating_Close=TRUE) THEN
   // DB コネクション切断
   DB_Close_instance( Execute:=TRUE, DBConnection:=MyDB1 );

   IF (DB_Close_instance.Done=TRUE) THEN
      Operating_Close := FALSE; // 正常終了
   END_IF;
   IF (DB_Close_instance.Error=TRUE) THEN
      // 異常終了時処理を実行します。
      // 装置の異常発生時の手順（FaultHandler_Close）については、実装してください。
      FaultHandler_Close();
      Operating_Close := FALSE;
   END_IF;
END_IF;
```

DB_Select（DB レコード取得）

DB の 1 つのテーブルからレコードを抽出し、DB マップ変数に取得します。

命令	名称	FB/FUN	グラフィック表現	ST 表現
DB_Select	DB レコード取得	FB		DB_Select_instance (Execute, DBConnection, Where, Sort, TimeOut, MapVar, Done, Busy, Error, ErrorID, RecCnt, SelectedCnt);

（注）DB_Select 命令のインスタンスとして、インスタンス名 DB_Select_instance で変数宣言されている場合の例です。

変数

入力変数

入力変数	名称	データ型	有効範囲	単位	初期値	内容
Execute	起動	BOOL	FALSE, TRUE	-	FALSE	実行条件を指定します
DBConnection	DB コネクション	DWORD	16#00000000 ～ FFFFFFFF	-	16#00000000	DB_Connect で取得した DB コネクション
Where	抽出条件	STRING	最大 1986 バイト（終端 NULL 含む）*1	-	"	抽出条件(WHERE 句)を表す文字列を指定します（'WHERE' は含みません） DB 接続サービスのバージョン Ver.2.00 以降、本変数への入力を省略しても、命令が異常終了となりません
Sort	ソート条件	STRING	最大 1986 バイト（終端 NULL 含む）*1	-	"	ソート条件(ORDER BY 句)を表す文字列を指定します（'ORDER BY' は含みません）

入力変数	名称	データ型	有効範囲	単位	初期値	内容
TimeOut	命令実行タイムアウト時間	TIME	T#0s, T#0.05s ～T#180s	-	T#0s	命令実行タイムアウト時間を指定します T#0s を指定した場合は、タイムアウト監視なしとなります

*1. DB 側が大文字／小文字を区別する場合、テーブル名は以下のように指定してください。
MySQL と接続するとき、半角バッククォートで囲む
例）`ColumnA`
MySQL 以外と接続するとき、半角ダブルクォートで囲む
例）"ColumnA"

入出力変数

入出力変数	名称	データ型	有効範囲	単位	内容
MapVar	DB マップ変数	構造体型、 構造体型の配列（全体配列）	データ型に従う	-	DB_CreateMapping 命令で関連付けた DB マップ変数

出力変数

出力変数	名称	データ型	有効範囲	単位	内容
Done	完了	BOOL	FALSE, TRUE	-	正常終了を表します
Busy	実行中	BOOL	FALSE, TRUE	-	処理中を表します
Error	エラー	BOOL	FALSE, TRUE	-	異常終了を表します
ErrorID	エラーコード	WORD	16#0000 ～ 16#FFFF	-	エラーコードを表します
RecCnt	レコード数	DINT	0 ～ 65535	-	DB マップ変数に格納したレコード数
SelectedCnt	抽出レコード数	DINT	0 ～ 2147483647	-	抽出条件で抽出したレコードの総数

関連するシステム定義変数

システム定義変数	名称	データ型	内容
_EIP_EtnOnlineSta	オンライン	BOOL	内蔵 EtherNet/IP ポートの通信機能の状態 TRUE:利用可能 FALSE:利用不可

関連するデバイスパラメータ

NX シリーズに接続した X バスユニットの Ethernet ポートを使用する場合は、X バスユニットのデバイスパラメータを参照してください。

関連するエラーコード一覧

エラーコード	名称	内容
0400Hex	入力値範囲外	「TimeOut」に有効範囲外の値を指定した場合

エラーコード	名称	内容
041DHex	命令同時実行リソースオーバ	DB 接続命令を同時に 32 個を超えて実行した場合
3000Hex	DB 接続サービス未開始	DB 接続サービスが動作中でないときに、本命令を実行した場合
3002Hex	DB 接続サービス終了／終了処理中	DB 接続サービスが終了後／終了処理中に、本命令を実行した場合
3008Hex	DB コネクション不正	「DBConnection」が不正、または既に切断している場合
300AHex	DB マップ変数未登録	「MapVar」で指定した変数を DB_CreateMapping でマッピングしていない場合
300BHex	SQL 実行エラー	実行した SQL 文が DB でエラーとなった場合 値が NULL のカラムを含むレコードを取得した場合 NJ/NX シリーズ側のデータ型と DB 側のデータ型の対応表に記載していないデータ型の組み合わせで、型変換できない場合
300EHex	抽出条件不正	「Where」が NULL 文字（16#00）だけの文字列である場合
3011Hex	DB コネクション異常切断発生中	ネットワーク障害などで DB と通信できない場合
3012Hex	DB 接続命令実行タイムアウト	指定した命令実行タイムアウト時間内に命令が終了しなかった場合
3013Hex	DB 接続サービス異常停止中	DB 接続サービスが異常停止中に、本命令を実行した場合
3014Hex	スプールデータあり	既にスプールに保留した SQL 文が存在するため実行できない場合
3015Hex	DB 接続サービス初期処理中	DB 接続サービスが初期処理中に、本命令を実行した場合
3016Hex	DB 処理中	前回実行した DB_Insert／DB_Update／DB_Select／DB_Delete 命令が命令実行タイムアウトし DB 側の処理がまだ完了していない状態で命令を実行した場合
301EHex	DB 接続サービス未起動	DB 接続サービスを［使用しない］に設定した状態で、本命令を実行した場合

機能説明

DB_CreateMapping 命令で関連付けたテーブルからレコードを抽出し、DB マップ変数「MapVar」に取得します。
複数のレコードを取得する場合は、DB マップ変数を配列として定義します。
DB マップ変数に取得したレコード数は「RecCnt」に出力します。抽出条件で抽出した結果のレコード数は「SelectedCnt」に出力します。

DB マップ変数の配列要素数と、取得したレコード数「RecCnt」および抽出した結果のレコード数「SelectedCnt」の関係について説明します。
【DB マップ変数の配列要素数 ≦抽出した結果のレコード数の場合】
DB マップ変数の最大要素まで、レコードを取得します。
例えば MapVar の配列要素数が 10 で、抽出したレコード数が 30 の場合、MapVar[0]から MapVar[9]までレコードを取得します。「RecCnt」は 10、「SelectedCnt」は 30 を出力します。

【DB マップ変数の配列要素数 ＞ 抽出した結果のレコード数 の場合】
抽出した結果のレコード数の要素まで、レコードを取得します。それより後の要素にはレコードを取得せず、以前の値を保持します。
例えば MapVar の配列要素数が 10 で、抽出したレコード数が 3 の場合、MapVar[0]から MapVar[2]までレコードを取得し、MapVar[3]から MapVar[9]の値は変化しません。「RecCnt」は 3、「SelectedCnt」は 3 を出力します。

取得するレコードは抽出条件「Where」(WHERE 句)で抽出します。「Where」は文字列で表現します。

抽出条件「Where」は、NULL 文字（16#00）だけを指定することはできません。命令が異常終了します。

抽出したレコードを並べ替えるにはソート条件「Sort」(ORDER BY 句)を指定します。「Sort」は文字列で表現します。
ソート条件を指定した場合は、ソート条件により並べ替えた順に、DB マップ変数に取得します。
ソート条件を指定しない場合は、DB マップ変数へのレコードの取得順は、接続する DB 種別の仕様に依存します。

WHERE 句、SORT 句でシングルクォートを使用する場合は、エスケープ文字（$' ）で指定します。
エスケープ文字については、『NJ/NX シリーズ CPU ユニット　ユーザーズマニュアル　ソフトウェア編（SBCA-467)』を参照してください。
WHERE 句、SORT 句の記述形式については、接続先の DB のマニュアルを参照してください。
抽出条件「Where」は以下のような値を指定します。

例 1）　特定のカラムの値が指定した値以上のレコードの値を取得する
「ColumnA」（符号なし整数）の値が、1234 以上のレコードの値を取得する
'" ColumnA" >= 1234'
生成する SQL 文：　SELECT FROM TableProduct Where "ColumnA" >= 1234
例 2）　2 つのカラムの値が指定した値の範囲にあるレコードを取得
「ColumnA」（符号なし整数）の値が、1000 より大きく、かつ、「ColumnB」（符号なし整数）の値が、2000 より小さいレコードを取得する
'" ColumnA" > 1000 AND "ColumnB" < 2000'
生成する SQL 文：　SELECT FROM TableProduct Where "ColumnA" > 1000 AND "ColumnB" < 2000
例 3）　特定のカラムの値が、指定した変数の値以上のレコードの値を取得する
「ColumnA」（符号なし整数）の値が、指定した変数の値以上の値を取得する
指定値：UINTVar := 1234;
抽出条件「Where」の入力パラメータ：WhereCond_Select := CONCAT('$" ColumnA$" >= ', UINT_TO_STRING(UINTVar));
生成する SQL 文：　SELECT FROM TableProduct Where "ColumnA" >= 1234

ソート条件「Sort」は以下のような値を指定します。

例）　2 つのカラムの値でソートしたレコードを取得する
「ColumnA」を昇順、「ColumnB」を降順にソートする
'" ColumnA" ASC, "ColumnB" DESC'
生成する SQL 文：　SELECT FROM TableProduct ORDER BY "ColumnA" ASC, "ColumnB" DESC

使用上の注意

- 本命令は、「Execute」の値が FALSE になっても、また実行時間がタスク周期を超えても、処理を最後まで続けます。処理が正常終了したかどうかは、「Done」の値が TRUE になることで確認してください。
- 「Execute」、「Done」、「Busy」、「Error」のタイミングチャートは、『NJ/NX シリーズコマンドリファレンスマニュアル　基本編』（SBCA-468）の“本章の見方”を参照してください。
- 本命令は、イベントタスク上で使うことができません。コンパイル時にエラーになります。
- 本命令は、抽出条件を指定せずに実行することはできません。
- 本命令を実行した結果、抽出したレコードが無い場合は、「RecCnt」、「SelectedCnt」ともに 0 で正常終了します。
- 本命令を実行した結果、DB マップ変数の配列要素数と抽出した結果のレコード数が一致しない場合も正常終了します。

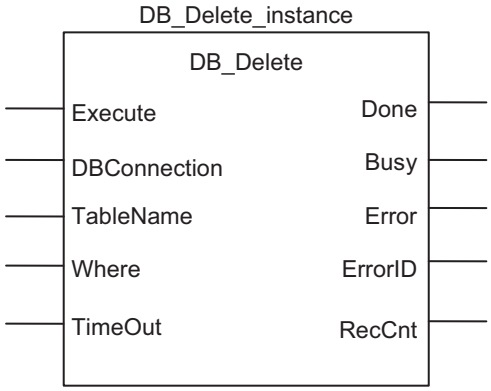
- DB 接続サービスをテストモードで開始している場合、本命令は実際には DB へ SELECT を実行せず正常終了します。DB マップ変数「MapVar」に値を取得せず、「RecCnt」、「SelectedCnt」は 0 を出力します。
- STRING 型のバイト数が不足していても、本命令は正常終了します。
  例)DB 側の任意のカラムが 12 文字の文字列を格納していて、対応する DB マップ変数のメンバのデータ型が STRING[11]の場合、11 文字までしか取得できませんが正常終了します。
- エラーコードが 300BHex（SQL 実行エラー）の場合は、DB_GetConnectionStatus 命令で、SQL 実行エラーの詳細情報を取得することができます。
- 命令実行タイムアウト時間の測定誤差の目安は、タスク実行時間の割合が 50%で 100 カラムのレコードの場合、+50ms です。ただし、タスク実行時間の割合およびカラム数によって、測定誤差が変動します。
- 同一の DB コネクションに対して DB 接続命令を同時に複数実行した場合、DB 接続サービスは命令を 1 個ずつ順番に実行します。2 個目以降に実行される命令の命令実行タイムアウト時間は、「Execute」の値が TRUE になった時点ではなく、その命令を DB 接続サービスが実行した時点から計測を始めます。そのため、「Execute」の値が TRUE になった時点から、その命令がタイムアウトするまでの時間は、命令実行タイムアウト時間で設定した値よりも長くなります。
- 本命令は、以下の場合、異常となります。「Error」は TRUE となります。
  a）DB 接続サービスが動作中でないときに、本命令を実行した場合
  b）DB 接続サービスが初期処理中に、本命令を実行した場合
  c）DB 接続サービスが異常停止中に、本命令を実行した場合
  d）DB 接続サービスが終了後／終了処理中に、本命令を実行した場合
  e）「DBConnection」が不正、または既に切断している場合
  f）「TimeOut」に有効範囲外の値を指定した場合
  g）「MapVar」で指定した変数を DB_CreateMapping でマッピングしていない場合
  h）実行した SQL 文が DB でエラーとなった場合
  i） NJ/NX シリーズ側のデータ型と DB 側のデータ型で、型変換できない場合
  j） ネットワーク障害などで DB と通信できない場合
  k）既にスプールに保留した SQL 文が存在する場合
  l） 指定した命令実行タイムアウト時間内に命令が終了しなかった場合
  m）前回実行した DB_Insert/DB_Update/DB_Select/DB_Delete 命令が命令実行タイムアウトし、サーバ側 DB の処理がまだ完了していない状態で命令を実行した場合
  n）DB 接続命令を同時に 32 個を超えて実行した場合

サンプルプログラム

サンプルプログラムは、DB_Delete 命令のサンプルプログラム「サンプルプログラム (P.7-43)」を参照してください。

DB_Delete（DB レコード削除）

指定したテーブルから条件に一致するレコードを削除します。

命令	名称	FB/FUN	グラフィック表現	ST 表現
DB_Delete	DB レコード削除	FB		DB_Delete_instance (Execute, DBConnection, TableName, Where, TimeOut, Done, Busy, Error, ErrorID, RecCnt);

（注）DB_Delete 命令のインスタンスとして、インスタンス名 DB_Delete_instance で変数宣言されている場合の例です。

変数

入力変数

入力変数	名称	データ型	有効範囲	単位	初期値	内容
Execute	起動	BOOL	FALSE, TRUE	-	FALSE	実行条件を指定します
DBConnection	DB コネクション	DWORD	16#00000000 ～ FFFFFFFF	-	16#00000000	DB_Connect で取得した DB コネクション
TableName	テーブル名	STRING	データ型に従う	-	''	DB のテーブル名
Where	抽出条件	STRING	最大 1986 バイト（終端 NULL 含む）*1	-	''	抽出条件(WHERE 句)を表す文字列を指定します（’WHERE’ は含みません）
TimeOut	命令実行タイムアウト時間	TIME	T#0s, T#0.05s ～ T#180s	-	T#0s	命令実行タイムアウト時間を指定します T#0s を指定した場合は、タイムアウト監視なしとなります

*1. DB 側が大文字／小文字を区別する場合、カラム名は以下のように指定してください。
MySQL と接続するとき、半角バッククォートで囲む
例）`ColumnA`
MySQL 以外と接続するとき、半角ダブルクォートで囲む
例）"ColumnA"

出力変数

出力変数	名称	データ型	有効範囲	単位	内容
Done	完了	BOOL	FALSE, TRUE	-	正常終了を表します
Busy	実行中	BOOL	FALSE, TRUE	-	処理中を表します
Error	エラー	BOOL	FALSE, TRUE	-	異常終了を表します
ErrorID	エラーコード	WORD	16#0000 ～ 16#FFFF	-	エラーコードを表します
RecCnt	レコード数	DINT	0 ～2147483647	-	削除したレコード数

関連するシステム定義変数

システム定義変数	名称	データ型	内容
_EIP_EtnOnlineSta	オンライン	BOOL	内蔵 EtherNet/IP ポートの通信機能の状態 TRUE:利用可能 FALSE:利用不可

関連するデバイスパラメータ

NX シリーズに接続した X バスユニットの Ethernet ポートを使用する場合は、X バスユニットのデバイスパラメータを参照してください。

関連するエラーコード一覧

エラーコード	名称	内容
0400Hex	入力値範囲外	「TimeOut」に有効範囲外の値を指定した場合
0406Hex	領域範囲外指定	「TableName」が NULL 文字（16#00）だけの文字列である場合
0410Hex	文字列フォーマット異常	「TableName」にスペース文字を含む文字列を指定した場合
041DHex	命令同時実行リソースオーバ	DB 接続命令を同時に 32 個を超えて実行した場合
3000Hex	DB 接続サービス未開始	DB 接続サービスが動作中でないときに、本命令を実行した場合
3002Hex	DB 接続サービス終了／終了処理中	DB 接続サービスが終了後／終了処理中に、本命令を実行した場合
3008Hex	DB コネクション不正	「DBConnection」が不正、または既に切断している場合
300BHex	SQL 実行エラー	実行した SQL 文が DB でエラーとなった場合
300EHex	抽出条件不正	「Where」が NULL 文字（16#00）だけの文字列である場合
3011Hex	DB コネクション異常切断発生中	ネットワーク障害などで DB と通信できない場合
3012Hex	DB 接続命令実行タイムアウト	指定した命令実行タイムアウト時間内に命令が終了しなかった場合
3013Hex	DB 接続サービス異常停止中	DB 接続サービスが異常停止中に、本命令を実行した場合
3014Hex	スプールデータあり	既にスプールに保留した SQL 文が存在するため実行できない場合
3015Hex	DB 接続サービス初期処理中	DB 接続サービスが初期処理中に、本命令を実行した場合
3016Hex	DB 処理中	前回実行した DB_Insert／DB_Update／DB_Select／DB_Delete 命令が命令実行タイムアウトし DB 側の処理がまだ完了していない状態で命令を実行した場合
301EHex	DB 接続サービス未起動	DB 接続サービスを［使用しない］に設定した状態で、本命令を実行した場合

機能説明

「TableName」で指定したテーブルから「Where」で指定した条件に一致するレコードを削除します。

削除するレコードは抽出条件「Where」(WHERE 句)で抽出します。「Where」は文字列で表現します。
抽出条件「Where」は、NULL 文字（16#00）だけを指定することはできません。命令が異常終了します。

WHERE 句でシングルクォートを使用する場合は、エスケープ文字（$' ）で指定します。
エスケープ文字については、『NJ/NX シリーズ CPU ユニット　ユーザーズマニュアル　ソフトウェア編（SBCA-467)』を参照してください。
WHERE 句の記述形式については、接続先の DB のマニュアルを参照してください。
「Where」の指定は以下のような値を指定します。

例） 2 つのカラムの何れかの値が指定した値と等しいレコードを削除する
「ColumnA」(符号なし整数）の値が、1000 と等しい、または、「ColumnB」(符号なし整数）の値が、2000 と等しいレコードを削除する
'" ColumnA" = 1000 OR "ColumnB" = 2000'
生成する SQL 文：　DELETE FROM TableProduct Where "ColumnA" = 1000 OR "ColumnB" = 2000

使用上の注意

- 本命令は、「Execute」の値が FALSE になっても、また実行時間がタスク周期を超えても、処理を最後まで続けます。処理が正常終了したかどうかは、「Done」の値が TRUE になることで確認してください。
- 「Execute」、「Done」、「Busy」、「Error」のタイミングチャートは、『NJ/NX シリーズコマンドリファレンスマニュアル　基本編』（SBCA-468）の“本章の見方”を参照してください。
- 本命令は、イベントタスク上で使うことができません。コンパイル時にエラーになります。
- 本命令は、抽出条件を指定せずに実行することはできません。
- DB 接続サービスをテストモードで開始している場合、本命令は実際には DB へ DELETE を実行せず正常終了します。
- エラーコードが 300BHex（SQL 実行エラー）の場合は、DB_GetConnectionStatus 命令で、SQL 実行エラーの詳細情報を取得することができます。
- 命令実行タイムアウト時間の測定誤差の目安は、タスク実行時間の割合が 50%で 100 カラムのレコードの場合、+50ms です。ただし、タスク実行時間の割合およびカラム数によって、測定誤差が変動します。
- 同一の DB コネクションに対して DB 接続命令を同時に複数実行した場合、DB 接続サービスは命令を 1 個ずつ順番に実行します。2 個目以降に実行される命令の命令実行タイムアウト時間は、「Execute」の値が TRUE になった時点ではなく、その命令を DB 接続サービスが実行した時点から計測を始めます。そのため、「Execute」の値が TRUE になった時点から、その命令がタイムアウトするまでの時間は、命令実行タイムアウト時間で設定した値よりも長くなります。
- 本命令は、以下の場合、異常となります。「Error」は TRUE となります。
  a）DB 接続サービスが動作中でないときに、本命令を実行した場合
  b）DB 接続サービスが初期処理中に、本命令を実行した場合
  c）DB 接続サービスが異常停止中に、本命令を実行した場合
  d）DB 接続サービスが終了後／終了処理中に、本命令を実行した場合
  e）「DBConnection」が不正、または既に切断している場合
  f）「TableName」が NULL 文字（16#00）だけの文字列である場合
  g）「TableName」にスペース文字を含む文字列を指定した場合
  h）「Where」が NULL 文字（16#00）だけの文字列である場合

i）「TimeOut」に有効範囲外の値を指定した場合
j） 実行した SQL 文が DB でエラーとなった場合
k）ネットワーク障害などで DB と通信できない場合
l） 既にスプールに保留した SQL 文が存在する場合
m）指定した命令実行タイムアウト時間内に命令が終了しなかった場合
n）前回実行した DB_Insert/DB_Update/DB_Select/DB_Delete 命令が命令実行タイムアウトし、サーバ側 DB の処理がまだ完了していない状態で命令を実行した場合
o）DB 接続命令を同時に 32 個を超えて実行した場合

サンプルプログラム

ここでは、接続対象のデータベース種別が Oracle の場合の、以下のサンプルプログラムを掲載します。

- DB レコード取得用トリガが TRUE になった時に DB のテーブルから指定したロット番号の生産データを取得
- 取得したレコードが複数ある場合は、生産終了時刻が最新のレコード以外のレコードを削除

DB コネクションと構造体型の設定

このサンプルプログラムで使用する必要最低限の設定を以下に示します。

● DB コネクションの設定

DB コネクション名：MyDatabase1

● 構造体型の設定

名称		データ型
PRODUCTION_SELECT		STRUCT
	Name	STRING[256]
	LotNo	STRING[32]
	Status	STRING[8]
	ProductionDate	DATE
	FinishTime	DATE_AND_TIME

ラダー図

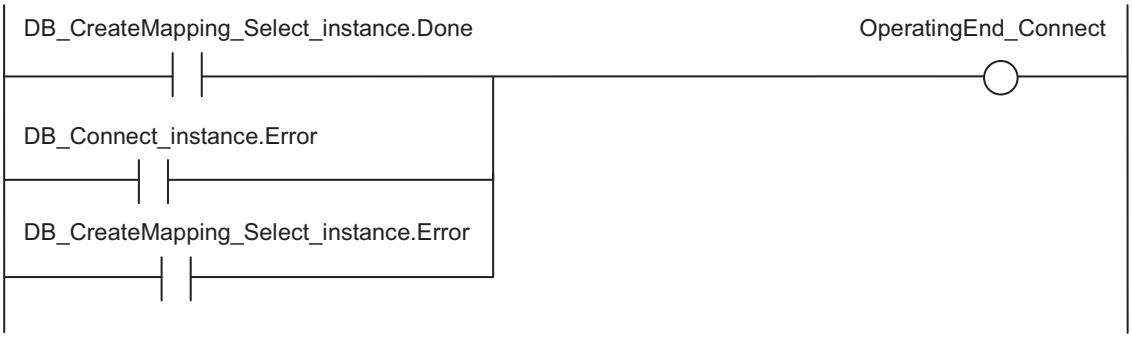
● 主な変数

名称	データ型	初期値	コメント
_DBC_Status	_sDBC_STATUS	-	DB 接続サービスのステータスを表すシステム定義変数です。
DB_Connect_instance	DB_Connect	-	DB_Connect 命令のインスタンスです。
MyDB1	DWORD	-	DB_Connect 命令のインスタンス DB_Connect_instance の出力 DBConnection に割り付けた変数です。
LotNo	UINT	1234	DB レコード取得、DB レコード削除の対象となるロット番号を指定する変数です。
Trigger_Connect	BOOL	FALSE	DB コネクション接続のトリガとなる変数です。

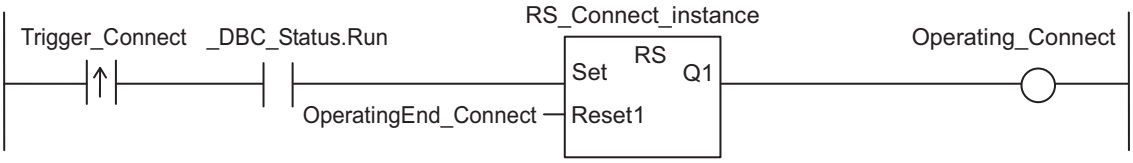
名称	データ型	初期値	コメント
RS_Connect_instance	RS	-	RS 命令のインスタンスです。
Operating_Connect	BOOL	FALSE	この変数が TRUE の時に、DB_Connect 命令を実行します。
OperatingEnd_Connect	BOOL	FALSE	DB_Connect 命令が終了すると、TRUE になります。
DB_CreateMapping _Select_instance	DB_CreateMapping	-	DB_CreateMapping 命令のインスタンスです。
MapVar_Select	ARRAY[0..9] OF PRODUCTION_SELECT		DB_CreateMapping 命令のインスタンス DB_CreateMapping _Select_instance の入力 MapVar に割り付けた変数です。
WhereCond_Select	STRING[256]	-	DB_Select 命令のインスタンス DB_Select_instance の入力 Where に割り付けた変数です。
SortCond_Select	STRING[256]	-	DB_Select 命令のインスタンス DB_Select_instance の入力 Sort に割り付けた変数です。
DB_Select_instance	DB_Select	-	DB_Select 命令のインスタンスです。
Trigger_Select	BOOL	FALSE	DB レコード取得のトリガとなる変数です。
RS_Select_instance	RS	-	RS 命令のインスタンスです。
Operating_Select	BOOL	FALSE	この変数が TRUE の時に、DB_Select 命令を実行します。
OperatingEnd_Select	BOOL	FALSE	DB_Select 命令が終了すると、TRUE になります。
WhereCond_Delete	STRING[256]	-	DB_Delete 命令のインスタンス DB_Delete_instance の入力 Where に割り付けた変数です。
Request_Delete	BOOL	FALSE	この変数が TRUE の時に、DB_Delete 命令を実行します。
DB_Delete_instance	DB_Delete	-	DB_Delete 命令のインスタンスです。
RS_Delete_instance	RS	-	RS 命令のインスタンスです。
Operating_Delete	BOOL	FALSE	この変数が TRUE の時に、DB_Delete 命令を実行します。
OperatingEnd_Delete	BOOL	FALSE	DB_Delete 命令が終了すると、TRUE になります。
DB_Close_instance	DB_Close	-	DB_Close 命令のインスタンスです。
Trigger_Close	BOOL	FALSE	DB コネクション切断のトリガとなる変数です。
RS_Close_instance	RS	-	RS 命令のインスタンスです。
Operating_Close	BOOL	FALSE	この変数が TRUE の時に、DB_Close 命令を実行します。
OperatingEnd_Close	BOOL	FALSE	DB_Close 命令が終了すると、TRUE になります。

● サンプルプログラム

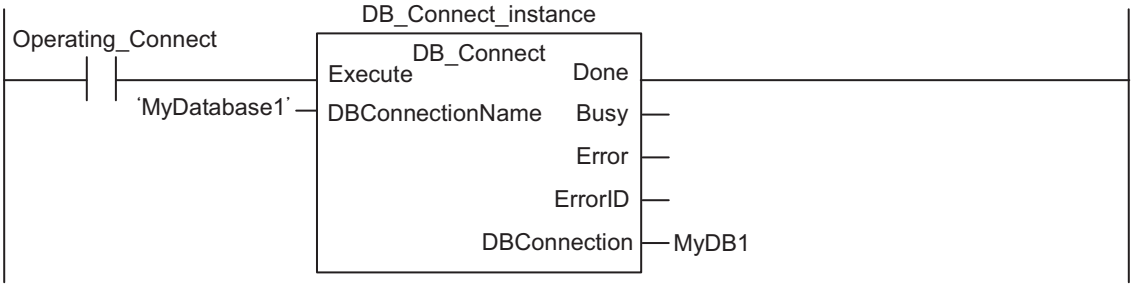
◆DB コネクション名「MyDatabase1」に接続し、DB のテーブルと変数を関連付けます
DB_Connect 命令、DB_CreateMapping 命令実行完了を判断する



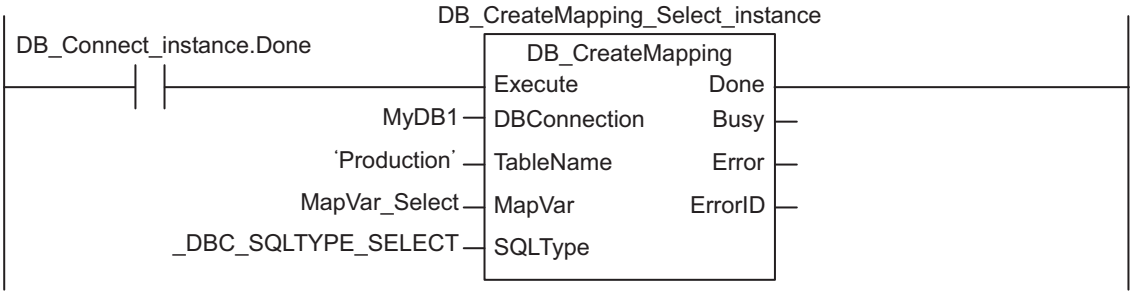
DB コネクション接続用トリガを受け付ける



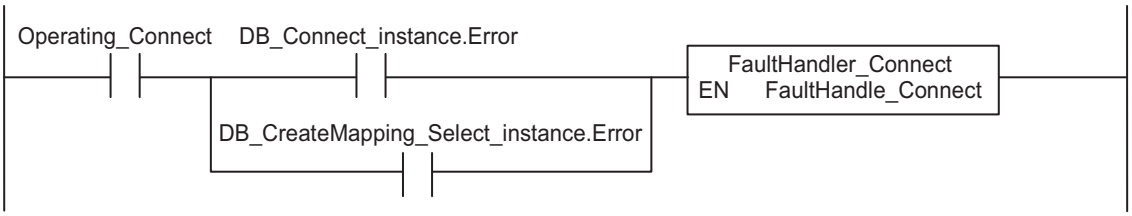
DB コネクション名「MyDataBase1」に接続する



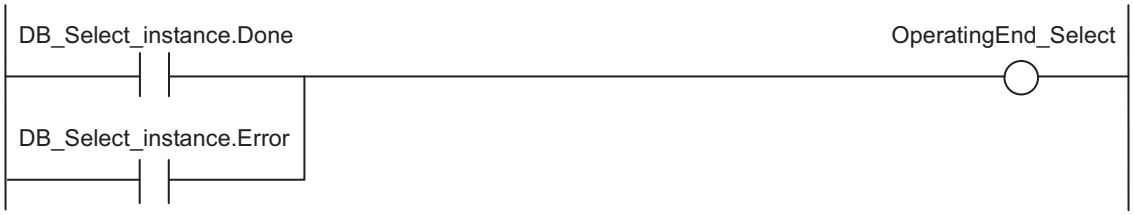
DB コネクション「MyDB1」の「Production」テーブルに、変数 MapVar_Select を SELECT 操作用に関連付ける



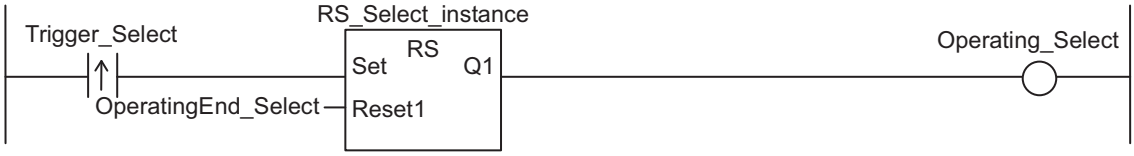
異常終了した場合、装置における異常終了時の手順（FaultHandler_Connect）を実行する。異常終了時の手順については別途、設計してください



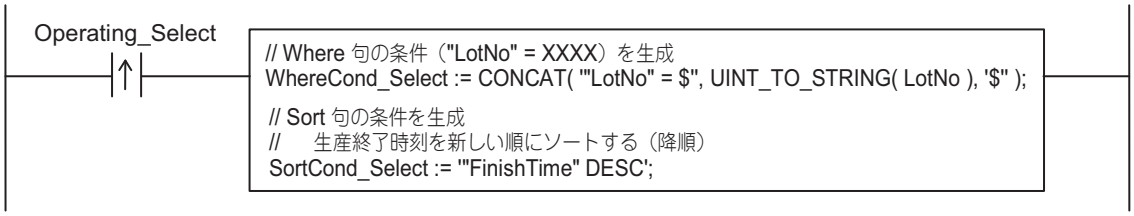
◆変数 Trigger_Select が TRUE になった場合に、DB コネクション「MyDB1」から指定したロット番号のレコードを取得します。
DB_Select 命令実行完了を判断する



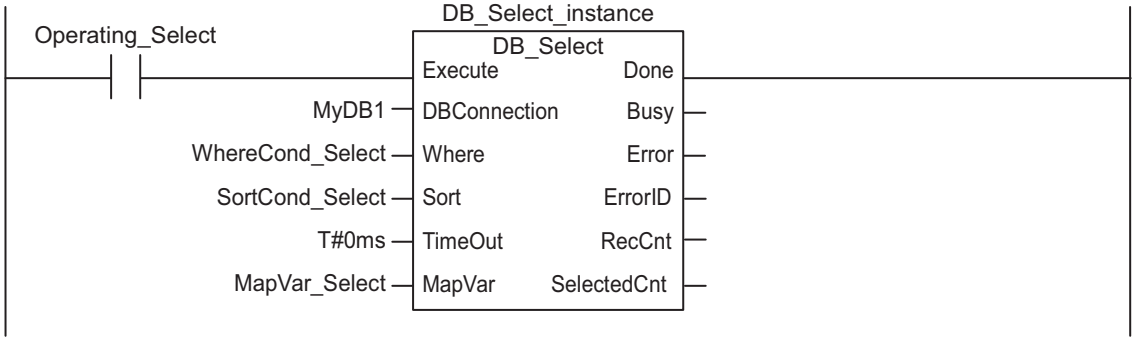
DB レコード取得用トリガを受け付ける



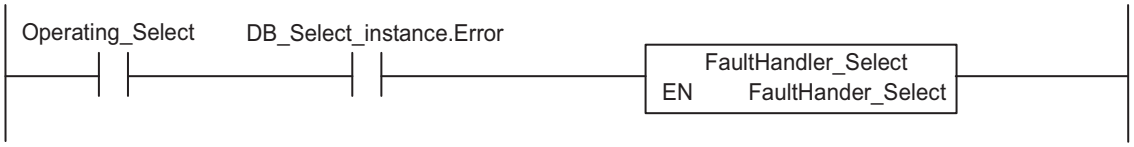
Where 句、Sort 句の条件を生成する。



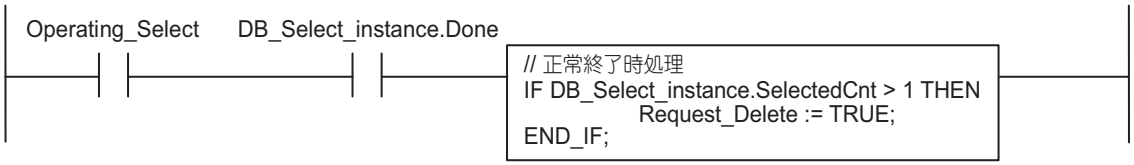
DB コネクション「MyDB1」のレコードを取得する。
命令実行のタイムアウト監視は行わない。



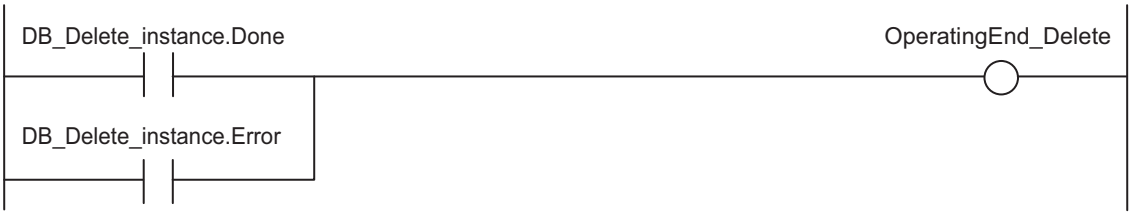
異常終了した場合、装置における異常終了時の手順（FaultHandler_Select）を実行する。
異常終了時の手順については別途、設計してください



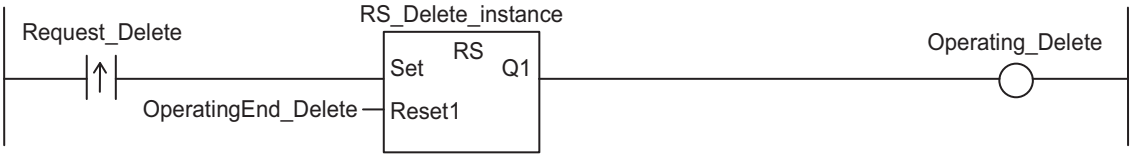
抽出レコード数が 2 個以上であれば、レコード削除を行う



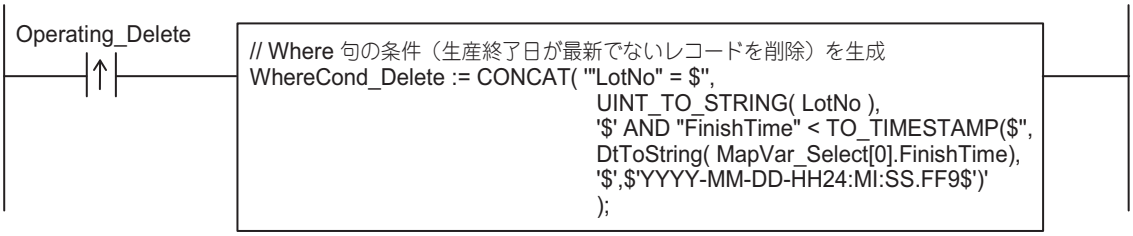
◆DB のテーブルから生産終了日が最新でないレコードを削除します。
DB_Delete 命令実行完了を判断する



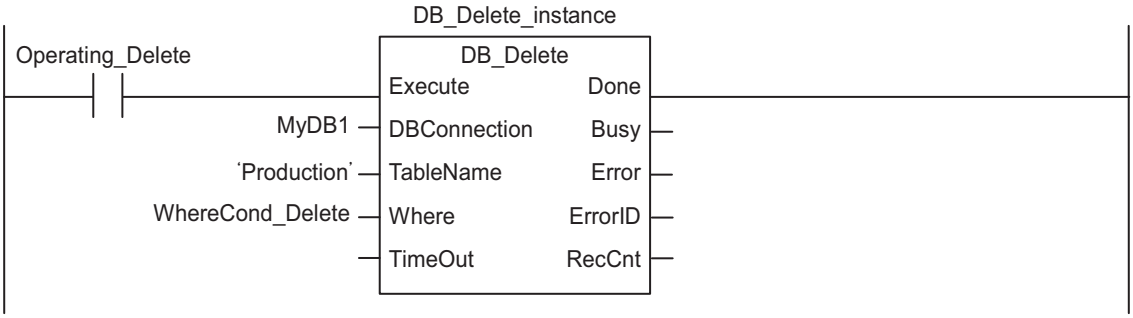
DB レコード削除用トリガを受け付ける



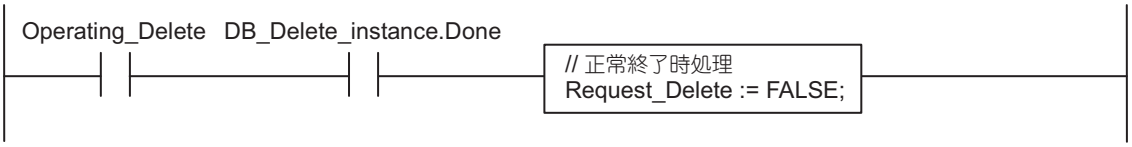
Where 句の条件を生成する



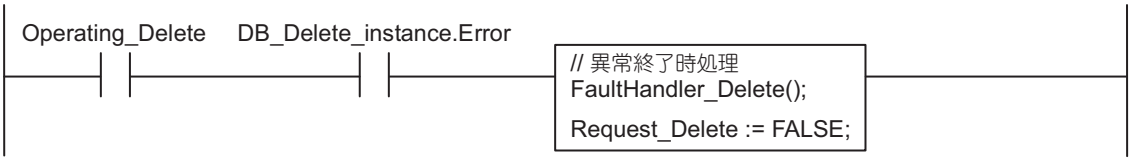
DB コネクション「MyDB1」の Production テーブルのレコードを削除する。命令実行のタイムアウト監視は行わない。



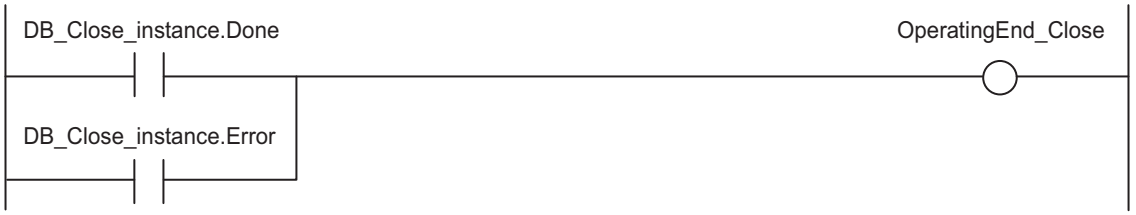
正常終了時の処理を実行する



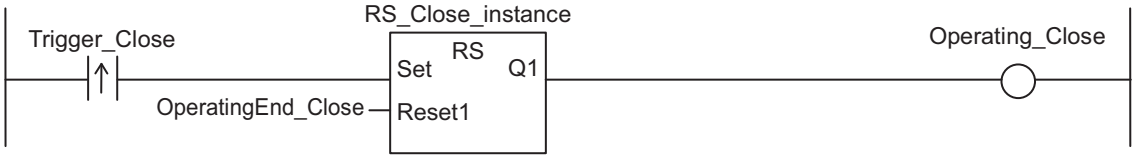
異常終了した場合、装置における異常終了時の手順（FaultHandler_Delete）を実行する。
異常終了時の手順については別途、設計してください



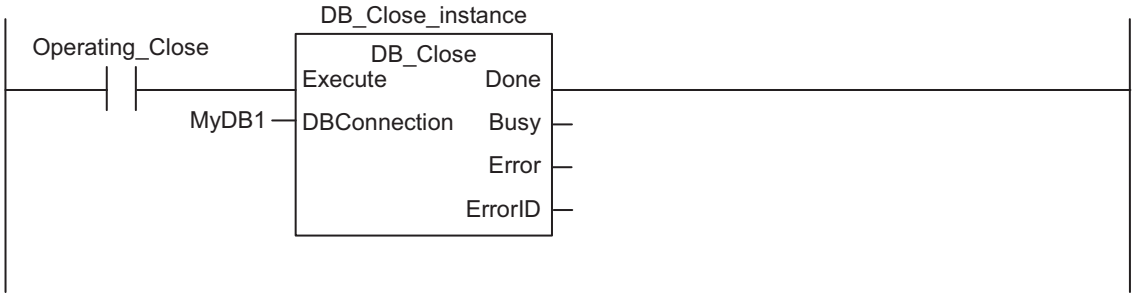
◆DB コネクション「MyDB1」との DB コネクションを切断する。
DB_Close 命令実行完了を判断する



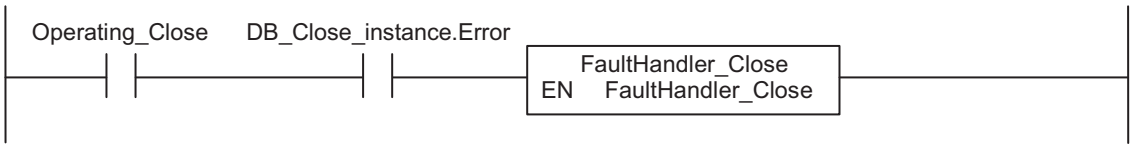
DB コネクション切断用トリガを受け付ける



DB コネクション「MyDB1」との DB コネクションを切断する



異常終了した場合、装置における異常終了時の手順（FaultHandler_Close）を実行する。
異常終了時の手順については別途、設計してください



ストラクチャードテキスト（ST）

● 主な変数

名称	データ型	初期値	コメント
_DBC_Status	_sDBC_STATUS	-	DB 接続サービスのステータスを表すシステム定義変数です。
DB_Connect_instance	DB_Connect	-	DB_Connect 命令のインスタンスです。
MyDB1	DWORD	-	DB_Connect 命令のインスタンス DB_Connect_instance の出力 DBConnection に割り付けた変数です。
LotNo	UINT	1234	DB レコード取得、DB レコード削除の対象となるロット番号を指定する変数です。
Trigger_Connect	BOOL	FALSE	DB コネクション接続のトリガとなる変数です。
LastTrigger_Connect	BOOL	FALSE	前回実行時のトリガ状態を保持する変数です。
Operating_Connect	BOOL	FALSE	この変数が TRUE の時に、DB_Connect 命令を実行します。
OperatingStart_Connect	BOOL	FALSE	この変数が TRUE の時に、DB コネクション接続開始時の処理を実行します。
DB_CreateMapping_Select_instance	DB_CreateMapping	-	DB_CreateMapping 命令のインスタンスです。

名称	データ型	初期値	コメント
MapVar_Select	ARRAY[0..99] OF PRODUCTION_SELECT	-	DB_CreateMapping 命令のインスタンス DB_CreateMapping _Select_instance の入力 MapVar に割り付けた変数です。
DB_Select_instance	DB_Select	-	DB_Select 命令のインスタンスです。
Trigger_Select	BOOL	FALSE	DB レコード取得トリガとなる変数です。
LastTrigger_Select	BOOL	FALSE	前回実行時のトリガ状態を保持する変数です。
Operating_Select	BOOL	FALSE	この変数が TRUE の時に、DB_Select 命令を実行します。
OperatingStart_Select	BOOL	FALSE	この変数が TRUE の時に、DB レコード取得開始時の処理を実行します。
WhereCond_Select	STRING[256]	-	DB_Select 命令のインスタンス DB_Select_instance の入力 Where に割り付けた変数です。
SortCond_Select	STRING[256]	-	DB_Select 命令のインスタンス DB_Select_instance の入力 Sort に割り付けた変数です。
DB_Delete_instance	DB_Delete	-	DB_Delete 命令のインスタンスです。
WhereCond_Delete	STRING[256]	-	DB_Delete 命令のインスタンス DB_Delete_instance の入力 Where に割り付けた変数です。
Request_Delete	BOOL	FALSE	この変数が TRUE の時に、DB_Delete 命令を実行します。
LastRequest_Delete	BOOL	FALSE	前回実行時の要求状態を保持する変数です。
Operating_Delete	BOOL	FALSE	この変数が TRUE の時に、DB_Delete 命令を実行します。
OperatingStart_Delete	BOOL	FALSE	この変数が TRUE の時に、DB レコード削除開始時の処理を実行します。
DB_Close_instance	DB_Close	-	DB_Close 命令のインスタンスです。
Trigger_Close	BOOL	FALSE	DB コネクション切断のトリガとなる変数です。
LastTrigger_Close	BOOL	FALSE	前回実行時のトリガ状態を保持する変数です。
Operating_Close	BOOL	FALSE	この変数が TRUE の時に、DB_Close 命令を実行します。
OperatingStart_Close	BOOL	FALSE	この変数が TRUE の時に、DB コネクション切断開始時の処理を実行します。
Stage	INT	-	DB コネクション接続の状態を表す変数です。

● サンプルプログラム

```
// ------------------------------------------------------------------------------------------
// ◆DB コネクション名「MyDatabese1」に接続し、DB のテーブルと変数を関連付けます

// 変数 Trigger_Connect の立ち上がりでシーケンス開始
IF ( (Trigger_Connect=TRUE)
   AND (LastTrigger_Connect=FALSE)
   AND (_DBC_Status.Run=TRUE) ) THEN
      OperatingStart_Connect := TRUE;
      Operating_Connect := TRUE;
```

```
END_IF;
LastTrigger_Connect:=Trigger_Connect;

// シーケンス開始時処理
IF (OperatingStart_Connect=TRUE) THEN
   // DB 接続命令インスタンス初期化
   DB_Connect_instance( Execute:=FALSE );
   DB_CreateMapping_Select_instance(
      Execute  := FALSE,
      MapVar   := MapVar_Select,
      SQLType  := _DBC_SQLTYPE_SELECT
   );

Stage := 1;
OperatingStart_Connect := FALSE;
END_IF;

// DB コネクション名「MyDatabese1」に接続します。
// DB コネクション「MyDB1」の「Production」テーブルに、変数 MapVar_Select を SELECT 操作用に関連付けます。
IF (Operating_Connect=TRUE) THEN
   CASE Stage OF
   1 : // DB コネクション接続
      DB_Connect_instance(
         Execute           := TRUE,
         DBConnectionName  := 'MyDatabase1',
         DBConnection      => MyDB1
      );

IF (DB_Connect_instance.Done=TRUE) THEN
   Stage := INT#2; // 正常終了
END_IF;
IF (DB_Connect_instance.Error=TRUE) THEN
   Stage := INT#99; // 異常終了
END_IF;

2 : // DB のテーブルと変数を関連付け
   DB_CreateMapping_Select_instance(
      Execute       := TRUE,
      DBConnection  := MyDB1,
      TableName     := 'Production',
      MapVar        := MapVar_Select,
      SQLType       := _DBC_SQLTYPE_SELECT
   );

   IF (DB_CreateMapping_Select_instance.Done=TRUE) THEN
      Operating_Connect:=FALSE; // 正常終了
   END_IF;
   IF (DB_CreateMapping_Select_instance.Error=TRUE) THEN
      Stage := INT#99; // 異常終了
   END_IF;
```

```
    99 :
       // 異常終了時処理
       // 装置の異常発生時の手順（FaultHandler_Connect）については、実装してください。
                  FaultHandler_Connect();
       Operating_Connect := FALSE;
    END_CASE;
END_IF;

    // ----------------------------------------------------------------------------------------
// ◆DB コネクション「MyDB1」から指定したロット番号のレコードを取得します。

// 変数 Trigger_Select の立ち上がりでシーケンス開始
IF ( (Trigger_Select=TRUE) AND (LastTrigger_Select=FALSE) ) THEN
    OperatingStart_Select := TRUE;
    Operating_Select := TRUE;
END_IF;
LastTrigger_Select := Trigger_Select;

// シーケンス開始時処理
IF (OperatingStart_Select=TRUE) THEN
    // DB 接続命令インスタンス初期化
    DB_Select_instance( Execute:=FALSE, MapVar:=MapVar_Select );

    // Where 句の条件（"LotNo" = XXXX）を生成します。
    WhereCond_Select := CONCAT( '"LotNo" = $', UINT_TO_STRING( LotNo ), '$' );

    // Sort 句の条件を生成します。
    // 生産終了時刻を新しい順にソートする（降順）
    SortCond_Select := '"FinishTime" DESC';

       OperatingStart_Select := FALSE;
    END_IF;

    // DB コネクション「MyDB1」のレコードを取得します。命令実行のタイムアウト監視は行いません。
    IF (Operating_Select=TRUE) THEN
       // レコード取得
       DB_Select_instance(
          Execute       := TRUE,
          DBConnection  := MyDB1,
          Where         := WhereCond_Select,
          Sort          := SortCond_Select,
          MapVar        := MapVar_Select
       );

    IF (DB_Select_instance.Done=TRUE) THEN
       // 抽出レコード数が 2 以上であれば、レコード削除を行います。
       IF (DB_Select_instance.SelectedCnt > 1) THEN
          Request_Delete := TRUE;
       END_IF;
       Operating_Select:=FALSE; // 正常終了
    END_IF;
```

```
    IF (DB_Select_instance.Error=TRUE) THEN
        // 異常終了時処理
        // 装置の異常発生時の手順（FaultHandler_Select）については、実装してください。
        FaultHandler_Select();

        Operating_Select := FALSE;
    END_IF;
END_IF;

    // ----------------------------------------------------------------------------------------
// ◆DB のテーブルから生産終了日が最新でないレコードを削除します。

// 変数 Request_Delete の立ち上がりでシーケンス開始
IF ( (Request_Delete=TRUE) AND (LastRequest_Delete=FALSE) ) THEN
    OperatingStart_Delete := TRUE;
    Operating_Delete := TRUE;
END_IF;
LastRequest_Delete := Request_Delete;

// シーケンス開始時処理
IF (OperatingStart_Delete=TRUE) THEN
    // DB 接続命令インスタンス初期化
    DB_Delete_instance( Execute:=FALSE );

    // Where 句の条件（生産終了日が最新でないレコードを削除）を生成します。
    WhereCond_Delete := CONCAT( '"LotNo" = $'',
                                UINT_TO_STRING( LotNo ),
                                '$' AND "FinishTime" < TO_TIMESTAMP($'',
                                DtToString( MapVar_Select[0].FinishTime),
                                '$',$'YYYY-MM-DD-HH24:MI:SS.FF9$')'
                                );
    OperatingStart_Delete := FALSE;
END_IF;

// DB コネクション「MyDB1」の'Production'テーブルのレコードを削除します。命令実行のタイムアウト監視は
行いません。
IF (Operating_Delete=TRUE) THEN
    // レコード削除
    DB_Delete_instance(
        Execute       := TRUE,
        DBConnection  := MyDB1,
        TableName     := 'Production',
        Where         := WhereCond_Delete
    );

    IF (DB_Delete_instance.Done=TRUE) THEN
        Operating_Delete :=FALSE; // 正常終了
        Request_Delete :=FALSE;
    END_IF;
    IF (DB_Delete_instance.Error=TRUE) THEN
        // 異常終了時処理
```

```
      // 装置の異常発生時の手順（FaultHandler_Delete）については、実装してください。
      FaultHandler_Delete();

      Operating_Delete := FALSE;
      Request_Delete :=FALSE;
   END_IF;
END_IF
// ----------------------------------------------------------------------------------------
// ◆DB コネクション「MyDB1」との DB コネクションを切断します。

// 変数 Trigger_Close の立ち上がりでシーケンス開始
IF ( (Trigger_Close=TRUE) AND (LastTrigger_Close=FALSE) ) THEN
   OperatingStart_Close := TRUE;
   Operating_Close := TRUE;
END_IF;
LastTrigger_Close := Trigger_Close;

// シーケンス開始時処理
IF (OperatingStart_Close=TRUE) THEN
   // DB 接続命令インスタンス初期化
   DB_Close_instance( Execute:=FALSE );

   OperatingStart_Close := FALSE;
END_IF;

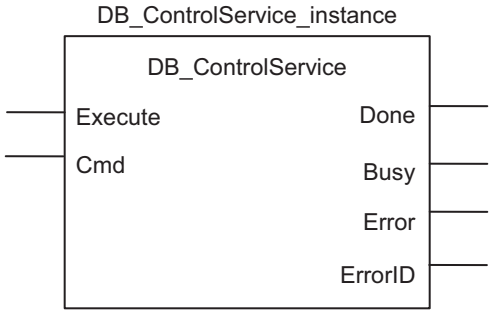
// DB コネクション「MyDB1」との DB コネクションを切断します。
IF (Operating_Close=TRUE) THEN
   // DB コネクション切断
   DB_Close_instance( Execute:=TRUE, DBConnection:=MyDB1 );

   IF (DB_Close_instance.Done=TRUE) THEN
      Operating_Close := FALSE; // 正常終了
   END_IF;
   IF (DB_Close_instance.Error=TRUE) THEN
      // 異常終了時処理
      // 装置の異常発生時の手順（FaultHandler_Close）については、実装してください。
      FaultHandler_Close();

      Operating_Close := FALSE;
   END_IF;
END_IF;
```

DB_ControlService（DB 接続サービスの制御）

DB 接続サービスの開始／停止、デバッグログの記録開始／記録終了を行います。

命令	名称	FB/FUN	グラフィック表現	ST 表現
DB_ControlService	DB 接続サービスの制御	FB		DB_ControlService_instance (Execute, Cmd, Done, Busy, Error, ErrorID);

（注）DB_ControlService 命令のインスタンスとして、インスタンス名 DB_ControlService_instance で変数宣言されている場合の例です。

変数

入力変数

入力変数	名称	データ型	有効範囲	単位	初期値	内容
Execute	起動	BOOL	FALSE, TRUE	-	FALSE	実行条件を指定します
Cmd	コマンド	_eDBC_CMD	_DBC_CMD_START(1)：稼働モードで開始 _DBC_CMD_START_TEST(2)：テストモードで開始 _DBC_CMD_STOP(3)：サービスを停止 _DBC_CMD_DEBUGLOG_ON(4)：デバッグログ記録開始 _DBC_CMD_DEBUGLOG_OFF(5)：デバッグログ記録終了		_DBC_CMD_START	実行するコマンド

出力変数

出力変数	名称	データ型	有効範囲	単位	内容
Done	完了	BOOL	FALSE, TRUE	-	正常終了を表します
Busy	実行中	BOOL	FALSE, TRUE	-	処理中を表します
Error	エラー	BOOL	FALSE, TRUE	-	異常終了を表します
ErrorID	エラーコード	WORD	16#0000 ～ 16#FFFF	-	エラーコードを表します

関連するシステム定義変数

システム定義変数	名称	データ型	有効範囲	単位	内容
_DBC_Status.Idle	DB 接続サービス状態アイドル	BOOL	FALSE, TRUE	-	DB 接続サービスの稼働状態がアイドル状態のとき、TRUE となります。それ以外では FALSE となります。
_DBC_Status.Run	DB 接続サービス状態動作中フラグ	BOOL	FALSE, TRUE	-	DB 接続サービスを稼働モード、または、テストモードで開始すると、TRUE となります。 DB 接続サービスを停止すると、FALSE となります。
_DBC_Status.Test	DB 接続サービス状態テストモード	BOOL	FALSE, TRUE	-	DB 接続サービスをテストモードで開始すると、TRUE となります。DB 接続サービスを停止すると、FALSE となります。
_DBC_Status.Shutdown	DB 接続サービス状態終了	BOOL	FALSE, TRUE	-	DB 接続サービスの稼働状態が終了状態のとき、TRUE となります。それ以外では FALSE となります。

関連するデバイスパラメータ

NX シリーズに接続した X バスユニットの Ethernet ポートを使用する場合は、X バスユニットのデバイスパラメータを参照してください。

関連するエラーコード一覧

エラーコード	名称	内容
0400Hex	入力値範囲外	「Cmd」に列挙子にない値を指定した場合
041DHex	命令同時実行リソースオーバ	DB 接続命令を同時に 32 個を超えて実行した場合
1400Hex	SD メモリカード使用不可	SD メモリカードが使用可能状態ではないのに_DBC_CMD_DEBUGLOG_ON を実行した場合
1401Hex	SD メモリカードライトプロテクト	SD メモリカードがライトプロテクト中に「Cmd」で _DBC_CMD_DEBUGLOG_ON を実行した場合
3001Hex	DB 接続サービス動作モード変更失敗	稼働モードで動作中に「Cmd」で_DBC_CMD_START_TEST を実行した場合 テストモードで動作中に「Cmd」で_DBC_CMD_START を実行した場合 DB 接続サービスが停止処理中に、DB 接続サービスの開始を指示した場合
3002Hex	DB 接続サービス終了／終了処理中	DB 接続サービスが終了後／終了処理中に、本命令を実行した場合
3013Hex	DB 接続サービス異常停止中	DB 接続サービスが異常停止中に、本命令を実行した場合
3015Hex	DB 接続サービス初期処理中	DB 接続サービスが初期処理中に、本命令を実行した場合
301EHex	DB 接続サービス未起動	DB 接続サービスを［使用しない］に設定した状態で、本命令を実行した場合

機能説明

DB 接続サービスの開始／停止、デバッグログの記録開始／記録終了を行います。

DB が接続可能な環境で、DB 接続サービスを開始するには、DB 接続サービスを稼働モードで開始します。

開発時など DB が存在しない環境では、DB 接続サービスをテストモードで開始します。この場合、DB_Connect 命令、DB_CreateMapping 命令、DB_Insert 命令、DB_Update 命令、DB_Select 命令、DB_Delete 命令は、実際には DB への接続、SQL 文の実行をせず正常終了します。
DB 接続サービスを停止すると、接続中のコネクションはすべて切断されます。

デバッグログの記録を開始すると、DB 接続命令を実行した時の詳細のログ(送信した SQL 文など)を SD メモリカードのデバッグログファイルに出力します。

使用上の注意

- 本命令は、「Execute」の値が FALSE になっても、また実行時間がタスク周期を超えても、処理を最後まで続けます。処理が正常終了したかどうかは、「Done」の値が TRUE になることで確認してください。
- 「Execute」、「Done」、「Busy」、「Error」のタイミングチャートは、『NJ/NX シリーズコマンドリファレンスマニュアル　基本編』（SBCA-468）の“本章の見方”を参照してください。
- 本命令は、イベントタスク上で使うことができません。コンパイル時にエラーになります。
- DB 接続サービスの開始を行う場合は、_DBC_Status.Idle の値が TRUE であることを確認してから本命令を実行してください。DB 接続サービスが初期処理中に本命令を実行した場合、異常終了（DB 接続サービス初期処理中）します。
- DB 接続サービス動作中に、DB 接続サービスの動作モードを稼働モードからテストモードへ変更できません。またテストモードから稼働モードへ変更できません。一度 DB 接続サービスを停止してください。
- デバッグログの記録状態(デバッグログ記録開始/終了)は、DB 接続サービスを停止/開始しても保持します。
- 本命令以外に、デバッグログを停止する条件は、以下のとおりです。
  a）DB_Shutdown 命令を実行した場合
  b）CPU ユニットの電源を OFF した場合
  c）SD メモリカードを取り出した場合
- 本命令は、以下の場合、異常となります。「Error」は TRUE になります。
  a）DB 接続サービスが初期処理中に、本命令を実行した場合
  b）DB 接続サービスが異常停止中に、本命令を実行した場合
  c）DB 接続サービスが終了後／終了処理中に、本命令を実行した場合
  d）稼働モードで動作中に「Cmd」で_DBC_CMD_START_TEST を実行した場合
  e）テストモードで動作中に「Cmd」で_DBC_CMD_START を実行した場合
  f）DB 接続サービスが停止処理中に、DB 接続サービスの開始を指示した場合
  g）SD メモリカードが使用可能状態ではない、またはライトプロテクト中に「Cmd」で_DBC_CMD_DEBUGLOG_ON を実行した場合
  h）「Cmd」に列挙子にない値を指定した場合
  i）DB 接続命令を同時に 32 個を超えて実行した場合

サンプルプログラム

ここでは、デバッグログ制御用トリガが TRUE になった時にデバッグログの記録を開始し、デバッグログ制御用トリガが OFF になった時にデバッグログの記録を終了するサンプルプログラムを掲載します。

ラダー図

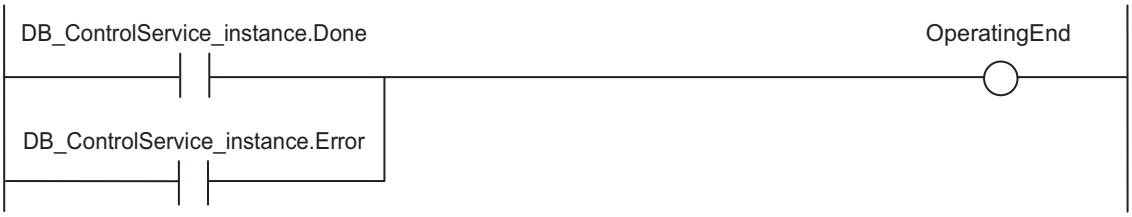
● 主な変数

名称	データ型	初期値	コメント
DB_ControlService_instance	DB_ControlService	-	DB_ControlService 命令のインスタンスです。
LogOn	BOOL	FALSE	デバッグログ制御のトリガとなる変数です。
Operating	BOOL	FALSE	この変数が TRUE の時に、DB_ControlService 命令を実行します。
OperatingEnd	BOOL	FALSE	DB_ControlService 命令が終了すると、TRUE になります。
RS_instance	RS	-	RS 命令のインスタンスです。
MyCmd	_eDBC_CMD	-	DB_ControlService 命令のインスタンス DB_ControlService_instance の入力 Cmd に割り付けた変数です。
ControlService_OK	BOOL	FALSE	DB_ControlService 命令が正常終了すると、TRUE になります。

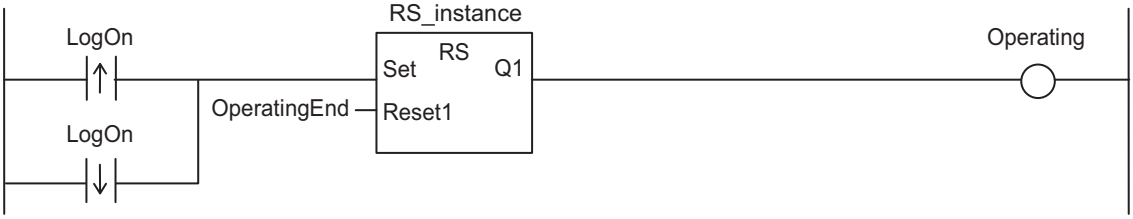
● サンプルプログラム

◆変数 LogOn の立ち上がりでデバッグログの記録を開始します。変数 LogOn の立ち下がりでデバッグログの記録を終了します。

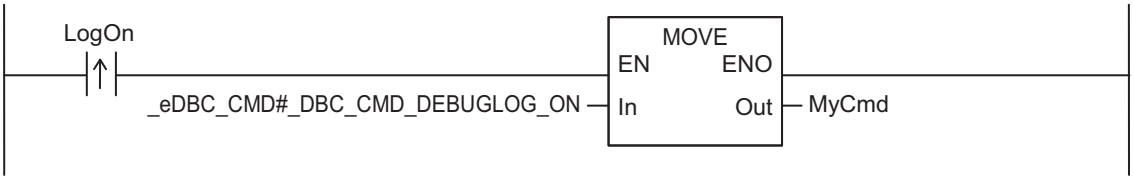
DB_ControlService 命令実行完了を判断する



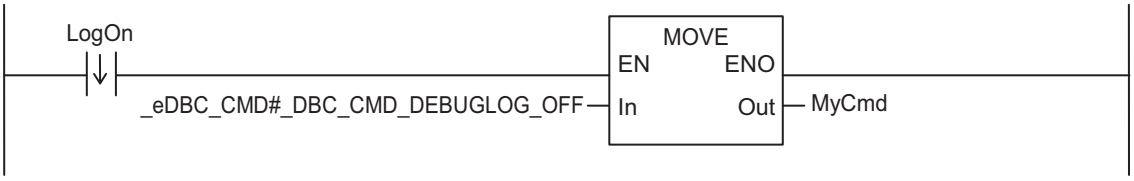
デバッグログ制御用トリガを受け付ける



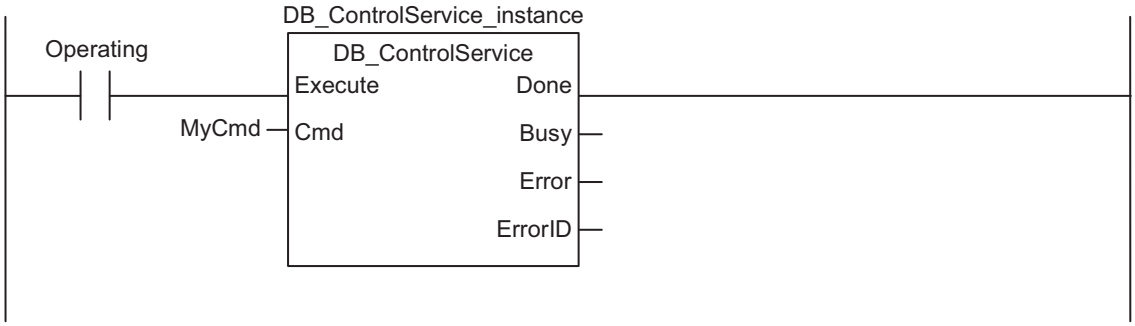
デバッグログの記録を開始する



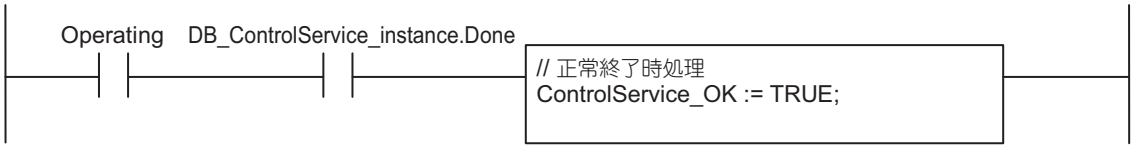
デバッグログの記録を終了する



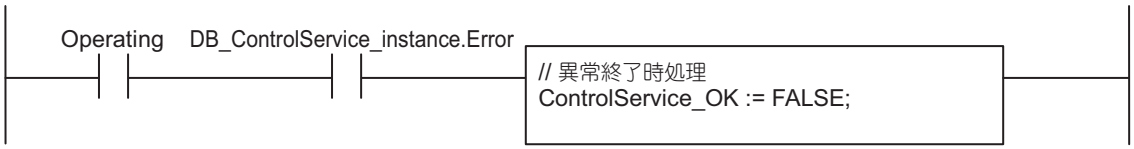
デバッグログ記録開始、またはデバッグログ記録終了を指示する



正常終了した場合、変数 ControlService_OK を TRUE にする



異常終了した場合、変数 ControlService_OK を FALSE にする



ストラクチャードテキスト（ST）

● 主な変数

名称	データ型	初期値	コメント
DB_ControlService_instance	DB_ControlService	-	DB_ControlService 命令のインスタンスです。
LogOn	BOOL	FALSE	デバッグログ制御のトリガとなる変数です。
LastTrigger	BOOL	FALSE	前回実行時のトリガ状態を保持する変数です。
Operating	BOOL	FALSE	この変数が TRUE の時に、DB_ControlService 命令を実行します。
OperatingStart	BOOL	FALSE	この変数が TRUE の時に、初期処理を実行します。
MyCmd	_eDBC_CMD	-	DB_ControlService 命令のインスタンス DB_ControlService_instance の入力 Cmd に割り付けた変数です。

● サンプルプログラム

```
(* ------------------------------------------------------------------------------
   ◆変数 LogOn の立ち上がりでデバッグログの記録を開始します。
      変数 LogOn の立ち下がりでデバッグログの記録を終了します。
   ------------------------------------------------------------------------------ *)

// 変数 LogOn の立ち上がりでシーケンス開始
IF ( (LogOn=TRUE) AND (LastTrigger=FALSE) ) THEN
   OperatingStart := TRUE;
   Operating := TRUE;
   MyCmd := _DBC_CMD_DEBUGLOG_ON;       // デバッグログ記録開始
```

```
ELSIF ( (LogOn=FALSE) AND (LastTrigger=TRUE) ) THEN
   OperatingStart := TRUE;
   Operating := TRUE;
   MyCmd := _DBC_CMD_DEBUGLOG_OFF;   // デバッグログ記録終了
END_IF;
LastTrigger := LogOn;

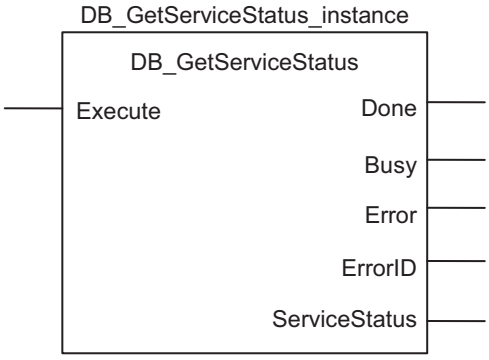
// シーケンス開始時処理
IF (OperatingStart=TRUE) THEN
   // 命令インスタンス初期化
   DB_ControlService_instance( Execute:=FALSE );
   OperatingStart := FALSE;
END_IF;

// デバッグログ記録開始、またはデバッグログ記録終了を指示します。
IF (Operating=TRUE) THEN
   // デバッグログ記録開始、または終了
   DB_ControlService_instance(
      Execute  := TRUE,
      Cmd      := MyCmd
   );

   IF (DB_ControlService_instance.Done=TRUE) THEN
      // 正常終了時処理
      Operating := FALSE;
   END_IF;
   IF (DB_ControlService_instance.Error=TRUE) THEN
      // 異常終了時処理
      Operating := FALSE;
   END_IF;
END_IF;
```

DB_GetServiceStatus（DB 接続サービスの状態取得）

DB 接続サービスの現在の状態を取得します。

命令	名称	FB/FUN	グラフィック表現	ST 表現
DB_GetServiceStatus	DB 接続サービスの状態取得	FB		DB_GetServiceStatus_instance (Execute, Done, Busy, Error, ErrorID, ServiceStatus);

（注）DB_GetServiceStatus 命令のインスタンスとして、インスタンス名 DB_GetServiceStatus_instance で変数宣言されている場合の例です。

変数

入力変数

入力変数	名称	データ型	有効範囲	単位	初期値	内容
Execute	起動	BOOL	FALSE, TRUE	-	FALSE	実行条件を指定します

出力変数

出力変数	名称	データ型	有効範囲	単位	内容
Done	完了	BOOL	FALSE, TRUE	-	正常終了を表します
Busy	実行中	BOOL	FALSE, TRUE	-	処理中を表します
Error	エラー	BOOL	FALSE, TRUE	-	異常終了を表します
ErrorID	エラーコード	WORD	16#0000 ～ 16#FFFF	-	エラーコードを表します
ServiceStatus	DB 接続サービス状態	_sDBC_SERVICE_STATUS	データ型に従う		DB 接続サービスの状態を表します。

関連するデバイスパラメータ

NX シリーズに接続した X バスユニットの Ethernet ポートを使用する場合は、X バスユニットのデバイスパラメータを参照してください。

関連するエラーコード一覧

エラーコード	名称	内容
041DHex	命令同時実行リソースオーバ	DB 接続命令を同時に 32 個を超えて実行した場合
3002Hex	DB 接続サービス終了／終了処理中	DB 接続サービスが終了後／終了処理中に、本命令を実行した場合
3013Hex	DB 接続サービス異常停止中	DB 接続サービスが異常停止中に、本命令を実行した場合
3015Hex	DB 接続サービス初期処理中	DB 接続サービスが初期処理中に、本命令を実行した場合
301EHex	DB 接続サービス未起動	DB 接続サービスを［使用しない］に設定した状態で、本命令を実行した場合

機能説明

DB 接続サービスの現在の状態を「ServiceStatus」に取得します。
取得できる状態については「DB 接続命令の共通入力変数・出力変数 (P.7-2)」、の「ServiceStatus (P.7-3)」を参照してください。

使用上の注意

- 本命令は、「Execute」の値が FALSE になっても、また実行時間がタスク周期を超えても、処理を最後まで続けます。処理が正常終了したかどうかは、「Done」の値が TRUE になることで確認してください。
- 「Execute」、「Done」、「Busy」、「Error」のタイミングチャートは、『NJ/NX シリーズコマンドリファレンスマニュアル　基本編』（SBCA-468）の“本章の見方”を参照してください。
- 本命令は、イベントタスク上で使うことができません。コンパイル時にエラーになります。
- 本命令は、以下の場合、異常となります。「Error」は TRUE となります。
  a）DB 接続サービスが初期処理中に、本命令を実行した場合
  b）DB 接続サービスが異常停止中に、本命令を実行した場合
  c）DB 接続サービスが終了後／終了処理中に、本命令を実行した場合
  d）DB 接続命令を同時に 32 個を超えて実行した場合

サンプルプログラム

ここでは、以下のサンプルプログラムを掲載します。
- トリガが TRUE になった時に DB 接続サービスの状態を取得
- 異常実行回数が 100 回以上であれば変数 Warning の値を TRUE にする

ラダー図

● 主な変数

名称	データ型	初期値	コメント
DB_GetServiceStatus_instance	DB_GetServiceStatus	-	DB_GetServiceStatus 命令のインスタンスです。
Trigger	BOOL	FALSE	DB 接続サービスの状態取得のトリガとなる変数です。
Operating	BOOL	FALSE	この変数が TRUE の時に、DB_GetServiceStatus 命令を実行します。

名称	データ型	初期値	コメント
OperatingEnd	BOOL	FALSE	DB_GetServiceStatus 命令が終了すると、TRUE になります。
RS_instance	RS	-	RS 命令のインスタンスです。
MyStatus	_sDBC_SERVICE_STATUS	-	DB_GetServiceStatus 命令のインスタンス DB_GetServiceStatus_instance の出力 ServiceStatus に割り付けた変数です。
Warning	BOOL	FALSE	異常実行回数が 100 回以上の時に、TRUE になります。
GetServiceStatus_OK	BOOL	FALSE	DB_GetServiceStatus 命令が正常終了すると、TRUE になります。

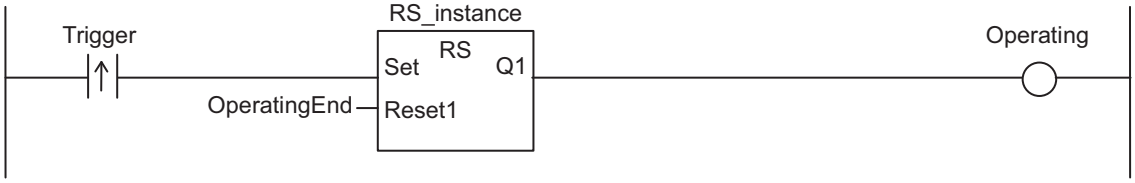
● サンプルプログラム

◆異常実行回数が 100 回以上であれば変数 Warning の値を TRUE にします。

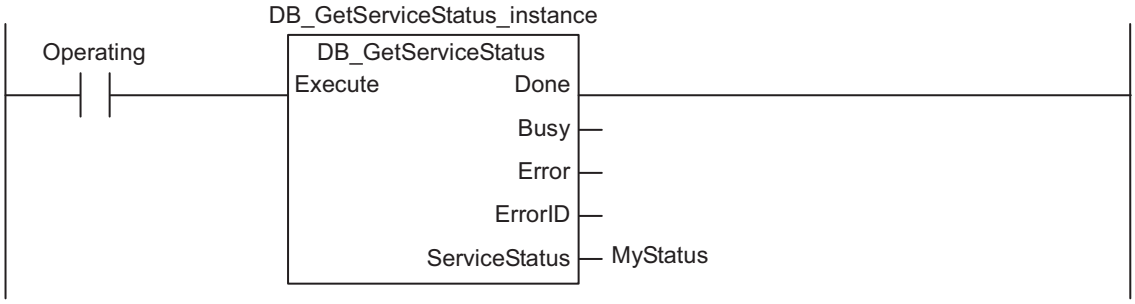
DB_GetServiceStatus 命令実行完了を判断する



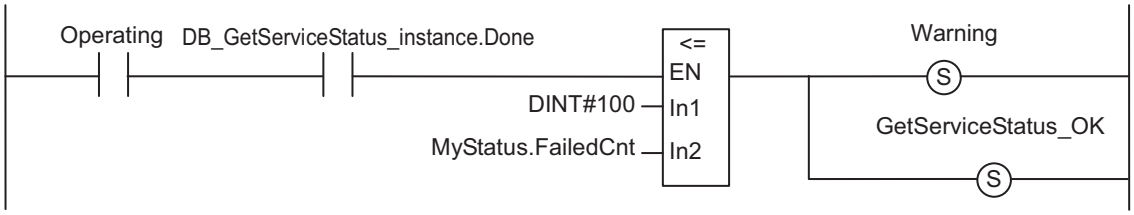
トリガを受け付ける



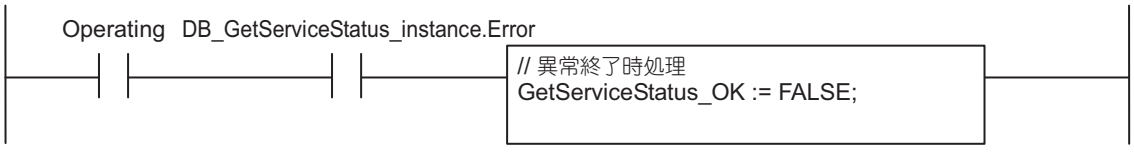
DB 接続サービスの状態を取得する



正常終了した場合、異常実行回数が 100 回以上であれば、変数 Warning を TRUE にする



異常終了した場合、変数 Warning を FALSE にする



ストラクチャードテキスト（ST）

● 主な変数

名称	データ型	初期値	コメント
DB_GetServiceStatus_instance	DB_GetServiceStatus	-	DB_GetServiceStatus 命令のインスタンスです。
Trigger	BOOL	FALSE	DB 接続サービスの状態取得のトリガとなる変数です。
LastTrigger	BOOL	FALSE	前回実行時のトリガ状態を保持する変数です。
Operating	BOOL	FALSE	この変数が TRUE の時に、DB_GetServiceStatus 命令を実行します。
OperatingStart	BOOL	FALSE	この変数が TRUE の時に、初期処理を実行します。
MyStatus	_sDBC_SERVICE_STATUS	-	DB_GetServiceStatus のインスタンス DB_GetServiceStatus_instance の出力 ServiceStatus に割り付けた変数です。
Warning	BOOL	FALSE	異常実行回数が 100 回以上の時に、TRUE になります。

● サンプルプログラム

```
(* ---------------------------------------------------------------------------------
   ◆すべてのコネクションで SQL 文の実行に失敗した回数が 100 回以上であれば、変数 Warning の
     値を TRUE にします
   --------------------------------------------------------------------------------- *)

// 変数 Trigger の立ち上がりでシーケンス開始
IF ( (Trigger=TRUE) AND (LastTrigger=FALSE) ) THEN
   OperatingStart := TRUE;
   Operating := TRUE;
END_IF;
LastTrigger := Trigger;

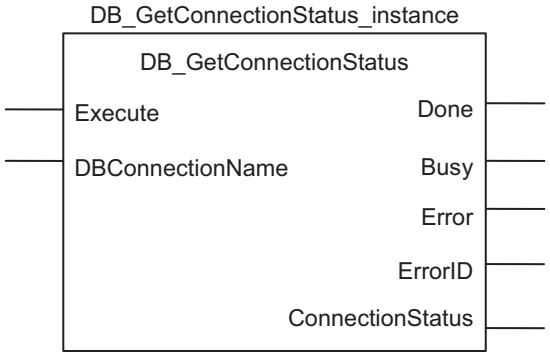
// シーケンス開始時処理
IF (OperatingStart=TRUE) THEN
   // 命令インスタンス初期化
   DB_GetServiceStatus_instance( Execute:=FALSE );
   OperatingStart := FALSE;
END_IF;

IF (Operating=TRUE) THEN
   // DB 接続サービスの状態を取得します。
   DB_GetServiceStatus_instance(
     Execute       := TRUE,
     ServiceStatus => MyStatus
   );
```

```
    IF (DB_GetServiceStatus_instance.Done=TRUE) THEN
      // 正常終了時処理
      // 異常実行回数が 100 回以上であれば、変数 Warning を TRUE にします。
      IF (MyStatus.FailedCnt >= DINT#100) THEN
         Warning := TRUE;
      END_IF;
      Operating := FALSE;
    END_IF;
    IF (DB_GetServiceStatus_instance.Error=TRUE) THEN
        // 異常終了時処理
        Operating := FALSE;
    END_IF;
END_IF;
```

DB_GetConnectionStatus（DB コネクションの状態取得）

DB コネクションの状態を取得します。

命令	名称	FB/FUN	グラフィック表現	ST 表現
DB_GetConnectionStatus	DB コネクションの状態取得	FB		DB_GetConnectionStatus_instance (Execute, DBConnectionName, Done, Busy, Error, ErrorID, ConnectionStatus);

（注）DB_GetConnectionStatus 命令のインスタンスとして、インスタンス名 DB_GetConnectionStatus_instance で変数宣言されている場合の例です。

変数

入力変数

入力変数	名称	データ型	有効範囲	単位	初期値	内容
Execute	起動	BOOL	FALSE, TRUE	-	FALSE	実行条件を指定します
DBConnectionName	DB コネクション名	STRING	最大 17 バイト（終端 NULL 含む）	-	''	Sysmac Studio で設定した DB コネクション名を指定します。

出力変数

出力変数	名称	データ型	有効範囲	単位	内容
Done	完了	BOOL	FALSE, TRUE	-	正常終了を表します
Busy	実行中	BOOL	FALSE, TRUE	-	処理中を表します
Error	エラー	BOOL	FALSE, TRUE	-	異常終了を表します

出力変数	名称	データ型	有効範囲	単位	内容
ErrorID	エラーコード	WORD	16#0000 ～ 16#FFFF	-	エラーコードを表します
ConnectionStatus	コネクション状態	_sDBC_CONNECTION_STATUS	データ型に従う		DBConnectionName で指定したコネクションの状態を表します。

関連するシステム定義変数

システム定義変数	名称	データ型	内容
_EIP_EtnOnlineSta	オンライン	BOOL	内蔵 EtherNet/IP ポートの通信機能の状態 TRUE:利用可能 FALSE:利用不可

関連するデバイスパラメータ

NX シリーズに接続した X バスユニットの Ethernet ポートを使用する場合は、X バスユニットのデバイスパラメータを参照してください。

関連するエラーコード一覧

エラーコード	名称	内容
0406Hex	領域範囲外指定	「DBConnectionName」が NULL 文字（16#00）だけの文字列である場合
0410Hex	文字列フォーマット異常	「DBConnectionName」にスペース文字を含む文字列を指定した場合 「DBConnectionName」が終端 NULL 文字で終わらない場合
041DHex	命令同時実行リソースオーバ	DB 接続命令を同時に 32 個を超えて実行した場合
3000Hex	DB 接続サービス未開始	DB 接続サービスが動作中でないときに、本命令を実行した場合
3002Hex	DB 接続サービス終了／終了処理中	DB 接続サービスが終了後／終了処理中に、本命令を実行した場合
3003Hex	DB コネクション名不正	「DBConnectionName」が DB コネクション設定に設定されていない場合
3013Hex	DB 接続サービス異常停止中	DB 接続サービスが異常停止中に、本命令を実行した場合
3015Hex	DB 接続サービス初期処理中	DB 接続サービスが初期処理中に、本命令を実行した場合
301EHex	DB 接続サービス未起動	DB 接続サービスを［使用しない］に設定した状態で、本命令を実行した場合

機能説明

「DBConnection」で指定した DB コネクションの状態を「ConnectionStatus」に取得します。
取得できる状態については、「DB 接続命令の共通入力変数・出力変数 (P.7-2)」の「ConnectionStatus (P.7-4)」を参照してください。

DB 応答時間の測定方法については、「A-2-3 DB 応答時間の測定方法 (P.A-23)」を参照してください。

使用上の注意

- 本命令は、「Execute」の値が FALSE になっても、また実行時間がタスク周期を超えても、処理を最後まで続けます。処理が正常終了したかどうかは、「Done」の値が TRUE になることで確認してください。
- 「Execute」、「Done」、「Busy」、「Error」のタイミングチャートは、『NJ/NX シリーズコマンドリファレンスマニュアル　基本編』（SBCA-468）の“本章の見方”を参照してください。
- 本命令は、イベントタスク上で使うことができません。コンパイル時にエラーになります。
- DB_Connect 命令の実行が完了する前に本命令を実行し、DB コネクションの接続状態が“接続中”であることを確認した上で、次の DB 接続命令を実行しても、命令エラー（DB コネクション不正）が発生することがあります。DB_Connect 命令の出力変数「DBConnection」を使用する場合は、DB_Connect 命令の出力変数「Done」が TRUE であること、または「DBConnection」の値が 16#00000000 以外であることを確認した上で、DB 接続命令を実行してください。
- 本命令は、以下の場合、異常となります。「Error」は TRUE となります。
  a）DB 接続サービスが動作中でないときに、本命令を実行した場合
  b）DB 接続サービスが初期処理中に、本命令を実行した場合
  c）DB 接続サービスが異常停止中に、本命令を実行した場合
  d）DB 接続サービスが終了後／終了処理中に、本命令を実行した場合
  e）「DBConnectionName」が DB コネクション設定に設定されていない場合
  f）「DBConnectionName」が NULL 文字（16#00）だけの文字列である場合
  g）「DBConnectionName」にスペース文字を含む文字列を指定した場合
  h）「DBConnectionName」が終端 NULL 文字で終わらない場合
  i）DB 接続命令を同時に 32 個を超えて実行した場合

サンプルプログラム

ここでは、以下のサンプルプログラムを掲載します。
- トリガが TRUE になった時に DB コネクションの状態を取得
- スプール使用率が 80%を超えていたら、変数 Warning の値を TRUE にする

ラダー図

● 主な変数

名称	データ型	初期値	コメント
DB_GetConnectionStatus_instance	DB_GetConnectionStatus	-	DB_GetConnectionStatus 命令のインスタンスです。
Trigger	BOOL	FALSE	DB コネクションの状態取得のトリガとなる変数です。
Operating	BOOL	FALSE	この変数が TRUE の時に、DB_GetConnectionStatus 命令を実行します。
OperatingEnd	BOOL	FALSE	DB_GetConnectionStatus 命令が終了すると、TRUE になります。
RS_instance	RS	-	RS 命令のインスタンスです。
MyStatus	_sDBC_CONNECTION_STATUS	-	DB_GetConnectionStatus 命令のインスタンス DB_GetConnectionStatus_instance の出力 ConnectionStatus に割り付けた変数です。

名称	データ型	初期値	コメント
Warning	BOOL	FALSE	スプール使用率が 80%を超える時に、TRUE になります。
GetConnectionStatus_OK	BOOL	FALSE	DB_GetConnectionStatus 命令が正常終了すると、TRUE になります。

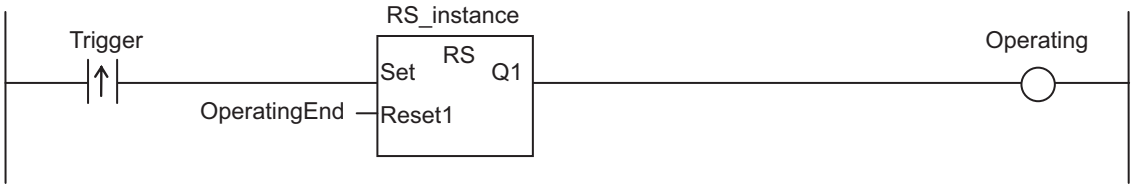
● **サンプルプログラム**

◆DB コネクション名「MyDatabase1」のスプール使用率が 80%を超えていたら、変数 Warning の値を TRUE にします。

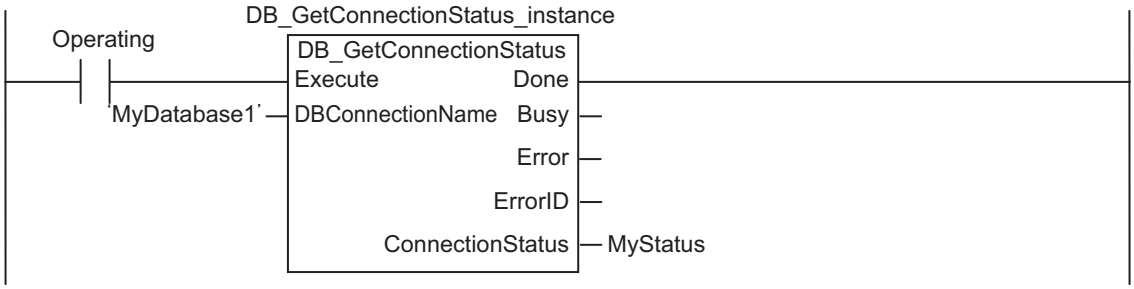
DB_GetConnectionStatus 命令実行完了を判断する



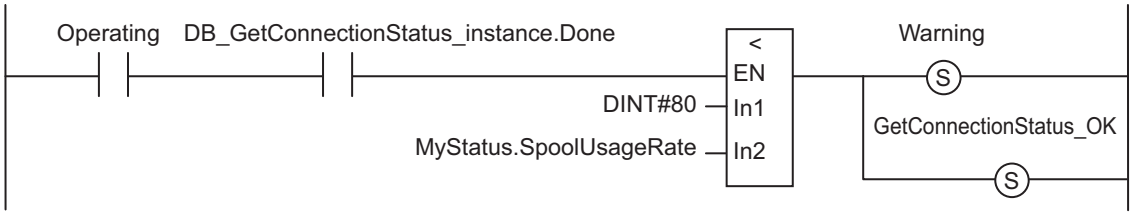
トリガを受け付ける



DB コネクションの状態を取得する



正常終了した場合、スプール使用率が 80%を超えていれば、変数 Warning を TRUE にする



異常終了した場合、変数 Warning を FALSE にする



ストラクチャードテキスト（ST）

● 主な変数

名称	データ型	初期値	コメント
DB_GetConnectionStatus_instance	DB_GetConnectionStatus	-	DB_GetConnectionStatus 命令のインスタンスです。
Trigger	BOOL	FALSE	DB コネクションの状態取得のトリガとなる変数です。
LastTrigger	BOOL	FALSE	前回実行時のトリガ状態を保持する変数です。
Operating	BOOL	FALSE	この変数が TRUE の時に、DB_GetConnectionStatus 命令を実行します。
OperatingStart	BOOL	FALSE	この変数が TRUE の時に、初期処理を実行します。
MyStatus	_sDBC_CONNECTION_STATUS	-	DB_GetConnectionStatus 命令のインスタンス DB_GetConnectionStatus_instance の出力 ConnectionStatus に割り付けた変数です。
Warning	BOOL	FALSE	スプール使用率が 80%を超える時に、TRUE になります。

● サンプルプログラム

```
(* ---------------------------------------------------------------------------------
   ◆DB コネクション名「MyDatabese1」のスプール使用率が 80%を超えていたら、
変数 Warning の値を TRUE にします
    --------------------------------------------------------------------------------- *)

// 変数 Trigger の立ち上がりでシーケンス開始
IF ( (Trigger=TRUE) AND (LastTrigger=FALSE) ) THEN
   OperatingStart := TRUE;
   Operating := TRUE;
END_IF;
LastTrigger := Trigger;

// シーケンス開始時処理
IF (OperatingStart=TRUE) THEN
   // 命令インスタンス初期化
   DB_GetConnectionStatus_instance( Execute:=FALSE );
   OperatingStart := FALSE;
END_IF;

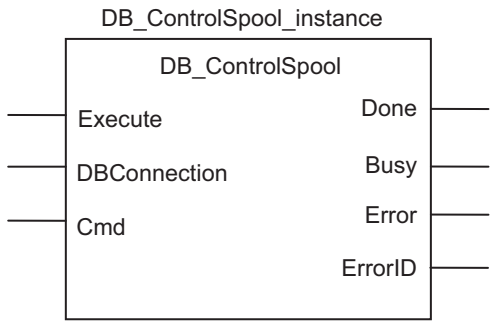
IF (Operating=TRUE) THEN
   // DB コネクションの状態を取得します。
   DB_GetConnectionStatus_instance(
     Execute            := TRUE,
     DBConnectionName   := 'MyDatabase1',
     ConnectionStatus   => MyStatus
   );

   IF (DB_GetConnectionStatus_instance.Done=TRUE) THEN
      // 正常終了時処理
```

```
        // スプール使用率が80%を超えていたら、変数 Warning を TRUE にします。
        IF (MyStatus.SpoolUsageRate > SINT#80) THEN
            Warning := TRUE;
        END_IF;
        Operating := FALSE;
    END_IF;
    IF (DB_GetConnectionStatus_instance.Error=TRUE) THEN
        // 異常終了時処理
        Operating := FALSE;
    END_IF;
END_IF;
```

DB_ControlSpool（スプールの再送・クリア）

DB_Insert 命令、DB_Update 命令でスプールした SQL 文を再送、またはクリアします。

命令	名称	FB/FUN	グラフィック表現	ST 表現
DB_ControlSpool	スプールの再送・クリア	FB		DB_ControlSpool_instance (Execute, DBConnection, Cmd, Done, Busy, Error, ErrorID);

（注）DB_ControlSpool 命令のインスタンスとして、インスタンス名 DB_ControlSpool_instance で変数宣言されている場合の例です。

変数

入力変数

入力変数	名称	データ型	有効範囲	単位	初期値	内容
Execute	起動	BOOL	FALSE, TRUE	-	FALSE	実行条件を指定します
DBConnection	DBコネクション	DWORD	16#00000000～FFFFFFFF		16#00000000	DB_Connectで取得したDBコネクション
Cmd	コマンド	_eDBC_SPOOL_CMD	_DBC_SPOOL_CLEAR(1)：クリア _DBC_SPOOL_RESEND(2)：再送		_DBC_SPOOL_CLEAR	実行するコマンド

出力変数

出力変数	名称	データ型	有効範囲	単位	内容
Done	完了	BOOL	FALSE, TRUE	-	正常終了を表します
Busy	実行中	BOOL	FALSE, TRUE	-	処理中を表します
Error	エラー	BOOL	FALSE, TRUE		異常終了を表します
ErrorID	エラーコード	WORD	16#0000～16#FFFF	-	エラーコードを表します

関連するシステム定義変数

システム定義変数	名称	データ型	内容
_EIP_EtnOnlineSta	オンライン	BOOL	内蔵 EtherNet/IP ポートの通信機能の状態 TRUE:利用可能 FALSE:利用不可

関連するデバイスパラメータ

NX シリーズに接続した X バスユニットの Ethernet ポートを使用する場合は、X バスユニットのデバイスパラメータを参照してください。

関連するエラーコード一覧

エラーコード	名称	内容
0400Hex	入力値範囲外	「Cmd」に列挙子にない値を指定した場合
041DHex	命令同時実行リソースオーバ	DB 接続命令を同時に 32 個を超えて実行した場合
3000Hex	DB 接続サービス未開始	DB 接続サービスが動作中でないときに、本命令でスプールの再送を実行した場合
3002Hex	DB 接続サービス終了／終了処理中	DB 接続サービスが終了後／終了処理中に、本命令を実行した場合
3008Hex	DB コネクション不正	「DBConnection」が不正、または既に切断している場合
300BHex	SQL 実行エラー	実行した SQL 文が DB でエラーとなった場合
3011Hex	DB コネクション異常切断発生中	ネットワーク障害などで DB と通信できない場合
3013Hex	DB 接続サービス異常停止中	DB 接続サービスが異常停止中に、本命令を実行した場合
3015Hex	DB 接続サービス初期処理中	DB 接続サービスが初期処理中に、本命令を実行した場合
301EHex	DB 接続サービス未起動	DB 接続サービスを［使用しない］に設定した状態で、本命令を実行した場合

機能説明

「DBConnection」で指定した DB コネクションのスプールに保留した SQL 文を再送、またはクリアすることができます。
スプールの再送方法が手動再送の場合、本命令でスプールに保留した SQL 文を再送します。

使用上の注意

- 本命令は、「Execute」の値が FALSE になっても、また実行時間がタスク周期を超えても、処理を最後まで続けます。処理が正常終了したかどうかは、「Done」の値が TRUE になることで確認してください。
- スプールの再送を指定した場合、本命令では、スプールの再送処理の起動のみを行います。「Done」の値が TRUE になっても、スプールに保留された SQL 文の再送処理は完了していません。DB_GetConnectionStatus 命令で“スプールデータ数”を読み出し、スプールの再送が完了しているかを確認してください。
- スプール機能を使用しない場合は、スプールに保留された SQL 文の再送、およびクリアは実行せず、正常終了となります。
- スプールのクリアは、DB 接続サービスが動作中でない場合でも実行できます。
- 「Execute」、「Done」、「Busy」、「Error」のタイミングチャートは、『NJ/NX シリーズコマンドリファレンスマニュアル　基本編』（SBCA-468）の“本章の見方”を参照してください。
- 本命令は、イベントタスク上で使うことができません。コンパイル時にエラーになります。

・本命令が以下の場合、異常となります。「Error」は TRUE になります。
a）DB 接続サービスが動作中でないときに、本命令でスプールの再送を実行した場合
b）DB 接続サービスが初期処理中に、本命令を実行した場合
c）DB 接続サービスが異常停止中に、本命令を実行した場合
d）DB 接続サービスが終了後／終了処理中に、本命令を実行した場合
e）「DBConnection」が不正、または既に切断している場合
f）「Cmd」に列挙子にない値を指定した場合
g）実行した SQL 文が DB でエラーとなった場合
h）ネットワーク障害などで DB と通信できない場合
i） DB 接続命令を同時に 32 個を超えて実行した場合

サンプルプログラム

ここでは、トリガが TRUE になった時に DB コネクションの接続状態が“接続中”であれば、スプールに保留された SQL 文を再送するサンプルプログラムを掲載します。

ラダー図

● 主な変数

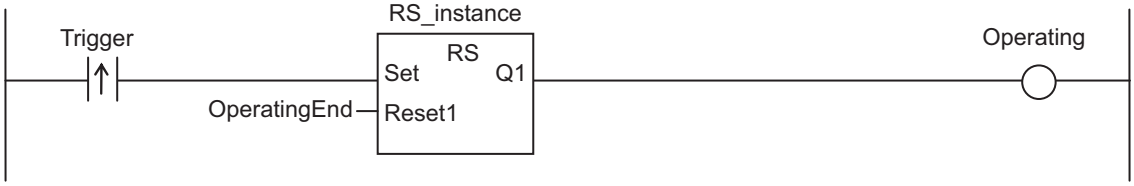
名称	データ型	初期値	コメント
DB_GetConnectionStatus_instance	DB_GetConnectionStatus	-	DB_GetConnectionStatus 命令のインスタンスです。
DB_ControlSpool_instance	DB_ControlSpool	-	DB_ControlSpool 命令のインスタンスです。
Trigger	BOOL	FALSE	スプールの再送のトリガとなる変数です。
Operating	BOOL	FALSE	この変数が TRUE の時に、スプール再送が必要であれば実行します。
OperatingEnd	BOOL	FALSE	スプールの再送処理が終了すると、TRUE になります。
RS_instance	RS	-	RS 命令のインスタンスです。
MyStatus	_sDBC_CONNECTION_STATUS	-	DB_GetConnectionStatus 命令のインスタンス DB_GetConnectionStatus_instance の出力 ConnectionStatus に割り付けた変数です。
Resend	BOOL	FALSE	DB コネクションの接続状態が“接続中”の時に、TRUE になります。
Nosent	BOOL	FALSE	DB コネクションの接続状態が“接続中”以外の時に、TRUE になります。
ControlSpool_OK	BOOL	FALSE	DB_ControlSpool 命令が正常終了すると、TRUE になります。

● サンプルプログラム

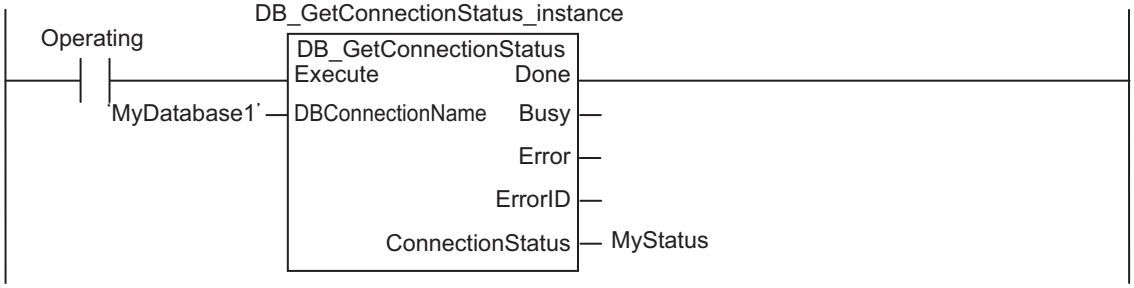
◆DB コネクションの接続状態が“接続中”の場合、スプールに保留された SQL 文を再送します。
実行完了を判断する



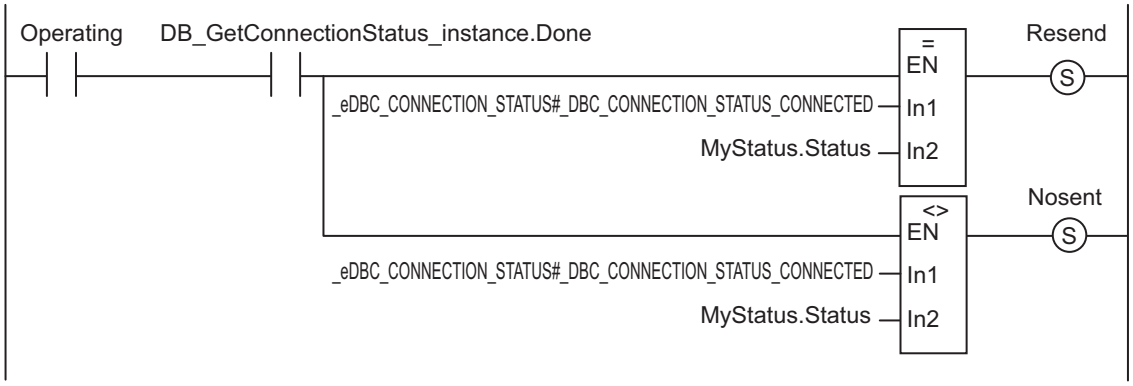
トリガを受け付ける



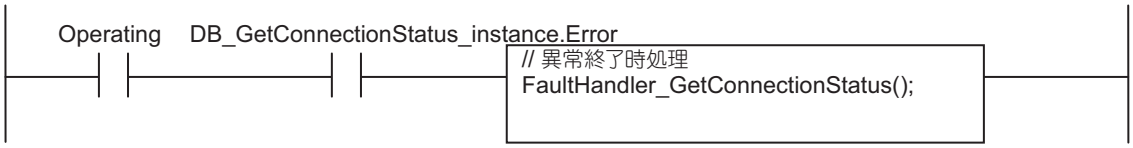
DB コネクションの状態を取得する



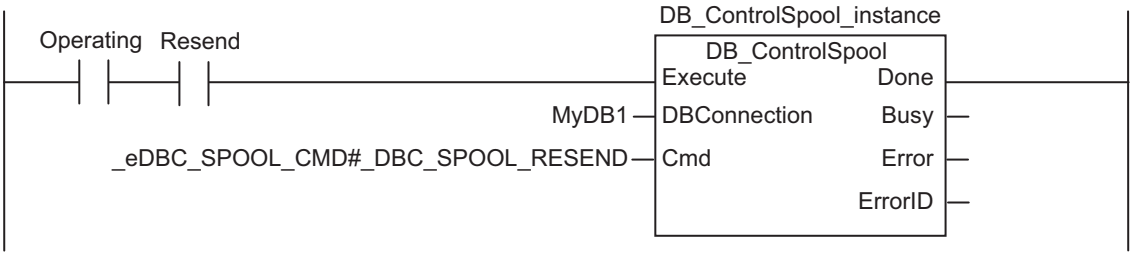
正常終了した場合、接続状態が“接続中”であれば、変数 Resend を TRUE にする



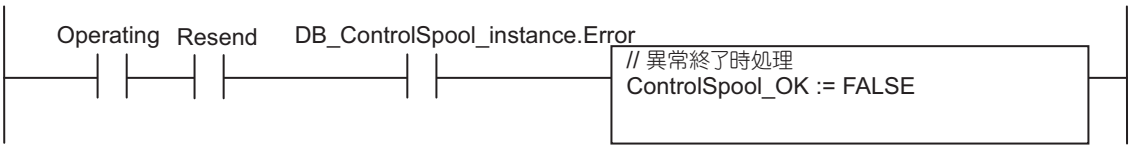
異常終了した場合、装置における異常終了時の手順（FaultHandler_GetConnectionStatus）を実行する
異常終了時の手順については別途、設計してください。



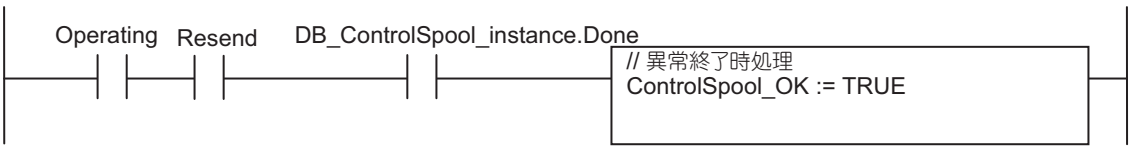
スプールを再送する



異常終了した場合、変数 ControlSpool_OK を FALSE にする



正常終了した場合、変数 ControlSpool_OK を TRUE にする



ストラクチャードテキスト（ST）

● 主な変数

名称	データ型	初期値	コメント
DB_GetConnectionStatus_instance	DB_GetConnectionStatus	-	DB_GetConnectionStatus 命令のインスタンスです。
DB_ControlSpool_instance	DB_ControlSpool	-	DB_ControlSpool 命令のインスタンスです。
Trigger	BOOL	FALSE	スプールの再送のトリガとなる変数です。
LastTrigger	BOOL	FALSE	前回実行時のトリガ状態を保持する変数です。
Operating	BOOL	FALSE	この変数が TRUE の時に、スプール再送が必要であれば実行します。
OperatingStart	BOOL	FALSE	この変数が TRUE の時に、初期処理を実行します。
Resend	BOOL	FALSE	DB コネクションの接続状態が“接続中”の時に、TRUE になります。
MyStatus	_sDBC_CONNECTION_STATUS	-	DB_GetConnectionStatus 命令のインスタンス DB_GetConnectionStatus_instance の出力 ConnectionStatus に割り付けた変数です。
MyDB1	DWORD	-	DB_ControlSpool 命令のインスタンス DB_ControlSpool_instance の入力 DBConnection に割り付けた変数です。

● サンプルプログラム

```
(* ------------------------------------------------------------------------------
   ◆DB コネクションの接続状態が「接続中」の場合、スプールに保留された SQL 文を再送します。
```

```
    ------------------------------------------------------------------------------ *)

// 変数 Trigger の立ち上がりでシーケンス開始
IF ( (Trigger=TRUE) AND (LastTrigger=FALSE) ) THEN
   OperatingStart := TRUE;
   Operating := TRUE;
END_IF;
LastTrigger := Trigger;

// シーケンス開始時処理
IF (OperatingStart=TRUE) THEN
   // 命令インスタンス初期化
   DB_GetConnectionStatus_instance( Execute:=FALSE );
   DB_ControlSpool_instance( Execute:=FALSE );
   OperatingStart := FALSE;
END_IF;

IF (Operating=TRUE) THEN
   // DB コネクションの状態を取得します。
   DB_GetConnectionStatus_instance(
     Execute            := TRUE,
     DBConnectionName   := 'MyDatabase1',
     ConnectionStatus   => MyStatus
   );

   IF (DB_GetConnectionStatus_instance.Done=TRUE) THEN
      // 正常終了時処理
      // DB コネクションの接続状態が「接続中」であれば、変数 Resend を TRUE にします。
      IF (MyStatus.Status = _DBC_CONNECTION_STATUS_CONNECTED) THEN
         Resend := TRUE;
      ELSE
        Resend := FALSE;
        Operating := FALSE;
      END_IF;
   END_IF;
   IF (DB_GetConnectionStatus_instance.Error=TRUE) THEN
      // 異常終了処理
      Operating := FALSE;
   END_IF;
END_IF;

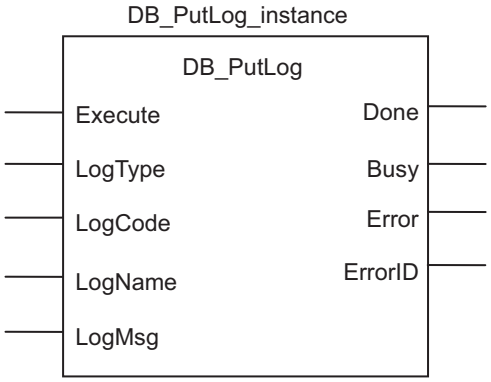
IF ( (Operating=TRUE) AND (Resend=TRUE) ) THEN
   // スプールを再送します。
   DB_ControlSpool_instance(
      Execute      := TRUE,
      DBConnection := MyDB1,
      Cmd          := _DBC_SPOOL_RESEND
   );

   IF (DB_ControlSpool_instance.Done=TRUE) THEN
      // 正常終了時処理
```

```
        Resend := FALSE;
        Operating := FALSE;
    END_IF;
    IF (DB_ControlSpool_instance.Error=TRUE) THEN
        // 異常終了処理
        Resend := FALSE;
        Operating := FALSE;
    END_IF;
END_IF;
```

DB_PutLog（稼働ログ記録）

実行ログまたはデバッグログにユーザ指定のログを記録します。

命令	名称	FB/FUN	グラフィック表現	ST 表現
DB_PutLog	稼働ログ記録	FB		DB_PutLog_instance (Execute, LogType, LogCode, LogName, LogMsg, Done, Busy, Error, ErrorID);

（注）DB_PutLog 命令のインスタンスとして、インスタンス名 DB_PutLog_instance で変数宣言されている場合の例です。

変数

入力変数

入力変数	名称	データ型	有効範囲	単位	初期値	内容
Execute	起動	BOOL	FALSE, TRUE	-	FALSE	実行条件を指定します
LogType	ログの種類	_eDBC_LOGTYPE	_DBC_LOGTYPE_EXECUTION(1)：実行ログ _DBC_LOGTYPE_DEBUG(2)：デバッグログ		_DBC_LOGTYPE_EXECUTION	出力先のログの種類
LogCode	ログコード	INT	0～9999	-	0	ログに記録するコード
LogName	ログ名	STRING	最大 33 バイト（終端 NULL 含む）	-	''	ログに記録する名前
LogMsg	ログメッセージ	STRING	最大 129 バイト（終端 NULL 含む）	-	''	ログに記録するメッセージ

出力変数

出力変数	名称	データ型	有効範囲	単位	内容
Done	完了	BOOL	FALSE, TRUE	-	正常終了を表します
Busy	実行中	BOOL	FALSE, TRUE	-	処理中を表します
Error	エラー	BOOL	FALSE, TRUE		異常終了を表します
ErrorID	エラーコード	WORD	16#0000～16#FFFF	-	エラーコードを表します

関連するエラーコード一覧

エラーコード	名称	内容
0400Hex	入力値範囲外	「LogType」に列挙子にない値を指定した場合
041DHex	命令同時実行リソースオーバ	DB 接続命令を同時に 32 個を超えて実行した場合
1400Hex	SD メモリカード使用不可	SD メモリカードが使用可能状態ではない場合
1401Hex	SD メモリカードライトプロテクト	SD メモリカードがライトプロテクト中の場合
3002Hex	DB 接続サービス終了／終了処理中	DB 接続サービスが終了後／終了処理中に、本命令を実行した場合
3010Hex	ログコード範囲外	「LogCode」の値が有効範囲外の場合
3013Hex	DB 接続サービス異常停止中	DB 接続サービスが異常停止中に、本命令を実行した場合
3015Hex	DB 接続サービス初期処理中	DB 接続サービスが初期処理中に、本命令を実行した場合
3017Hex	稼働ログ無効	指定した稼働ログが無効のためログを記録できない場合
301EHex	DB 接続サービス未起動	DB 接続サービスを［使用しない］に設定した状態で、本命令を実行した場合

機能説明

実行ログまたはデバッグログにユーザ指定のログを記録します。
ログの種類「LogType」で実行ログかデバッグログかを指定します。
ユーザプログラムで指定した任意のログコード「LogCode」、ログメッセージ「LogMsg」を稼働ログに記録することができます。

ログの書式は以下のとおりです。
[連番]<tab>[時刻]<tab>[カテゴリ]<tab>[コード]<tab>[ログ名]<tab>[結果]<tab>[詳細情報]<CR><LF>

[連番]：	0～65535 の連番。65535 を超えたら、0 に戻る
[時刻]：	命令を実行した時刻
[カテゴリ]：	“USER”固定
[コード]：	ログコード「LogCode」で指定した値 NULL 文字（16#00）だけの文字列の場合は何も出力しません。
[ログ名]：	ログ名「LogName」で指定した文字列 NULL 文字（16#00）だけの文字列の場合は何も出力しません。
[結果]：	“0x0000”固定
[詳細情報]：	ログメッセージ「LogMsg」で指定した文字列

使用上の注意

- 本命令は、「Execute」の値が FALSE になっても、また実行時間がタスク周期を超えても、処理を最後まで続けます。処理が正常終了したかどうかは、「Done」の値が TRUE になることで確認してください。
- 「Execute」、「Done」、「Busy」、「Error」のタイミングチャートは、『NJ/NX シリーズコマンドリファレンスマニュアル　基本編』（SBCA-468）の“本章の見方”を参照してください。
- 本命令は、イベントタスク上で使うことができません。コンパイル時にエラーになります。
- SD メモリカード交換中に本命令を実行した場合、以下のように動作します。
  実行ログを指定した場合
  a）CPU ユニットの内部バッファにログを記録し、正常終了します。
  b）SD メモリカードが CPU ユニットに装着された時点で、内部バッファのログを SD メモリカードに保存します。
  デバッグログを指定した場合

a）デバッグログは記録できません。異常終了（稼働ログ無効）します。
- 本命令は、以下の場合、異常となります。「Error」は TRUE となります。
  a）DB 接続サービスが初期処理中に、本命令を実行した場合
  b）DB 接続サービスが異常停止中に、本命令を実行した場合
  c）DB 接続サービスが終了後／終了処理中に、本命令を実行した場合
  d）「LogType」に列挙子にない値を指定した場合
  e）「LogCode」の値が有効範囲外の場合
  f）SD メモリカードが使用可能状態ではない、またはライトプロテクト中の場合
  g）指定した稼働ログが無効のためログを記録できない場合
  h）DB 接続命令を同時に 32 個を超えて実行した場合

サンプルプログラム

ここでは、トリガが TRUE になった時に以下のログを実行ログに記録するサンプルプログラムを掲載します。
- ログコード：100
- ログ名：“生産指示”
- ログメッセージ：“生産開始,RecipeCode=12345678”

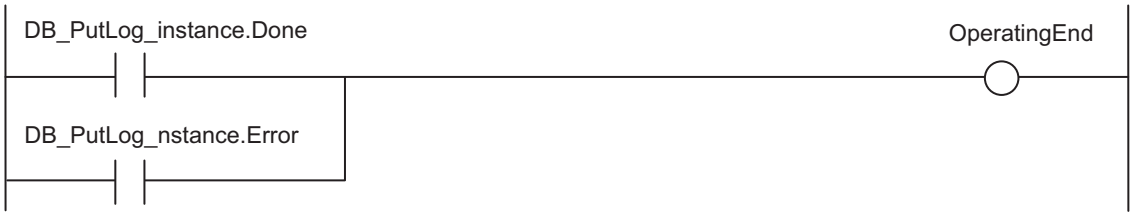
ラダー図

● 主な変数

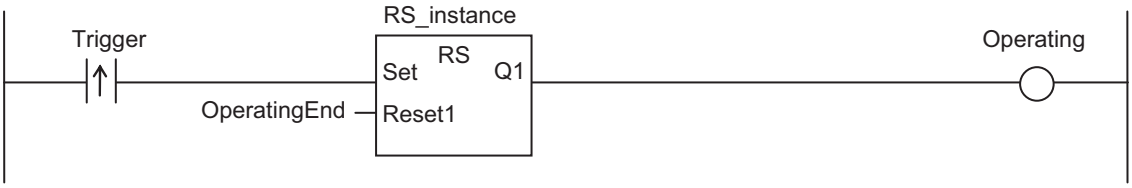
名称	データ型	初期値	コメント
DB_PutLog_instance	DB_PutLog	-	DB_PutLog 命令のインスタンスです。
Trigger	BOOL	FALSE	ユーザ指定ログの記録のトリガとなる変数です。
Operating	BOOL	FALSE	この変数が TRUE の時に、ユーザ指定ログの記録を実行します。
OperatingEnd	BOOL	FALSE	ユーザ指定ログの記録が終了すると、TRUE になります。
RS_instance	RS	-	RS 命令のインスタンスです。
RecipeCode	UDINT	1234678	ログメッセージに使用するレシピコードです。
Msg	STRING[256]	''	ログメッセージです。
PutLog_OK	BOOL	FALSE	DB_PutLog 命令が正常終了すると、TRUE になります。

● サンプルプログラム

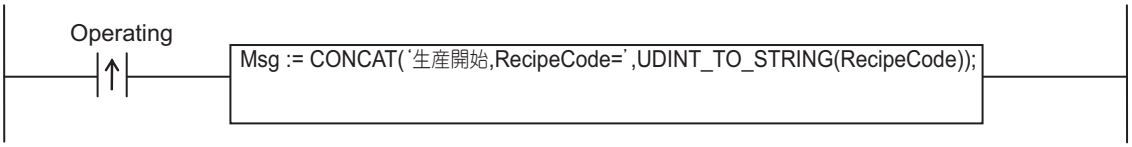
◆実行ログに、ログコード 100、ログ名“生産指示”、ログメッセージ“生産開始,RecipeCode=12345678”を記録します。
DB_PutLog 命令実行完了を判断する



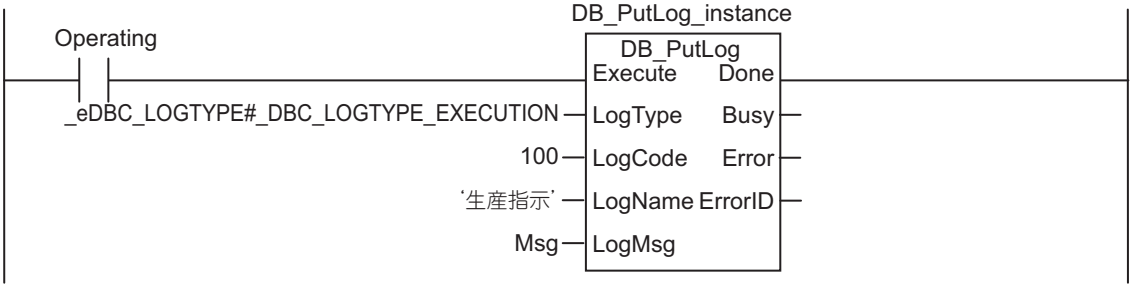
トリガを受け付ける



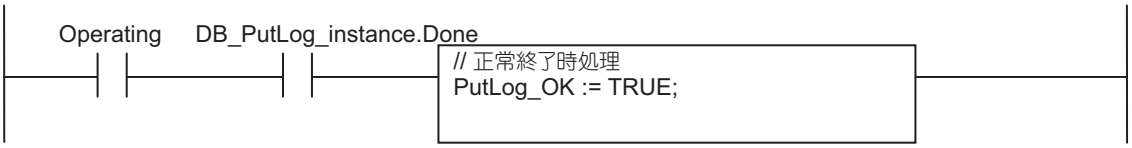
ログメッセージを作成する



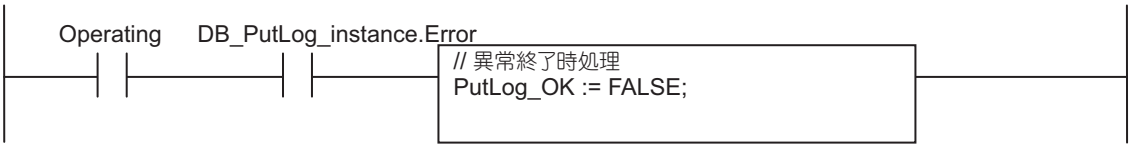
実行ログにログメッセージを記録する



正常終了した場合、変数 PutLog_OK を TRUE にする



異常終了した場合、変数 PutLog_OK を FALSE にする



ストラクチャードテキスト（ST）

● 主な変数

名称	データ型	初期値	コメント
DB_PutLog_instance	DB_PutLog	-	DB_PutLog 命令のインスタンスです。
Trigger	BOOL	FALSE	ユーザ指定ログの記録のトリガとなる変数です。
LastTrigger	BOOL	FALSE	前回実行時のトリガ状態を保持する変数です。
Operating	BOOL	FALSE	この変数が TRUE の時に、ユーザ指定ログの記録を実行します。
OperatingStart	BOOL	FALSE	この変数が TRUE の時に、初期処理を実行します。
RecipeCode	UDINT	1234678	ログメッセージに使用するレシピコードです。
Msg	STRING[256]	''	ログメッセージです。

● サンプルプログラム

```
(* ---------------------------------------------------------------------------------
   ◆実行ログに、ログコード 100、ログ名「生産指示」、ログメッセージ「生産開始,
RecipeCode=12345678」を記録します。
```

```
    ------------------------------------------------------------------------------ *)

// 変数 Trigger の立ち上がりでシーケンス開始
IF ( (Trigger=TRUE) AND (LastTrigger=FALSE) ) THEN
   OperatingStart := TRUE;
   Operating := TRUE;
END_IF;
LastTrigger := Trigger;

// シーケンス開始時処理
IF (OperatingStart=TRUE) THEN
   // 命令インスタンス初期化
   DB_PutLog_instance( Execute:=FALSE );
   // ログメッセージ作成
   Msg := CONCAT('生産開始,RecipeCode=',UDINT_TO_STRING(RecipeCode));

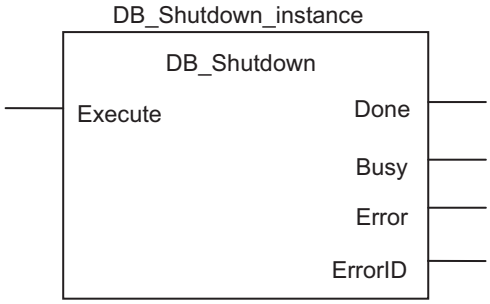
   OperatingStart := FALSE;
END_IF;

IF (Operating=TRUE) THEN
   // 実行ログにログメッセージを記録します。
   DB_PutLog_instance(
     Execute      := TRUE,
     LogType      := _DBC_LOGTYPE_EXECUTION,
     LogCode   := 100,
     LogName   := '生産指示',
     LogMsg       := Msg );

   IF (DB_PutLog_instance.Done=TRUE) THEN
     // 正常終了時処理
      Operating := FALSE;
   END_IF;
   IF (DB_PutLog_instance.Error=TRUE) THEN
     // 異常終了時処理
     Operating := FALSE;
   END_IF;
END_IF;
```

DB_Shutdown（DB 接続サービスのシャットダウン）

DB 接続サービスに対して終了を要求し、DB 接続サービスをシャットダウンします。

命令	名称	FB/FUN	グラフィック表現	ST 表現
DB_Shutdown	DB 接続サービスのシャットダウン	FB		DB_Shutdown_instance (Execute, Done, Busy, Error, ErrorID);

（注）DB_Shutdown 命令のインスタンスとして、インスタンス名 DB_Shutdown_instance で変数宣言されている場合の例です。

変数

入力変数

入力変数	名称	データ型	有効範囲	単位	初期値	内容
Execute	起動	BOOL	FALSE, TRUE	-	FALSE	実行条件を指定します

出力変数

出力変数	名称	データ型	有効範囲	単位	内容
Done	完了	BOOL	FALSE, TRUE	-	正常終了を表します
Busy	実行中	BOOL	FALSE, TRUE	-	処理中を表します
Error	エラー	BOOL	FALSE, TRUE		異常終了を表します
ErrorID	エラーコード	WORD	16#0000 ～ 16#FFFF	-	エラーコードを表します

関連するシステム定義変数

システム定義変数	名称	データ型	有効範囲	内容
_DBC_Status.Run	DB 接続サービス状態動作中フラグ	BOOL	FALSE, TRUE	本命令を実行すると、FALSE となります。
_DBC_Status.Test	DB 接続サービス状態テストモード	BOOL	FALSE, TRUE	本命令を実行すると、FALSE となります。
_DBC_Status.Shutdown	DB 接続サービス状態終了	BOOL	FALSE, TRUE	本命令を実行すると、TRUE となります。

関連するデバイスパラメータ

NX シリーズに接続した X バスユニットの Ethernet ポートを使用する場合は、X バスユニットのデバイスパラメータを参照してください。

関連するエラーコード一覧

エラーコード	名称	内容
041DHex	命令同時実行リソースオーバ	DB 接続命令を同時に 32 個を超えて実行した場合
3001Hex	DB 接続サービス動作モード変更失敗	DB 接続サービスが停止処理中に本命令を実行した場合
3002Hex	DB 接続サービス終了／終了処理中	DB 接続サービスが終了後／終了処理中に、本命令を実行した場合
3015Hex	DB 接続サービス初期処理中	DB 接続サービスが初期処理中に、本命令を実行した場合
301EHex	DB 接続サービス未起動	DB 接続サービスを［使用しない］に設定した状態で、本命令を実行した場合

機能説明

DB 接続機能に対して終了を要求し、DB 接続サービスをシャットダウンします。
稼働ログの内容が消失しないように、CPU ユニットの電源を OFF にする場合は、本命令を実行してください。

使用上の注意

- 本命令は、「Execute」の値が FALSE になっても、また実行時間がタスク周期を超えても、処理を最後まで続けます。処理が正常終了したかどうかは、「Done」の値が TRUE になることで確認してください。
- 「Execute」、「Done」、「Busy」、「Error」のタイミングチャートは、『NJ/NX シリーズコマンドリファレンスマニュアル　基本編』（SBCA-468）の“本章の見方”を参照してください。
- 本命令は、イベントタスク上で使うことができません。コンパイル時にエラーになります。
- 本命令の実行中および実行後は、DB 接続命令は実行できません。DB 接続命令を実行すると異常終了します。
- システムの電源を OFF にする場合は、本命令を実行してください。本命令を実行せずに電源を OFF にした場合は、稼働ログファイルが壊れる、または内容が消失する場合があります。
- 本命令は、以下の場合、異常となります。「Error」は TRUE となります。
  a）DB 接続サービスが初期処理中に、本命令を実行した場合
  b）DB 接続サービスが停止処理中に、本命令を実行した場合
  c）DB 接続サービスが終了後／終了処理中に、本命令を実行した場合
  d）DB 接続命令を同時に 32 個を超えて実行した場合

サンプルプログラム

ここでは、トリガが TRUE になった時に DB 接続サービスをシャットダウンするサンプルプログラムを掲載します。

ラダー図

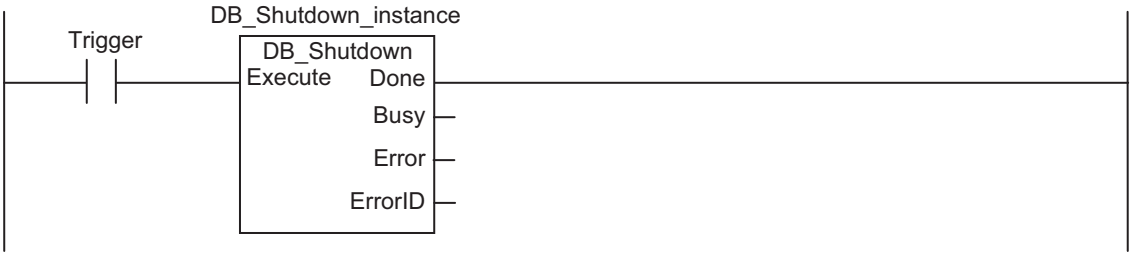
● 主な変数

名称	データ型	初期値	コメント
DB_Shutdown_instance	DB_Shutdown	-	DB_Shutdown 命令のインスタンスです。
Trigger	BOOL	FALSE	DB 接続サービスのシャットダウンのトリガとなる変数です。
Shutdown_OK	BOOL	FALSE	DB_Shutdown 命令が正常終了すると、TRUE になります。

● サンプルプログラム

◆DB 接続サービスをシャットダウンします

DB 接続サービスをシャットダウンする



正常終了した場合、変数 Shutdown_OK を TRUE にする



ストラクチャードテキスト（ST）

● 主な変数

名称	データ型	初期値	コメント
DB_Shutdown_instance	DB_Shutdown	-	DB_Shutdown 命令のインスタンスです。
Trigger	BOOL	FALSE	DB 接続サービスのシャットダウンのトリガとなる変数です。
LastTrigger	BOOL	FALSE	前回実行時のトリガ状態を保持する変数です。
Operating	BOOL	FALSE	この変数が TRUE の時に、DB 接続サービスのシャットダウンを実行します。
OperatingStart	BOOL	FALSE	この変数が TRUE の時に、初期処理を実行します。
ShutdownOK	BOOL	FALSE	DB_Shutdown 命令が正常終了すると、TRUE になります。

● サンプルプログラム

```
(* ------------------------------------------------------------------------------
   ◆DB 接続サービスをシャットダウンします
   ------------------------------------------------------------------------------ *)

// 変数 Trigger の立ち上がりでシーケンス開始
IF ( (Trigger=TRUE) AND (LastTrigger=FALSE) ) THEN
   OperatingStart := TRUE;
   Operating := TRUE;
```

```
END_IF;
LastTrigger := Trigger;

// シーケンス開始時処理
IF (OperatingStart=TRUE) THEN
   // 命令インスタンス初期化
   DB_Shutdown_instance( Execute:=FALSE );

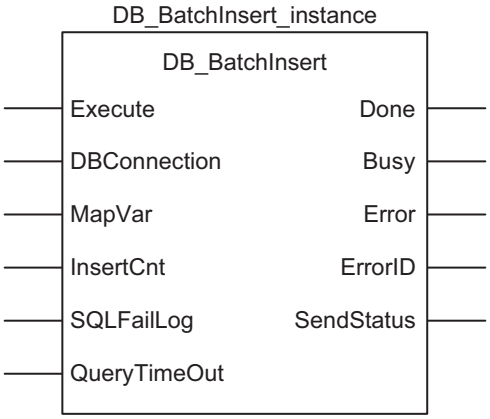
   OperatingStart := FALSE;
END_IF;

IF (Operating=TRUE) THEN
   // DB 接続サービスのシャットダウン
   DB_Shutdown_instance( Execute:=TRUE );

   IF (DB_Shutdown_instance.Done=TRUE) THEN
      // 正常終了時処理
      ShutdownOK := TRUE;
      Operating := FALSE;
   END_IF;
   IF (DB_Shutdown_instance.Error=TRUE) THEN
      // 異常終了時処理
      Operating := FALSE;
   END_IF;
END_IF;
```

DB_BatchInsert（DB レコードー括挿入）

DB のテーブルへ DB マップ変数の各配列要素の値を 1 つのレコードとして一括挿入します。

命令	名称	FB/FUN	グラフィック表現	ST 表現
DB_BatchInsert	DB レコードー括挿入	FB		DB_BatchInsert_instance (Execute, DBConnection, MapVar, InsertCnt, SQLFailLog, QueryTimeOut, Done, Busy, Error, ErrorID, SendStatus);

（注）DB_BatchInsert 命令のインスタンスとして、インスタンス名 DB_BatchInsert_instance で変数宣言されている場合の例です。

変数

入力変数

入力変数	名称	データ型	有効範囲	単位	初期値	内容
Execute	起動	BOOL	FALSE, TRUE	-	FALSE	実行条件を指定します
DBConnection	DB コネクション	DWORD	16#0 ～ FFFFFFFF	-	16#0	DB_Connect で取得した DB コネクション
MapVar	DB マップ変数	構造体型の配列（全体配列）	データ型に従う	-	-	DB_CreateMapping 命令で関連付けた DB マップ変数
InsertCnt	挿入レコード数	DINT	0 ～ 65535	-	0	DB マップ変数「MapVar」の構造体配列の先頭から「InsertCnt」で指定された数のレコードを挿入します
SQLFailLog	SQL 実行失敗ログ出力	BOOL	FALSE, TRUE	-	FALSE	SQL 実行失敗ログに出力するか指定します
QueryTimeOut	クエリ実行タイムアウト時間	TIME	T#0s, T#1s ～ T#600s	-	T#0s	クエリ実行タイムアウト時間を指定します T#0s を指定した場合は、DB コネクション設定の「クエリ実行タイムアウト」で設定した時間を参照します

出力変数

出力変数	名称	データ型	有効範囲	単位	内容
Done	完了	BOOL	FALSE, TRUE	-	正常終了を表します
Busy	実行中	BOOL	FALSE, TRUE	-	処理中を表します
Error	エラー	BOOL	FALSE, TRUE	-	異常終了を表します
ErrorID	エラーコード	WORD	16#0 ～ 16#FFFF	-	エラーコードを表します
SendStatus	送信状態	_eDBC_SEND_STATUS	データ型に従う	-	SQL 文の送信の進捗状況を出力します

関連するシステム定義変数

「DB 接続サービス関連のシステム定義変数 (P.7-5)」を参照してください。

関連するデバイスパラメータ

NX シリーズに接続した X バスユニットの Ethernet ポートを使用する場合は、X バスユニットのデバイスパラメータを参照してください。

関連するエラーコード一覧

エラーコード	名称	内容
0400Hex	入力値範囲外	• 「QueryTimeOut」に有効範囲外の値を指定した場合 • 「InsertCnt」に有効範囲外の値を指定した場合
041DHex	命令同時実行リソースオーバ	DB 接続命令を同時に 32 個以上実行した場合
3000Hex	DB 接続サービス未開始	DB 接続サービスが動作中でないときに、本命令を実行した場合
3002Hex	DB 接続サービス終了／終了処理中	DB 接続サービスが終了／終了処理中に、本命令を実行した場合
3008Hex	DB コネクション不正	「DBConnection」が不正または既に切断している場合
300AHex	DB マップ変数未登録	「MapVar」で指定した変数を DB_CreateMapping でマッピングしていない場合
300BHex	SQL 実行エラー	実行した SQL 文が DB でエラーとなった場合
3011Hex	DB コネクション異常切断発生中	ネットワーク障害などで DB と通信できない場合
3013Hex	DB 接続サービス異常停止中	DB 接続サービスが異常停止中に、本命令を実行した場合
3014Hex	スプールデータあり	既にスプールに保留した SQL 文が存在するため実行できない場合
3015Hex	DB 接続サービス初期処理中	DB 接続サービスが初期処理中に、本命令を実行した場合
3016Hex	DB 処理中	前回実行した DB_Insert／DB_Update／DB_Select／DB_Delete 命令が命令実行タイムアウトし DB 側の処理がまだ完了していない状態で命令を実行した場合
3019Hex	対象外データベース種別への命令実行	本命令の対象となっていないデータベース種別を指定して、命令を実行した場合
301EHex	DB 接続サービス未起動	DB 接続サービスを［使用しない］に設定した状態で、本命令を実行した場合

機能説明

DB_CreateMapping 命令で関連付けたテーブルへ、DB マップ変数「MapVar」の値を一括レコードとして挿入します。
DB_BatchInsert 命令では、スプール機能は使用できません。

DB マップ変数の配列要素数と、挿入レコード数「InsertCnt」の関係について説明します。
- DB マップ変数の配列要素数 ≦ 挿入レコード数または挿入レコード数 = 0 の場合：
  先頭から DB マップ変数の最大要素まで、レコードを一括挿入します。
- DB マップ変数の配列要素数 > 挿入レコード数の場合：
  先頭から「InsertCnt」で指定された数のレコードを挿入します。

SQL 実行失敗ログを有効にする場合は、DB 接続サービス設定で「SQL 失敗ログ」を「記録する」に設定すること、かつ「SQLFailLog」を TRUE にすることで記録可能になります。

DB_BatchInsert 命令では、命令実行タイムアウトは使用できません。入力変数「QueryTimeOut」は、クエリ実行タイムアウトの時間になります。入力変数「QueryTimeOut」に 0 以外の値を指定した場合は、DB コネクション設定の「クエリ実行タイムアウト」で設定した時間ではなく、入力変数「QueryTimeOut」が有効になります。クエリ実行タイムアウトが発生した場合、命令エラー（SQL 実行エラー）が発生します。

使用上の注意

- 本命令は、「Execute」の値が FALSE になっても、また実行時間がタスク周期を超えても、処理を最後まで続けます。処理が正常終了したかどうかは、「Done」の値が TRUE になることで確認してください。
- 「Execute」、「Done」、「Busy」、「Error」のタイミングチャートは、『NJ/NX シリーズコマンドリファレンスマニュアル　基本編』（SBCA-468）の“本章の見方”を参照してください。
- 本命令は、イベントタスク上で使うことができません。コンパイル時にエラーになります。
- 本命令は、SQL 文の不正などに起因し DB へ登録できない場合、異常終了します。
- DB 接続サービスをテストモードで開始している場合、本命令は実際には DB へ INSERT を実行せず正常終了します。
- エラーコードが 300BHex（SQL 実行エラー）の場合は、DB_GetConnectionStatus 命令で、SQL 実行エラーの詳細情報を取得することができます。
- SQL 実行失敗ログを有効にする場合は、他の処理の実行時間が長くなる場合があります。最終的には、必ず実機を使用してあらゆる条件で性能確認を行い、DB 接続命令が適切な実行時間で動作することを確認してください。
- レコードの一括挿入が、設定されたクエリ実行タイムアウトの時間内に完了しないと、命令エラー（SQL 実行エラー）が発生します。「A-2 DB 接続命令の実行時間（P.A-17）」を参考して、DB を配置するサーバの性能およびサーバの負荷変動を加味した時間を設定してください。
- 同一の DB コネクションに対して DB 接続命令を同時に複数実行した場合、DB 接続サービスは命令を 1 個ずつ順番に実行します。2 個目以降に実行される命令の命令実行タイムアウト時間は、「Execute」の値が TRUE になった時点ではなく、その命令を DB 接続サービスが実行した時点から計測を始めます。そのため、「Execute」の値が TRUE になった時点から、その命令がタイムアウトするまでの時間は、命令実行タイムアウト時間で設定した値よりも長くなります。
- 本命令で「Done」の値が TRUE になるのは、SQL サーバが書き込みを完了したことをコントローラが認識し、次のシステムサービスで処理したタイミングです。したがって、本命令と他の DB 接続命令を同一の DB コネクションに対して同時に実行し、かつ本命令が先に実行された場合、本命令の「Done」の値が TRUE になった後に、2 個目の命令が実行されます。
- 同時に実行できる DB 接続命令の最大数は 32 です。命令の「Execute」の値が TRUE になってから、「Done」の値が TRUE になるまでの間、その命令が実行中であると判断されます。

- 命令実行タイムアウト時間の測定誤差の目安は、タスク実行時間の割合が 50%で 100 カラムのレコードの場合、+50ms です。ただし、タスク実行時間の割合およびカラム数によって、測定誤差が変動します。
- 本命令は、以下の場合、異常となります。「Error」は TRUE となります。
  a）DB 接続サービスが動作中でないときに、本命令を実行した場合
  b）DB 接続サービスが初期処理中に、本命令を実行した場合
  c）DB 接続サービスが異常停止中に、本命令を実行した場合
  d）DB 接続サービスが終了後／終了処理中に、本命令を実行した場合
  e）DB 接続サービスが停止処理中に、DB 接続サービスの開始を指示した場合
  f）「DBConnection」が不正、または既に切断している場合
  g）「MapVar」で指定した変数を DB_CreateMapping でマッピングしていない場合
  h）実行した SQL 文が DB でエラーとなった場合
  i）ネットワーク障害などで DB と通信できない場合
  j）既にスプールに保留した SQL 文が存在する場合
  k）指定したクエリ実行タイムアウト時間内に命令が終了しなかった場合
  l）「QueryTimeOut」に有効範囲外の値を指定した場合
  m）前回実行した DB_Insert/DB_Update/DB_Select/DB_Delete 命令が命令実行タイムアウトし、サーバ側 DB の処理がまだ完了していない状態で命令を実行した場合
  n）DB 接続命令を同時に 32 個を超えて実行した場合

サンプルプログラム

ここでは、DB レコード一括挿入を行うサンプルプログラムを掲載します。

構造体型の設定

このサンプルプログラムで使用する構造体の設定を以下に示します。

名称		データ型
PRODUCTION_BATCHINSERT		STRUCT
	NAME	STRING[256]
	LOTNO	STRING[32]
	STATUS	STRING[8]
	PRODUCTIONDATE	DATE

ラダー図

● 主な変数

名称	データ型	初期値	コメント
DB_Connect_instance	DB_Connect	-	DB_Connect 命令のインスタンスです。
DB_CreateMapping_instance	DB_CreateMapping	-	DB_CreateMapping 命令のインスタンスです。
DB_BatchInsert_instance	DB_BatchInsert	-	DB_BatchInsert 命令のインスタンスです。
DB_Close_instance	DB_Close	-	DB_Close 命令のインスタンスです。

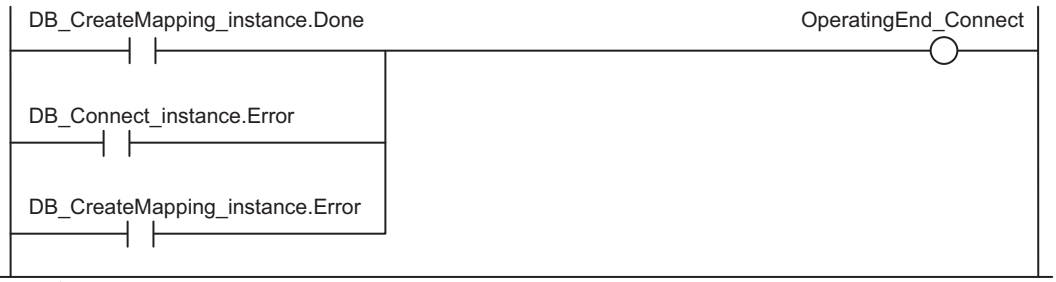
名称	データ型	初期値	コメント
MyDB1	DWORD	-	DB_Connect 命令のインスタンス DB_Connect_instance の出力 DBConnection に割り付けた変数です。
Name	STRING[256]	'WORK001'	生産情報：製品名
LotNo	UINT	1234	生産情報：ロット番号
Trigger_Connect	BOOL	-	DB コネクション接続のトリガとなる変数です。
Operating_Connect	BOOL	-	この変数が TRUE のときに、DB_Connect 命令を実行します。
OperatingEnd_Connect	BOOL	-	DB_Connect 命令が終了すると、TRUE になります。
RS_Connect_instance	RS	-	RS 命令のインスタンスです。
MapVar_BatchInsert	ARRAY[0..99] OF PRODUCTION_BATCHINSERT		DB_CreateMapping 命令のインスタンス DB_CreateMapping_instance の入力 MapVar に割り付けた変数です。
Trigger_BatchInsert	BOOL	-	DB レコード一括挿入のトリガとなる変数です。
Operating_BatchInsert	BOOL	-	この変数が TRUE で、Execute_BatchInsert が TRUE のときに、DB_BatchInsert 命令を実行します。
OperatingEnd_BatchInsert	BOOL	-	DB_BatchInsert 命令が終了すると、TRUE になります。
RS_BatchInsert_instance	RS	-	RS 命令のインスタンスです。
Trigger_Close	BOOL	-	DB コネクション切断のトリガとなる変数です。
Operating_Close	BOOL	-	この変数が TRUE のときに、DB_Close 命令を実行します。
OperatingEnd_Close	BOOL	-	DB_Close 命令が終了すると、TRUE になります。
RS_Close_instance	RS	-	RS 命令のインスタンスです。
Index	UINT	-	レコード番号を表す変数です。
TON_instance	TON	-	TON 命令のインスタンスです。
Execute_BatchInsert	BOOL	-	この変数が TRUE の時に、DB_BatchInsert 命令を実行します。
RS_ExecuteBatchInsert_instance	RS	-	RS 命令のインスタンスです。

● サンプルプログラム

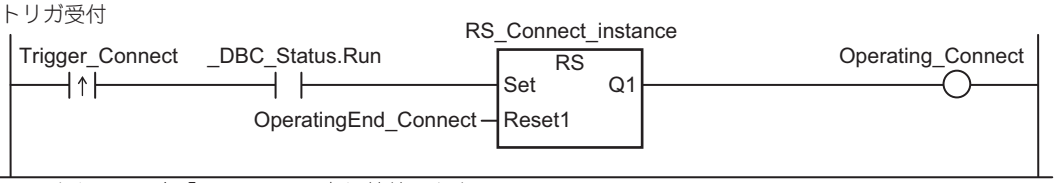
0　◆DBに生産データを格納します。
処理の手順は以下のとおりです。
1. DB_Connect命令で、DBに接続します。
2. DB_CreateMapping命令で、DBのテーブルと変数を関連付けします。
3-1. DBへ格納するデータを準備します。
3-2. DB_BatchInsert命令で、DBのテーブルへデータを格納します。
4. DB_Close命令で、DBとのDBコネクションを切断します。

◆DBコネクション名「MyDatabese1」に接続し、DBのテーブルと変数を関連付けます。

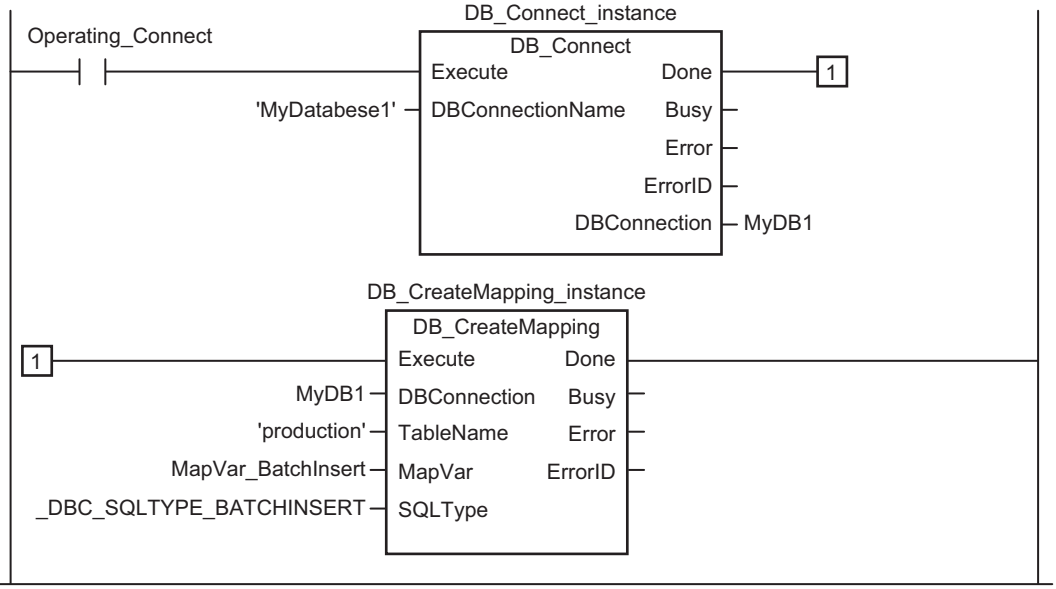
DB_Connect命令、DB_CreateMapping実行完了の判断



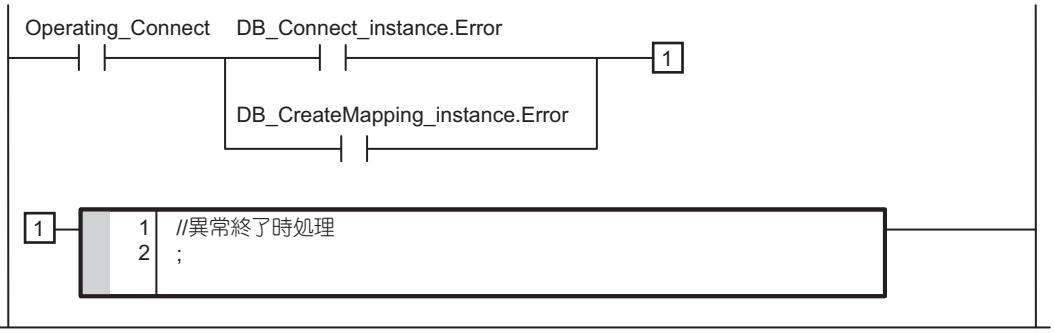
1　トリガ受付



2　DBコネクション名「MyDatabase1」に接続します。
DBコネクション「MyDB1」の「Production」テーブルに、変数「MapVar_BatchInsert」を
BATCHINSERT操作用に関連付けます。



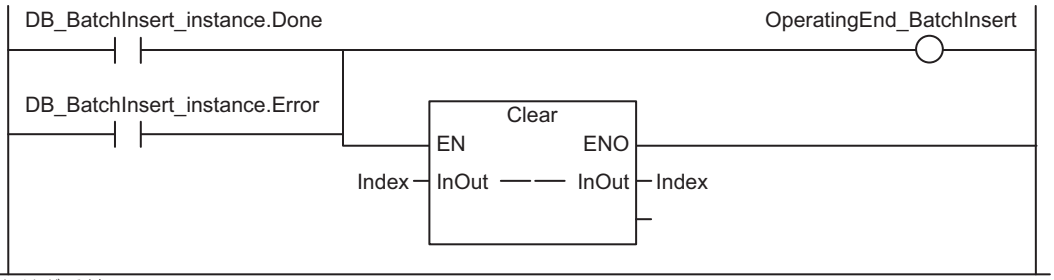
3 異常終了時の処理



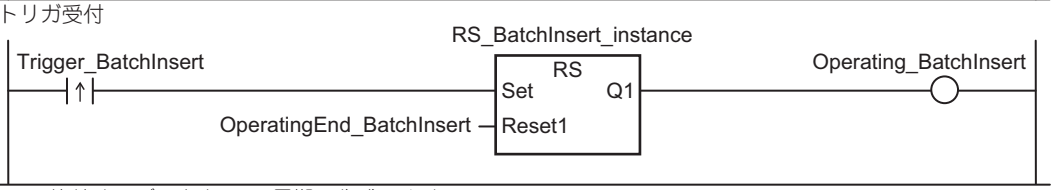
4 ----------

◆DBコネクション「MyDB1」にデータを格納します。

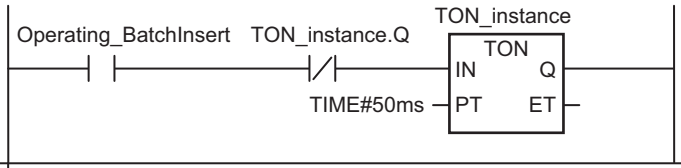
DB_BatchInsert命令実行完了の判断



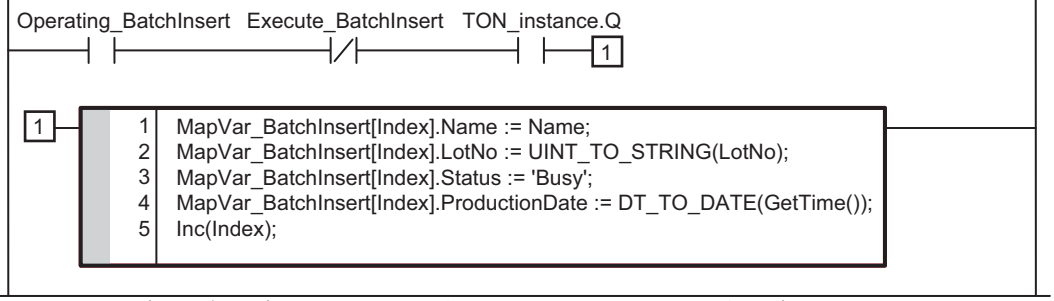
5 トリガ受付



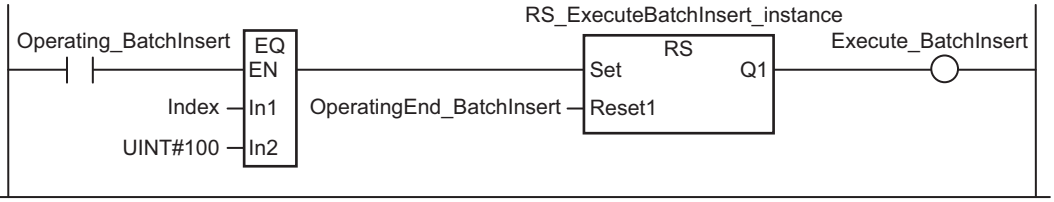
6 DBへ格納するデータを50ms周期で生成します。

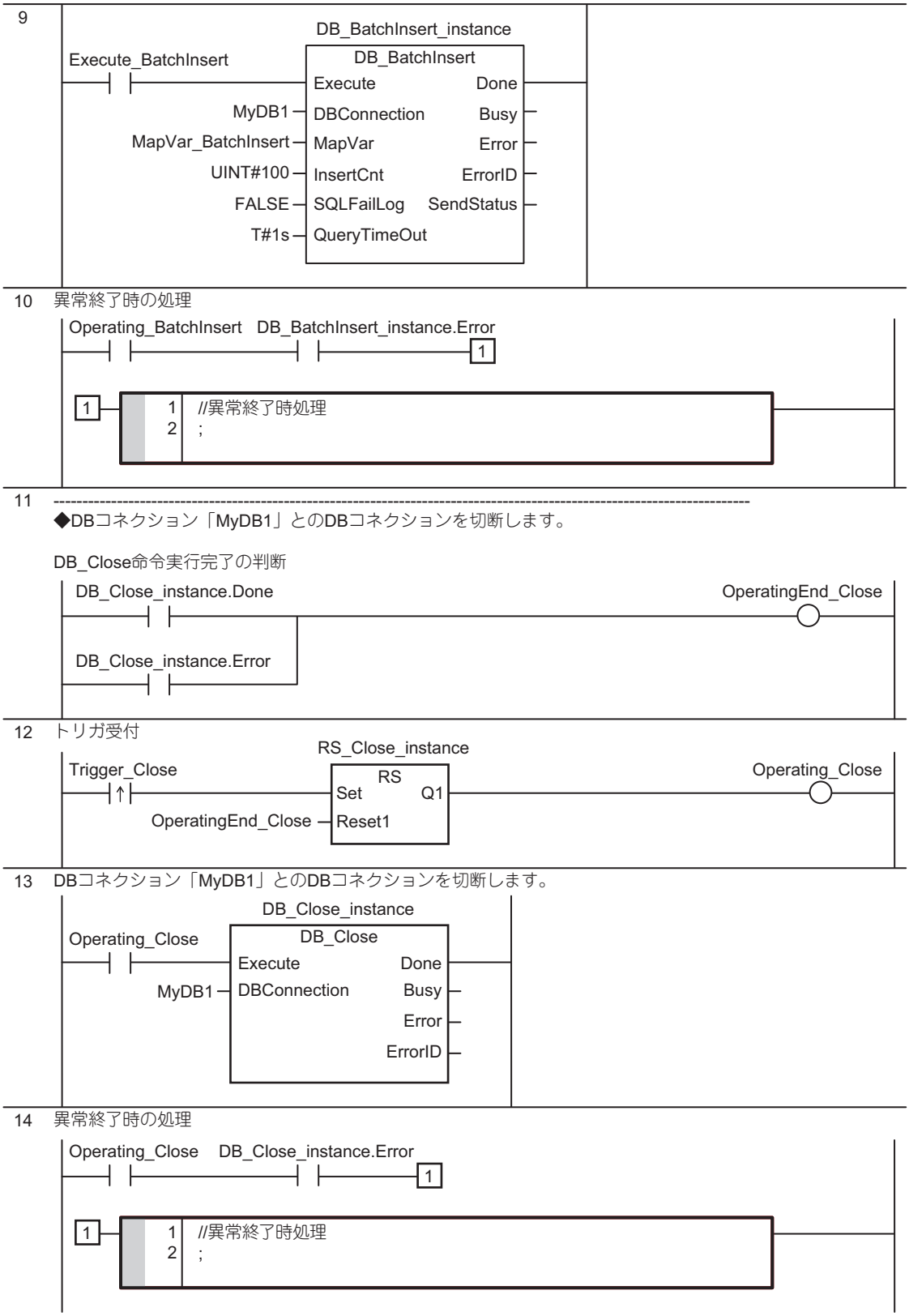


7



8 DBへ格納するデータが100データ蓄積された場合、DBコネクション「MyDB1」にデータを格納します。




9
DB_BatchInsert_instance
DB_BatchInsert
Execute_BatchInsert
Execute
Done
MyDB1
DBConnection
Busy
MapVar_BatchInsert
MapVar
Error
UINT#100
InsertCnt
ErrorID
FALSE
SQLFailLog
SendStatus
T#1s
QueryTimeOut
10 異常終了時の処理
Operating_BatchInsert
DB_BatchInsert_instance.Error
1
1
1 //異常終了時処理
2 ;
11
◆DBコネクション「MyDB1」とのDBコネクションを切断します。
DB_Close命令実行完了の判断
DB_Close_instance.Done
OperatingEnd_Close
DB_Close_instance.Error
12 トリガ受付
RS_Close_instance
Trigger_Close
RS
Set
Q1
Operating_Close
OperatingEnd_Close
Reset1
13 DBコネクション「MyDB1」とのDBコネクションを切断します。
DB_Close_instance
DB_Close
Operating_Close
Execute
Done
MyDB1
DBConnection
Busy
Error
ErrorID
14 異常終了時の処理
Operating_Close
DB_Close_instance.Error
1
1
1 //異常終了時処理
2 ;

ストラクチャードテキスト（ST）

● 主な変数

名称	データ型	初期値	コメント
DB_Connect_instance	DB_Connect	-	DB_Connect 命令のインスタンスです。
DB_CreateMapping_instance	DB_CreateMapping	-	DB_CreateMapping 命令のインスタンスです。
DB_BatchInsert_instance	DB_BatchInsert	-	DB_BatchInsert 命令のインスタンスです。
DB_Close_instance	DB_Close	-	DB_Close 命令のインスタンスです。
MyDB1	DWORD	-	DB_Connect 命令のインスタンス DB_Connect_instance の出力 DBConnection に割り付けた変数です。
Name	STRING[256]	'WORK001'	生産情報：製品名
LotNo	UINT	1234	生産情報：ロット番号
Trigger_Connect	BOOL	-	DB コネクション接続のトリガとなる変数です。
LastTrigger_Connect	BOOL	-	前回実行時のトリガ状態を保持する変数です。
Operating_Connect	BOOL	-	この変数が TRUE のときに、DB_Connect 命令を実行します。
OperatingStart_Connect	BOOL	-	この変数が TRUE のときに、DB コネクション接続開始時の処理を実行します。
MapVar_BatchInsert	ARRAY[0..99] OF PRODUCTION_BATCHINSERT		DB_CreateMapping 命令のインスタンス DB_CreateMapping_instance の入力 MapVar に割り付けた変数です。
Trigger_BatchInsert	BOOL	-	DB レコード一括挿入のトリガとなる変数です。
LastTrigger_BatchInsert	BOOL	-	前回実行時のトリガ状態を保持する変数です。
Operating_BatchInsert	BOOL	-	この変数が TRUE で、Execute_BatchInsert が TRUE のときに、DB_BatchInsert 命令を実行します。
Execute_BatchInsert	BOOL	-	この変数が TRUE のときに、DB_BatchInsert 命令を実行します。
Trigger_Close	BOOL	-	DB コネクション切断のトリガとなる変数です。
LastTrigger_Close	BOOL	-	前回実行時のトリガ状態を保持する変数です。
Operating_Close	BOOL	-	この変数が TRUE のときに、DB_Close 命令を実行します。
OperatingStart_Close	BOOL	-	この変数が TRUE のときに、DB コネクション切断開始時の処理を実行します。

名称	データ型	初期値	コメント
Stage	INT	-	DB コネクション接続の状態を表す変数です。
Index	UINT	-	レコード番号を表す変数です。
TON_instance	TON	-	TON 命令のインスタンスです。

● サンプルプログラム

```
(* ------------------------------------------------------------------------------
   ◆DB に生産データを格納します。
   処理の手順は以下のとおりです。
   1．DB_Connect 命令で、DB に接続します。
   2．DB_CreateMapping 命令で、DB のテーブルと変数を関連付けします。
   3-1. DB へ格納するデータを準備します。
   3-2. DB_BatchInsert 命令で、DB のテーブルへデータを格納します。
   4．DB_Close 命令で、DB との DB コネクションを切断します。
    ------------------------------------------------------------------------------ *)

//------------------------------------------------------------------------------
//◆DB コネクション名「MyDatabese1」に接続し、DB のテーブルと変数を関連付けます

//Trigger_Connect の立上りでシーケンス開始
IF ( (Trigger_Connect=TRUE) AND (LastTrigger_Connect=FALSE) AND (_DBC_Status.Run=TRUE) ) THEN
        OperatingStart_Connect := TRUE;
        Operating_Connect := TRUE;
END_IF;
LastTrigger_Connect:=Trigger_Connect;

//シーケンス開始時処理
IF (OperatingStart_Connect=TRUE) THEN
        //DB 接続命令インスタンス初期化
        DB_Connect_instance(Execute:=FALSE );
        DB_CreateMapping_instance(Execute := FALSE, MapVar:=MapVar_BatchInsert, SQLType:=_DBC_SQLTYP
E_BATCHINSERT);
        Stage := INT#1;
        Index:=UINT#0;
        OperatingStart_Connect  := FALSE;
END_IF;


//DB コネクション名「MyDatabese1」に接続します。
//DB コネクション「MyDB1」の「Production」テーブルに、変数「MapVar_BatchInsert」を BATCHINSERT 操作用に
関連付けます。
IF (Operating_Connect=TRUE) THEN
        CASE Stage OF
        1 : // DB コネクション接続
                DB_Connect_instance( Execute:=TRUE, DBConnectionName:='MyDatabase1', DBConnect
ion=>MyDB1 );

                IF (DB_Connect_instance.Done=TRUE) THEN
                        Stage := INT#2; // 正常終了
```

```
                    ELSIF (DB_Connect_instance.Error=TRUE) THEN
                            Stage := INT#99; // 異常終了
                    END_IF;

        2 : // DB のテーブルと変数を関連付け
                    DB_CreateMapping_instance( Execute := TRUE, DBConnection:=MyDB1, TableName:='P
roduction', MapVar:=MapVar_BatchInsert, SQLType:=_DBC_SQLTYPE_BATCHINSERT);
                    IF ( DB_CreateMapping_instance.Done=TRUE) THEN
                            Operating_Connect:=FALSE; // 正常終了
                    ELSIF ( DB_CreateMapping_instance.Error=TRUE ) THEN
                            Stage := INT#99; // 異常終了
                    END_IF;

        99 :
                    // 異常終了処理
                    Operating_Connect := FALSE;
        END_CASE;
END_IF;

//-----------------------------------------------------------------------------
//◆DB コネクション「MyDB1」にデータを格納します。

//Trigger_BatchInsert の立上りでシーケンス開始
IF ( (Trigger_BatchInsert=TRUE) AND (LastTrigger_BatchInsert=FALSE) ) THEN
        Operating_BatchInsert := TRUE;
        //DB 接続命令インスタンス初期化
        DB_BatchInsert_instance( Execute:=FALSE, MapVar:=MapVar_BatchInsert );
END_IF;
LastTrigger_BatchInsert := Trigger_BatchInsert;

//DB へ格納するデータを 50ms 周期で生成します。
TON_instance( In:=NOT(TON_instance.Q), PT:=TIME#50ms );

IF( (Operating_BatchInsert=TRUE) AND (Execute_BatchInsert=FALSE) AND (TON_instance.Q=TRUE))THEN
        MapVar_BatchInsert[Index].Name := Name;
        MapVar_BatchInsert[Index].LotNo := UINT_TO_STRING(LotNo);
        MapVar_BatchInsert[Index].Status := 'Busy';
        MapVar_BatchInsert[Index].ProductionDate := DT_TO_DATE(GetTime( ));
        Inc(Index);
END_IF;

//DB へ格納するデータが 100 データ蓄積された場合、DB コネクション「MyDB1」に生産データを格納します。
IF( (Operating_BatchInsert=TRUE) AND (Index=UINT#100) )THEN
        Execute_BatchInsert:=TRUE;
        Index:=0;
END_IF;

IF( (Operating_BatchInsert=TRUE) AND (Execute_BatchInsert=TRUE))THEN
        DB_BatchInsert_instance(Execute:=TRUE, DBConnection:=MyDB1, MapVar:=MapVar_BatchInsert, Inse
rtCnt:=UINT#100, SQLFailLog:=FALSE, QueryTimeOut:=T#1s);
        IF (DB_BatchInsert_instance.Done=TRUE) THEN
```

```
                Execute_BatchInsert:=FALSE; // 正常終了
                Operating_BatchInsert := FALSE;
        ELSIF (DB_BatchInsert_instance.Error=TRUE) THEN
                // 異常終了処理
                Execute_BatchInsert:=FALSE;
                Operating_BatchInsert := FALSE;
        END_IF;
END_IF;

//-----------------------------------------------------------------------------------
//◆DB コネクション「MyDB1」との DB コネクションを切断します。

//Trigger_Close の立上りでシーケンス開始
IF ( (Trigger_Close=TRUE) AND (LastTrigger_Close=FALSE) ) THEN
        OperatingStart_Close := TRUE;
        Operating_Close := TRUE;
END_IF;
LastTrigger_Close := Trigger_Close;

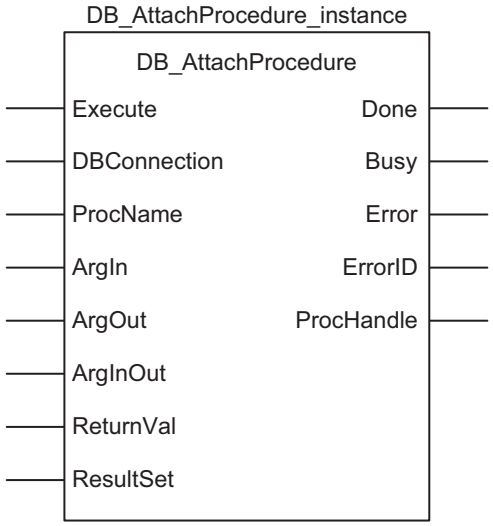
//シーケンス開始時処理
IF (OperatingStart_Close=TRUE) THEN
        //DB 接続命令インスタンス初期化
        DB_Close_instance(Execute:=FALSE );
        OperatingStart_Close := FALSE;
END_IF;

//DB コネクション「MyDB1」との DB コネクションを切断します。
IF (Operating_Close=TRUE) THEN
        //DB コネクション切断
        DB_Close_instance(Execute:=TRUE, DBConnection:=MyDB1 );

        IF (DB_Close_instance.Done=TRUE) THEN
                Operating_Close := FALSE; // 正常終了
        ELSIF (DB_Close_instance.Error=TRUE) THEN
                //異常終了処理
                Operating_Close := FALSE;
        END_IF;
END_IF;
```

DB_AttachProcedure（DB ストアドプロシージャ ハンドル生成）

DB のストアドプロシージャ呼び出すためのプロシージャハンドルの取得を行います。

命令	名称	FB/FUN	グラフィック表現	ST 表現
DB_AttachProcedure	DB ストアドプロシージャハンドル生成	FB		DB_AttachProcedure_instance (Execute, DBConnection, ProcName, ArgIn, ArgOut, ArgInOut, ReturnVal, ResultSet, Done, Busy, Error, ErrorID, ProcHandle);

（注）DB_AttachProcedure 命令のインスタンスとして、インスタンス名 DB_AttachProcedure_instance で変数宣言されている場合の例です。

変数

入力変数

入力変数	名称	データ型	有効範囲	単位	初期値	内容
Execute	起動	BOOL	FALSE, TRUE	-	FALSE	実行条件を指定します
DBConnection	DB コネクション	DWORD	16#0 ～ FFFFFFFF	-	16#0	DB_Connect で取得した DB コネクション
ProcName	ストアドプロシージャ名	STRING	データ型に従う	-	''	DB のプロシージャ、または、ファンクション名
ArgIn	ストアドプロシージャの IN 引数	構造体型	データ型に従う	-	-	ストアドプロシージャの引数に関連付ける変数(IN)
ArgOut	ストアドプロシージャの OUT 引数	構造体型	データ型に従う	-	-	ストアドプロシージャの引数に関連付ける変数(OUT)
ArgInOut	ストアドプロシージャの INOUT 引数	構造体型	データ型に従う	-	-	ストアドプロシージャの引数に関連付ける変数(INOUT)
ReturnVal	ストアドプロシージャの戻り値	基本型（構造体、共用体、列挙体除く）	データ型に従う	-	-	ストアドプロシージャの戻り値に関連付ける変数

入力変数	名称	データ型	有効範囲	単位	初期値	内容
ResultSet	ストアドプロシージャの結果セット	構造体型、構造体型の配列（全体配列）	データ型に従う	-	-	ストアドプロシージャの結果セットに関連付ける変数

出力変数

出力変数	名称	データ型	有効範囲	単位	内容
Done	完了	BOOL	FALSE, TRUE	-	正常終了を表します
Busy	実行中	BOOL	FALSE, TRUE	-	処理中を表します
Error	エラー	BOOL	FALSE, TRUE	-	異常終了を表します
ErrorID	エラーコード	WORD	16#0 ～ 16#FFFF	-	エラーコードを表します
ProcHandle	プロシージャハンドル	DWORD	16#0 ～ FFFFFFFF	-	DB 接続命令からストアドプロシージャを呼び出すときのハンドル

関連するシステム定義変数

「DB 接続サービス関連のシステム定義変数 (P.7-5)」を参照してください。

関連するデバイスパラメータ

NX シリーズに接続した X バスユニットの Ethernet ポートを使用する場合は、X バスユニットのデバイスパラメータを参照してください。

関連するエラーコード一覧

エラーコード	名称	内容
0406Hex	領域範囲外指定	「ProcName」が NULL 文字（16#00）だけの文字列である場合
0410Hex	文字列フォーマット異常	「ProcName」にスペース文字を含む文字列を指定した場合 「ProcName」が終端 NULL で終わらない場合
041BHex	データ容量超過	マッピング可能な DB マップ変数の最大数を超えるプロシージャハンドルを取得した場合
041DHex	命令同時実行リソースオーバ	DB 接続命令を同時に 32 個を超えて実行した場合
3000Hex	DB 接続サービス未開始	DB 接続サービスが動作中でないときに、本命令を実行した場合
3002Hex	DB 接続サービス終了／終了処理中	DB 接続サービスが終了後／終了処理中に、本命令を実行した場合
3008Hex	DB コネクション不正	「DBConnection」が不正、または既に切断している場合
3009Hex	DB マップ変数不正	「ArgIn」「ArgOut」「ArgInOut」「ResultSet」で指定した変数のデータ型が構造体でない場合 「ArgIn」「ArgOut」「ArgInOut」「ResultSet」で指定した変数の構造体のメンバに派生データ型が含まれている場合 構造体型の配列変数を「ArgIn」「ArgOut」「ArgInOut」に指定した場合 基本型以外を「ReturnVal」に指定した場合
3011Hex	DB コネクション異常切断発生中	ネットワーク障害などで DB と通信できない場合
3013Hex	DB 接続サービス異常停止中	DB 接続サービスが異常停止中に、本命令を実行した場合
3015Hex	DB 接続サービス初期処理中	DB 接続サービスが初期処理中に、本命令を実行した場合

エラーコード	名称	内容
3019Hex	対象外データベース種別への命令実行	本命令の対象となっていないデータベース種別を指定して、命令を実行した場合
301AHex	ストアドプロシージャ名不正	指定したストアドプロシージャ名が存在しない場合 （注）指定したストアドプロシージャ名が DB 上に見つからない場合も含みます。ストアドプロシージャが存在する場合でも、ストアドプロシージャへのアクセス権限がないなどの理由により、DB 接続サービスがストアドプロシージャ名を見つけられないことがあります
301BHex	ストアドプロシージャ引数不正	ストアドプロシージャの引数とアタッチした引数の情報が一致しない場合
301EHex	DB 接続サービス未起動	DB 接続サービスを［使用しない］に設定した状態で、本命令を実行した場合

機能説明

DB のストアドプロシージャ呼び出すためのプロシージャハンドルの取得を行います。
取得したプロシージャハンドル「ProcHandle」はマッピング可能な DB マップ変数の最大数の制約を受けます。たとえば、数の最大が 15 個[*1] の場合に、DB マップ変数「MapVar」を 14 個使用しているとすると DB_AttachProcedure 命令で取得できるプロシージャハンドル「ProcHandle」は 1 個までとなります。

*1. 機種ごとのマッピング可能な DB マップ変数の最大数は、「1-2-1 DB 接続サービスの仕様 (P.1-4)」を参照してください。

呼び出したいストアドプロシージャの名前を「ProcName」に指定します。
そのストアドプロシージャの引数・戻り値・結果セットをそれぞれ「ArgIn」「ArgOut」「ArgInOut」「ReturnVal」「ResultSet」に変数で指定します。
「ProcName」で指定したストアドプロシージャと「ArgIn」「ArgOut」「ArgInOut」「ReturnVal」「ResultSet」で指定した DB マップ変数を関連付けします。さらに、DB コネクションとも関連付けします。

「ProcName」で指定したストアドプロシージャのメタ情報を取得して、ストアドプロシージャのインタフェース（引数・戻り値・結果セット）の確認を行います。

稼働ログを使用すると、実行時のエラーについて確認できます。

【ストアドプロシージャ引数】
プロシージャの引数と、「ArgIn/ArgOut/ArgInOut」で指定した名称、データ型、および引数の数は一致させてください。
プロシージャの引数属性に、INOUT/OUT が存在する場合、DB_ExecuteProcedure 命令を実行後に、「ArgOut/ArgInOut」で指定した構造体変数に値が設定されます。
ストアドプロシージャの引数を省略する場合は、入力変数「ArgIn/ArgOut/ArgInOut」それぞれにシステム定義変数「_DBC_Unused」を代入してください。[*1]
DB ごとのストアドプロシージャ呼出機能の仕様については、「5-3-2 DB ごとのストアドプロシージャ呼出機能の仕様 (P.5-16)」を参照してください。

【ストアドプロシージャ戻り値】
プロシージャの戻り値と、「ReturnVal」で指定した名称、データ型は一致させてください。
ストアドプロシージャの戻り値を省略する場合は、入力変数「ReturnVal」にシステム定義変数「_DBC_Unused」を代入してください。[*1]

【ストアドプロシージャ結果セット】

プロシージャの結果セットと、「ResultSet」で指定した名称、データ型、および引数の数は一致させてください。
複数のレコードを取得する場合は、配列として定義します。
ストアドプロシージャの結果セットを省略する場合は、入力変数「ResultSet」にシステム定義変数「_DBC_Unused」を代入してください。*1
DB ごとのストアドプロシージャ呼出機能の仕様については、「5-3-2 DB ごとのストアドプロシージャ呼出機能の仕様 (P.5-16)」を参照してください。

*1.「_DBC_Unused」の値が TRUE、FALSE のどちらでも、命令の実行結果は同じです。

使用上の注意

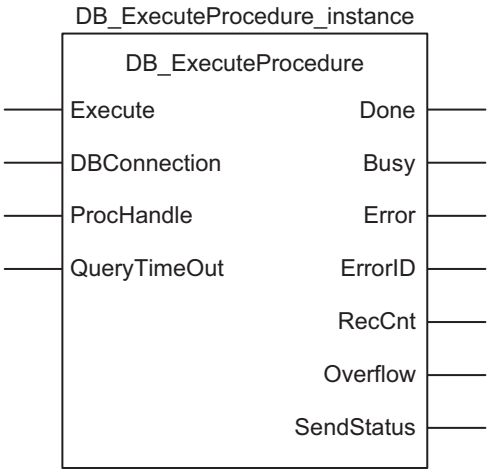
- 本命令は、「Execute」の値が FALSE になっても、また実行時間がタスク周期を超えても、処理を最後まで続けます。処理が正常終了したかどうかは、「Done」の値が TRUE になることで確認してください。
- 「Execute」、「Done」、「Busy」、「Error」のタイミングチャートは、『NJ/NX シリーズコマンドリファレンスマニュアル　基本編』（SBCA-468）の“本章の見方”を参照してください。
- 本命令は、イベントタスク上で使うことができません。コンパイル時にエラーになります。
- DB 接続サービスをテストモードで開始している場合、本命令は正常終了します。
- 本命令は、以下の場合、異常となります。「Error」は TRUE となります。
  a）DB 接続サービスが動作中でないときに、本命令を実行した場合
  b）DB 接続サービスが初期処理中に、本命令を実行した場合
  c）DB 接続サービスが異常停止中に、本命令を実行した場合
  d）DB 接続サービスが終了後／終了処理中に、本命令を実行した場合
  e）「DBConnection」が不正、または既に切断している場合
  f）「ProcName」が NULL 文字（16#00）だけの文字列である場合
  g）「ProcName」にスペース文字を含む文字列を指定した場合
  「ProcName」が終端 NULL で終わらない場合
  h）マッピング可能な DB マップ変数の最大数を超える DB マップ変数を関連付けようとした場合
  マッピング可能な DB マップ変数の最大数を超えるプロシージャハンドルを取得した場合
  i）「ArgIn」「ArgOut」「ArgInOut」「ResultSet」で指定した変数のデータ型が構造体でない場合
  「ArgIn」「ArgOut」「ArgInOut」「ResultSet」で指定した変数の構造体のメンバに派生データ型が含まれている場合
  構造体型の配列変数を「ArgIn」「ArgOut」「ArgInOut」に指定した場合
  基本型以外を「ReturnVal」に指定した場合
  j）指定したストアドプロシージャ名が存在しない場合
  k）ストアドプロシージャの引数とアタッチした引数の情報が一致しない場合
  l）ネットワーク障害などで DB と通信できない場合
  m）本命令の対象となっていないデータベース種別を指定して、命令を実行した場合
  n）DB 接続命令を同時に 32 個を超えて実行した場合

サンプルプログラム

サンプルプログラムは、DB_ExecuteProcedure 命令のサンプルプログラム「サンプルプログラム (P.7-106)」を参照してください。

DB_ExecuteProcedure（DB ストアドプロシージャ 実行）

DB_AttachProcedure で取得したプロシージャハンドルを使って、ストアドプロシージャを呼び出します。

命令	名称	FB/FUN	グラフィック表現	ST 表現
DB_ExecuteProcedure	DB ストアドプロシージャ実行	FB		DB_ExecuteProcedure_instance (Execute, DBConnection, ProcHandle, QueryTimeOut, Done, Busy, Error, ErrorID, RecCnt, Overflow, SendStatus);

（注）DB_ExecuteProcedure 命令のインスタンスとして、インスタンス名 DB_ExecuteProcedure_instance で変数宣言されている場合の例です。

変数

入力変数

入力変数	名称	データ型	有効範囲	単位	初期値	内容
Execute	起動	BOOL	FALSE, TRUE	-	FALSE	実行条件を指定します
DBConnection	DB コネクション	DWORD	16#0 ～ FFFFFFFF	-	16#0	DB_Connect で取得した DB コネクション
ProcHandle	プロシージャハンドル	DWORD	16#0 ～ FFFFFFFF	-	16#0	DB_AttachProcedure 命令で取得したプロシージャハンドル
QueryTimeOut	クエリ実行タイムアウト時間	TIME	T#0s, T#1s ～ T#600s	-	T#0s	クエリ実行タイムアウト時間を指定します。T#0s を指定した場合は、DB コネクション設定の「クエリ実行タイムアウト」で設定した時間を参照します。

出力変数

出力変数	名称	データ型	有効範囲	単位	内容
Done	完了	BOOL	FALSE, TRUE	-	正常終了を表します

出力変数	名称	データ型	有効範囲	単位	内容
Busy	実行中	BOOL	FALSE, TRUE	-	処理中を表します
Error	エラー	BOOL	FALSE, TRUE	-	異常終了を表します
ErrorID	エラーコード	WORD	16#0 ～ 16#FFFF	-	エラーコードを表します
RecCnt	レコード数	DINT	0 ～ 65535	-	結果セット変数「ResultSet」に格納したレコード数を表します
Overflow	レコード数オーバー	BOOL	FALSE, TRUE	-	結果セットで抽出したレコード数が、結果セット変数「ResultSet」に格納可能な数を超えたことを表します
SendStatus	送信状態	_eDBC_SEND_STATUS	データ型に従う	-	SQL 文の送信の進捗状況を出力します

関連するシステム定義変数

「DB 接続サービス関連のシステム定義変数 (P.7-5)」を参照してください。

関連するデバイスパラメータ

NX シリーズに接続した X バスユニットの Ethernet ポートを使用する場合は、X バスユニットのデバイスパラメータを参照してください。

関連するエラーコード一覧

エラーコード	名称	内容
0400Hex	入力値範囲外	「QueryTimeOut」に有効範囲外の値を指定した場合
041BHex	データ容量超過	DB_AttachProcedure 命令で指定した結果セット変数「ResultSet」の配列要素数が多く処理できない場合 結果セット変数「ResultSet」の配列要素数を減らし、コントローラを再起動してください。
041DHex	命令同時実行リソースオーバ	DB 接続命令を同時に 32 個以上実行した場合
3000Hex	DB 接続サービス未開始	DB 接続サービスが動作中でないときに、本命令を実行した場合
3002Hex	DB 接続サービス終了／終了処理中	DB 接続サービスが終了／終了処理中に、本命令を実行した場合
3008Hex	DB コネクション不正	「DBConnection」が不正または既に切断している場合
300BHex	SQL 実行エラー	実行した SQL 文が DB でエラーとなった場合
3011Hex	DB コネクション異常切断発生中	ネットワーク障害などで DB と通信できない場合
3013Hex	DB 接続サービス異常停止中	DB 接続サービスが異常停止中に、本命令を実行した場合
3014Hex	スプールデータあり	既にスプールに保留した SQL 文が存在するため実行できない場合
3015Hex	DB 接続サービス初期処理中	DB 接続サービスが初期処理中に、本命令を実行した場合
3016Hex	DB 処理中	前回実行した DB_Insert／DB_Update／DB_Select／DB_Delete 命令が命令実行タイムアウトし DB 側の処理がまだ完了していない状態で命令を実行した場合
3018Hex	ストアドプロシージャハンドル不正	DB_AttachProcedure 命令にて取得した「ProcHandle」を指定していない場合 DB_DetachProcedure 命令にて解放済みの「ProcHandle」を指定した場合 該当コネクション以外で取得した「ProcHandle」を指定した場合
3019Hex	対象外データベース種別への命令実行	本命令の対象となっていないデータベース種別を指定して、命令を実行した場合

エラーコード	名称	内容
301CHex	ストアドプロシージャ結果セットカラム数不正	取得された結果セットのカラム数と格納先の構造体型変数「ResultSet」のメンバ数が一致しない場合
301EHex	DB 接続サービス未起動	DB 接続サービスを［使用しない］に設定した状態で、本命令を実行した場合

機能説明

DB_AttachProcedure 命令で取得したプロシージャハンドルを使って、ストアドプロシージャを呼び出します。
ストアドプロシージャの引数・戻り値・結果セットは、DB_AttachProcedure 命令で指定したそれぞれの変数を利用してストアドプロシージャを呼び出します。DB_ExecuteProcedure 命令では、スプール機能は、使用できません。

DB_ExecuteProcedure 命令を実行後に、結果セット変数「ResultSet」の配列に、結果セットのレコード数分の値を格納し、格納したレコード数を「RecCnt」に出力します。また、結果セットのレコード数が結果セット変数に格納可能なレコード数を超えた場合「Overflow」が TRUE になります。実行結果の詳細は、下記の表を参照してください。

DB_AttachProcedure 命令で指定した入力変数「ResultSet」のメンバ数と、DB_ExecuteProcedure 命令で呼び出したストアドプロシージャから返信された結果セットの列方向のデータ数（カラム数）の関係により、DB_ExecuteProcedure 命令は以下の実行結果となります。

<table>
<tr><th></th><th>結果セットのカラム数 ＜「ResultSet」のメンバ数</th><th>結果セットのカラム数 ＝「ResultSet」のメンバ数</th><th>結果セットのカラム数 ＞「ResultSet」のメンバ数</th></tr>
<tr><td>結果セットのレコード数 ＜「ResultSet」の配列要素数</td><td rowspan="3">命令エラー（ストアドプロシージャ結果セットカラム数不正）</td><td>正常終了。
「ResultSet」の配列には結果セットのレコード数分だけ値が書き込まれる。「OverFlow」は、False。「RecCnt」は、結果セットのレコード数。</td><td rowspan="3">命令エラー（ストアドプロシージャ結果セットカラム数不正）</td></tr>
<tr><td>結果セットのレコード数 ＝「ResultSet」の配列要素数</td></tr>
<tr><td>結果セットのレコード数 ＞「ResultSet」の配列要素数</td><td>正常終了。
「ResultSet」の配列には「ResultSet」の配列要素数分だけ値が書き込まれる。「OverFlow」は、True。「RecCnt」は、「ResultSet」の配列要素数。</td></tr>
</table>

DB_ExecuteProcedure 命令では、命令実行タイムアウトは使用できません。入力変数「QueryTimeOut」は、クエリ実行タイムアウトの時間になります。入力変数「QueryTimeOut」に 0 以外の値を指定した場合は、DB コネクション設定の「クエリ実行タイムアウト」で設定した時間ではなく、入力変数「QueryTimeOut」が有効になります。クエリ実行タイムアウトが発生した場合、命令エラー（SQL 実行エラー）が発生します。

使用上の注意

- 本命令は、「Execute」の値が FALSE になっても、また実行時間がタスク周期を超えても、処理を最後まで続けます。処理が正常終了したかどうかは、「Done」の値が TRUE になることで確認してください。
- 「Execute」、「Done」、「Busy」、「Error」のタイミングチャートは、『NJ/NX シリーズコマンドリファレンスマニュアル　基本編』（SBCA-468）の“本章の見方”を参照してください。
- ストアドプロシージャの呼び出し処理が、設定されたクエリ実行タイムアウトの時間内に完了しないと、命令エラー（SQL 実行エラー）が発生します。そのため、ストアドプロシージャを実行させ

るサーバでのストアドプロシージャの実行時間に、サーバの負荷変動を加味した時間を設定してください。
- 結果セットを格納する結果セット変数「ResultSet」の配列要素数が多い場合、本命令はエラーコード 041BHex（データ容量超過）で異常終了することがあります。その場合は、「ResultSet」の配列要素数を減らし、コントローラを再起動してください。
- 本命令は、イベントタスク上で使うことができません。コンパイル時にエラーになります。
- DB 接続サービスをテストモードで開始している場合、本命令は正常終了します。
- 本命令は、以下の場合、異常となります。「Error」は TRUE となります。
  a）DB 接続サービスが動作中でないときに、本命令を実行した場合
  b）DB 接続サービスが初期処理中に、本命令を実行した場合
  c）DB 接続サービスが異常停止中に、本命令を実行した場合
  d）DB 接続サービスが終了後／終了処理中に、本命令を実行した場合
  e）「DBConnection」が不正、または既に切断している場合
  f）「ProcHandle」が不正である場合
  g）実行した SQL 文が DB でエラーとなった場合
  h）ネットワーク障害などで DB と通信できない場合
  i）既にスプールに保留した SQL 文が存在する場合
  j）指定したクエリ実行タイムアウト時間内に命令が終了しなかった場合
  k）「QueryTimeOut」に有効範囲外の値を指定した場合
  l）前回実行した DB_Insert/DB_Update/DB_Select/DB_Delete 命令が命令実行タイムアウトし、サーバ側 DB の処理がまだ完了していない状態で命令を実行した場合
  m）DB 接続命令を同時に 32 個を超えて実行した場合

サンプルプログラム

ここでは、ストアドプロシージャの実行を行うサンプルプログラムを掲載します。
以下の処理を行うストアドプロシージャ「Proc1」を実行します。
- Result_Proc1 = ArgIn.Add1 + ArgIn.Add2;

構造体型の設定

このサンプルプログラムで使用する構造体の設定を以下に示します。

名称		データ型
ADD_STOREDPROCEDURE		STRUCT
	Add1	DINT
	Add2	DINT

ラダー図

● 主な変数

名称	データ型	初期値	コメント
DB_Connect_instance	DB_Connect	-	DB_Connect 命令のインスタンスです。
DB_Close_instance	DB_Close	-	DB_Close 命令のインスタンスです。

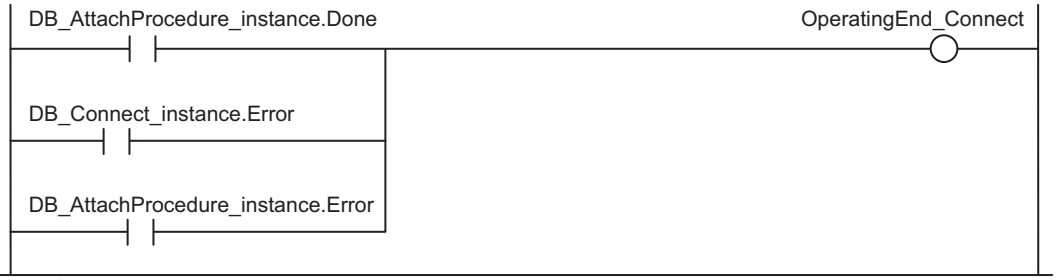
名称	データ型	初期値	コメント
MyDB1	DWORD	-	DB_Connect 命令のインスタンス DB_Connect_instance の出力 DBConnection に割り付けた変数です。
Handle_Proc1	DWORD	-	プロシージャハンドルを表す変数です。
Trigger_Connect	BOOL	-	DB コネクション接続のトリガとなる変数です。
Operating_Connect	BOOL	-	この変数が TRUE のときに、DB_Connect 命令を実行します。
OperatingEnd_Connect	BOOL	-	DB_Connect 命令が終了すると、TRUE になります。
RS_Connect_instance	RS	-	RS 命令のインスタンスです。
Trigger_StoredProcedure	BOOL	-	DB ストアドプロシージャ実行のトリガとなる変数です。
Operating_StoredProcedure	BOOL	-	この変数が TRUE のときに、DB_ExecuteProcedure 命令を実行します。
OperatingEnd_StoredProcedure	BOOL	-	DB_ExecuteProcedure 命令が終了すると、TRUE になります。
RS_StoredProcedure_instance	RS	-	RS 命令のインスタンスです。
Trigger_Close	BOOL	-	DB コネクション切断のトリガとなる変数です。
Operating_Close	BOOL	-	この変数が TRUE のときに、DB_Close 命令を実行します。
OperatingEnd_Close	BOOL	-	この変数が TRUE のときに、DB_Close 命令を実行します。
RS_Close_instance	RS	-	RS 命令のインスタンスです。
DB_AttachProcedure_instance	DB_AttachProcedure	-	DB_AttachProcedure 命令のインスタンスです。
DB_ExecuteProcedure_instance	DB_ExecuteProcedure	-	DB_ExecuteProcedure 命令のインスタンスです。
DB_DetachProcedure_instance	DB_DetachProcedure	-	DB_DetachProcedure 命令のインスタンスです。
Result_Proc1	DINT	-	ストアドプロシージャの戻り値を表す変数です。
ArgIn	ADD_STOREDPROCEDURE	-	ストアドプロシージャの IN 引数を表す変数です。

● サンプルプログラム

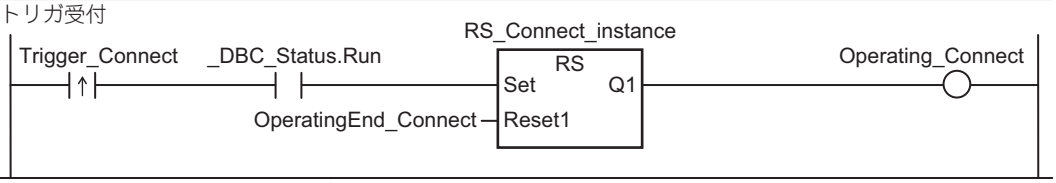
0 ◆2つの変数の加算処理をストアドプロシージャによって、DB側で処理させます。
処理の手順は以下のとおりです。
1．DB_Connect命令で、DBに接続します。
2．DB_AttachProcedure命令で、実行するストアドプロシージャのプロシージャハンドルを取得します。
3．DB_ExecuteProcedure命令で、ストアドプロシージャを実行します。
4．DB_DetachProcedure命令でプロシージャハンドルを解放します。DB_Close命令で、DBとのDBコネクションを切断します。

◆DBコネクション名「MyDatabese1」に接続し、ストアドプロシージャのプロシージャハンドルを取得します。

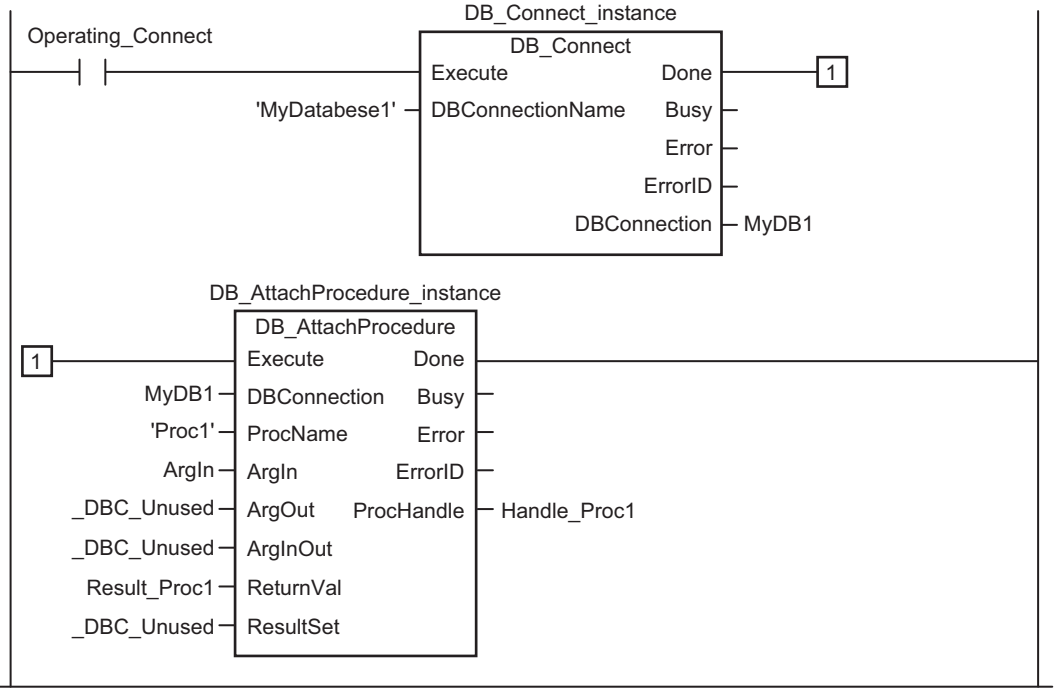
DB_Connect命令、DB_AttachProcedure命令の実行完了の判断



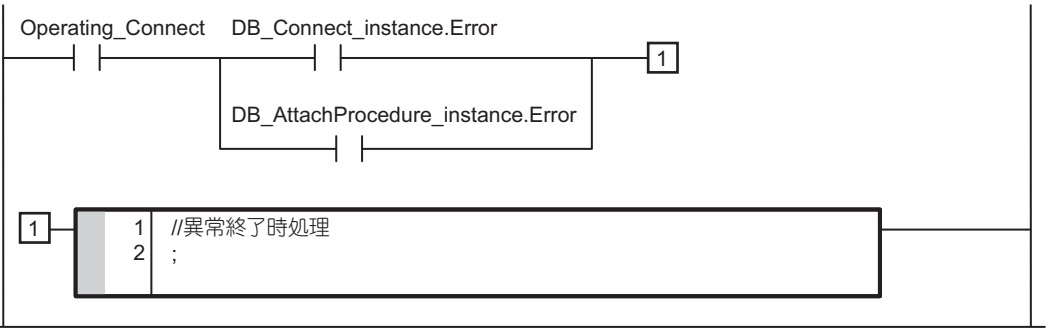
1 トリガ受付



2 DBコネクション名「MyDatabase1」に接続します。
DBコネクション「MyDB1」のストアドプロシージャ「Proc1」のプロシージャハンドルを取得します。



3 異常終了時の処理



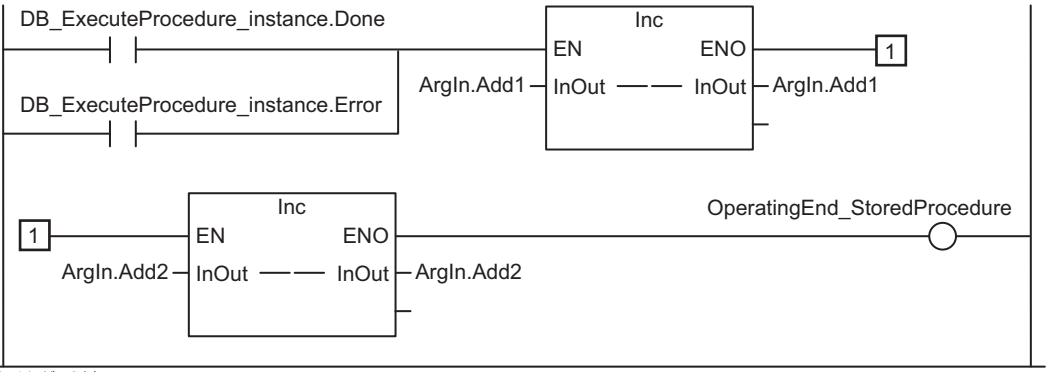
4 ----------

◆以下のストアドプロシージャ「Proc1」を実行します。

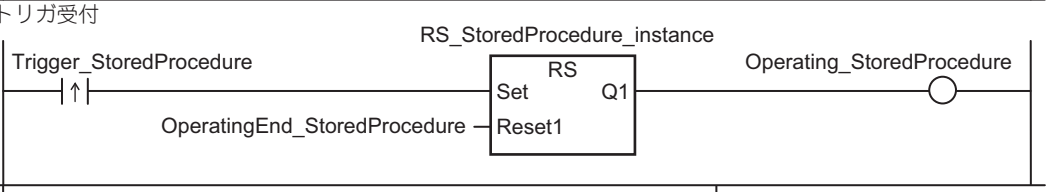
・Proc1

```
Result_Proc1 = ArgIn.Add1 + ArgIn.Add2;
```

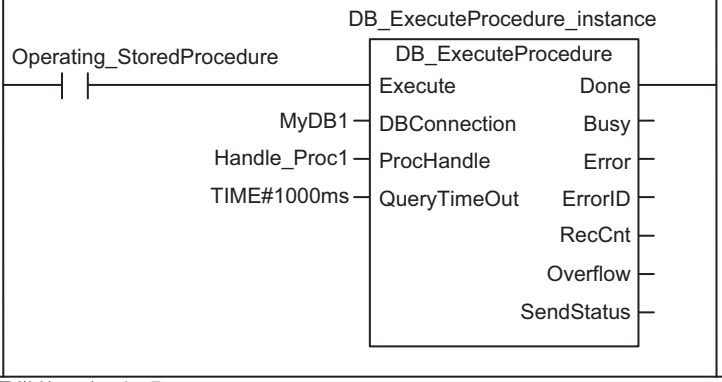
DB_ExecuteStoredProcedure命令実行完了の判断



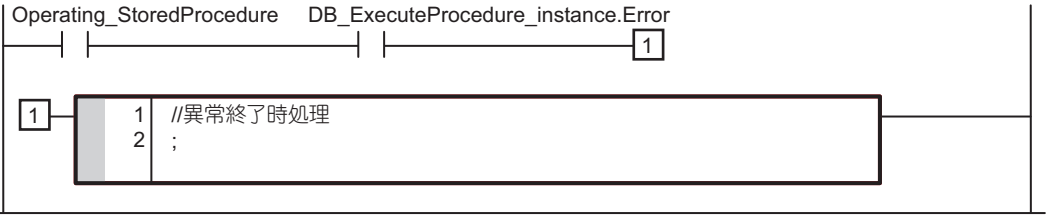
5 トリガ受付

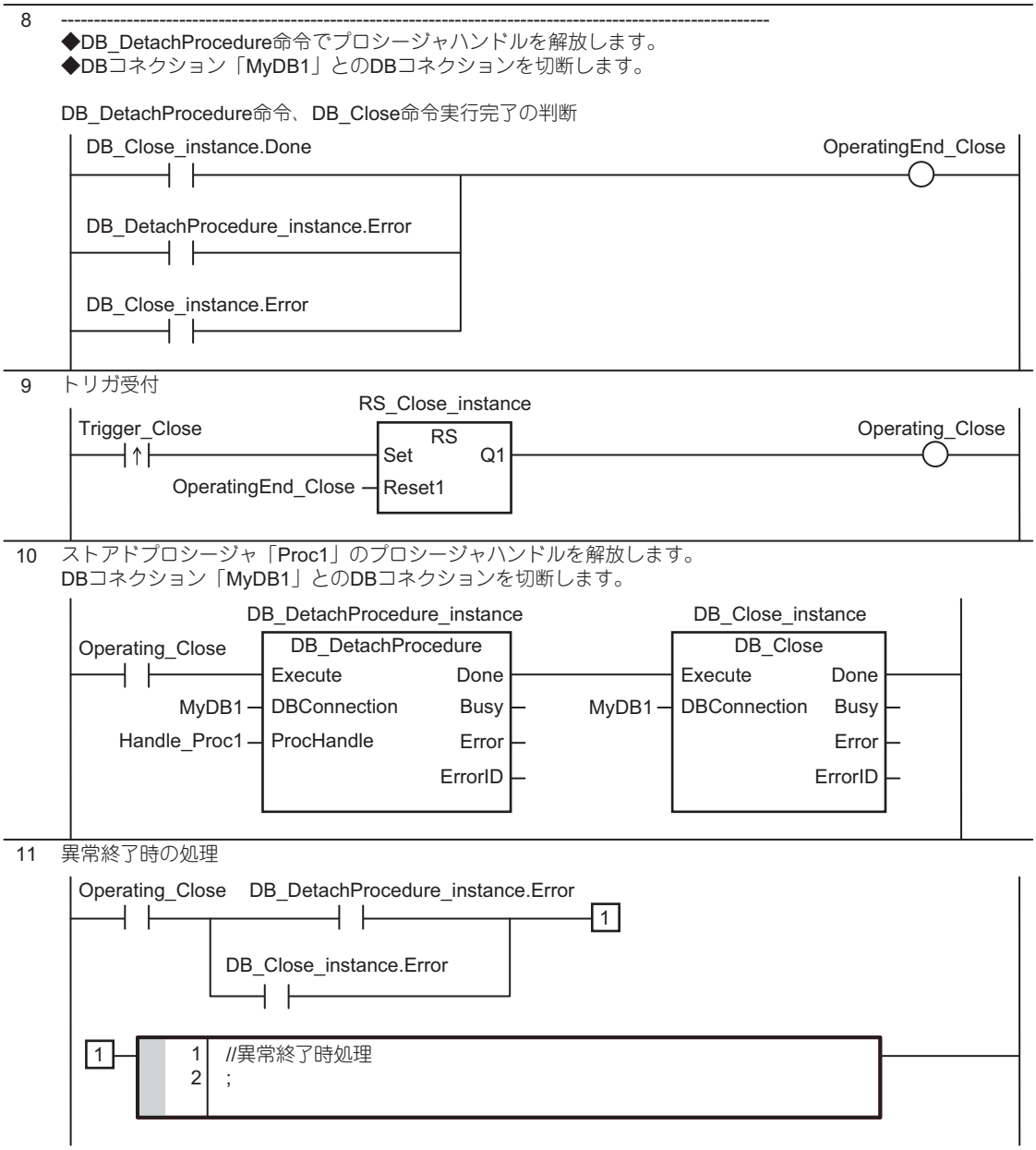


6



7 異常終了時の処理





ストラクチャードテキスト（ST）

● 主な変数

名称	データ型	初期値	コメント
DB_Connect_instance	DB_Connect	-	DB_Connect 命令のインスタンスです。
DB_Close_instance	DB_Close	-	DB_Close 命令のインスタンスです。
MyDB1	DWORD	-	DB_Connect 命令のインスタンス DB_Connect_instance の出力 DBConnection に割り付けた変数です。

名称	データ型	初期値	コメント
Trigger_Connect	BOOL	-	DB コネクション接続のトリガとなる変数です。
LastTrigger_Connect	BOOL	-	前回実行時のトリガ状態を保持する変数です。
Operating_Connect	BOOL	-	この変数が TRUE のときに、DB_Connect 命令を実行します。
OperatingStart_Connect	BOOL	-	この変数が TRUE のときに、DB コネクション接続開始時の処理を実行します。
Trigger_StoredProcedure	BOOL	-	DB ストアドプロシージャ実行のトリガとなる変数です。
LastTrigger_StoredProcedure	BOOL	-	前回実行時のトリガ状態を保持する変数です。
Operating_StoredProcedure	BOOL	-	この変数が TRUE のときに、DB_ExecuteProcedure 命令を実行します。
Trigger_Close	BOOL	-	DB コネクション切断のトリガとなる変数です。
LastTrigger_Close	BOOL	-	前回実行時のトリガ状態を保持する変数です。
Operating_Close	BOOL	-	この変数が TRUE のときに、DB_Close 命令を実行します。
OperatingStart_Close	BOOL	-	この変数が TRUE のときに、DB コネクション切断開始時の処理を実行します。
Stage	INT	-	DB コネクション接続の状態を表す変数です。
DB_AttachProcedure_instance	DB_AttachProcedure	-	DB_AttachProcedure 命令のインスタンスです。
DB_ExecuteProcedure_instance	DB_ExecuteProcedure	-	DB_ExecuteProcedure 命令のインスタンスです。
DB_DetachProcedure_instance	DB_DetachProcedure	-	DB_DetachProcedure 命令のインスタンスです。
ArgIn	ADD_STOREDPROCEDURE	-	ストアドプロシージャの IN 引数を表す変数です。
Handle_Proc1	DWORD	-	プロシージャハンドルを表す変数です。
Result_Proc1	DINT	-	ストアドプロシージャの戻り値を表す変数です。

● サンプルプログラム

```
(* ---------------------------------------------------------------------------------
◆２つの変数の加算処理をストアドプロシージャによって、DB 側で処理させます。
処理の手順は以下のとおりです。
１．DB_Connect 命令で、DB に接続します。
２．DB_AttachProcedure 命令で、実行するストアドプロシージャのプロシージャハンドルを取得します。
３．DB_ExecuteProcedure 命令で、ストアドプロシージャを実行します。
４．DB_DetachProcedure 命令でプロシージャハンドルを解放します。DB_Close 命令で、DB との DB コネクションを切断します。
*)
```

```
//-------------------------------------------------------------------------------
//◆DB コネクション名「MyDatabese1」に接続し、ストアドプロシージャのプロシージャハンドルを取得します。

//Trigger_Connect の立上りでシーケンス開始
IF ( (Trigger_Connect=TRUE) AND (LastTrigger_Connect=FALSE) AND (_DBC_Status.Run=TRUE) ) THEN
        OperatingStart_Connect := TRUE;
        Operating_Connect := TRUE;
END_IF;
LastTrigger_Connect:=Trigger_Connect;

//シーケンス開始時処理
IF (OperatingStart_Connect=TRUE) THEN
        //DB 接続命令インスタンス初期化
        DB_Connect_instance(Execute:=FALSE );
        DB_AttachProcedure_instance( Execute:=FALSE, DBConnection:=MyDB1, ProcName:='Proc1', ArgIn:=
_DBC_Unused, ArgOut:=_DBC_Unused, ArgInOut:=_DBC_Unused, ReturnVal:=Result_Proc1, ResultSet:=_DBC_Un
used);

        Stage := INT#1;
        OperatingStart_Connect := FALSE;
END_IF;

//DB コネクション名「MyDatabese1」に接続します。
//DB コネクション「MyDB1」のストアドプロシージャ「Proc1」のプロシージャハンドルを取得します。
IF (Operating_Connect=TRUE) THEN
        CASE Stage OF
        1 : //DB コネクション接続
                DB_Connect_instance( Execute:=TRUE, DBConnectionName:='MyDatabase1', DBConnecti
on=>MyDB1 );

                IF (DB_Connect_instance.Done=TRUE) THEN
                        Stage := INT#2; // 正常終了
                ELSIF (DB_Connect_instance.Error=TRUE) THEN
                        Stage := INT#99; // 異常終了
                END_IF;

         2 : //プロシージャハンドルを取得
                DB_AttachProcedure_instance( Execute:=TRUE, DBConnection:=MyDB1, ProcName:='Pro
c1', ArgIn:=ArgIn, ArgOut:=_DBC_Unused, ArgInOut:=_DBC_Unused, ReturnVal:=Result_Proc1, ResultSet:=_
DBC_Unused, ProcHandle=>Handle_Proc1);

                IF ( DB_AttachProcedure_instance.Done=TRUE) THEN
                        Operating_Connect:=FALSE; // 正常終了
                ELSIF ( DB_AttachProcedure_instance.Error=TRUE ) THEN
                        Stage := INT#99; // 異常終了
                END_IF;

         99 :
                //異常終了処理
                Operating_Connect := FALSE;
        END_CASE;
```

```
END_IF;

//----------------------------------------------------------------------------------
//◆以下のストアドプロシージャ「Proc1」を実行します。
//・Proc1
//Result_Proc1 = ArgIn.Add1 + ArgIn.Add2;

//Trigger_StoredProcedure の立上りでシーケンス開始
IF ( (Trigger_StoredProcedure=TRUE) AND (LastTrigger_StoredProcedure=FALSE) ) THEN
        Operating_StoredProcedure := TRUE;
        //DB 接続命令インスタンス初期化
        DB_ExecuteProcedure_instance( Execute:=FALSE );
END_IF;
LastTrigger_StoredProcedure := Trigger_StoredProcedure;

IF( Operating_StoredProcedure=TRUE )THEN
        DB_ExecuteProcedure_instance( Execute:=TRUE, DBConnection:=MyDB1, ProcHandle:=Handle_Proc1,
QueryTimeOut:=TIME#1000ms );

        IF (DB_ExecuteProcedure_instance.Done=TRUE) THEN
                Operating_StoredProcedure := FALSE; // 正常終了
                Inc(ArgIn.Add1);
                Inc(ArgIn.Add2);
        ELSIF (DB_ExecuteProcedure_instance.Error=TRUE) THEN
                //異常終了処理
                Operating_StoredProcedure := FALSE;
                Inc(ArgIn.Add1);
                Inc(ArgIn.Add2);
        END_IF;
END_IF;

(*----------------------------------------------------------------------------------
◆DB_DetachProcedure 命令でプロシージャハンドルを解放します。
◆DB コネクション「MyDB1」との DB コネクションを切断します。
*)

//Trigger_Close の立上りでシーケンス開始
IF ( (Trigger_Close=TRUE) AND (LastTrigger_Close=FALSE) ) THEN
        OperatingStart_Close := TRUE;
        Operating_Close := TRUE;
END_IF;
LastTrigger_Close := Trigger_Close;

//シーケンス開始時処理
IF (OperatingStart_Close=TRUE) THEN
        //DB 接続命令インスタンス初期化
        DB_DetachProcedure_instance( Execute:=FALSE );
        DB_Close_instance( Execute:=FALSE );

        OperatingStart_Close := FALSE;
END_IF;
```

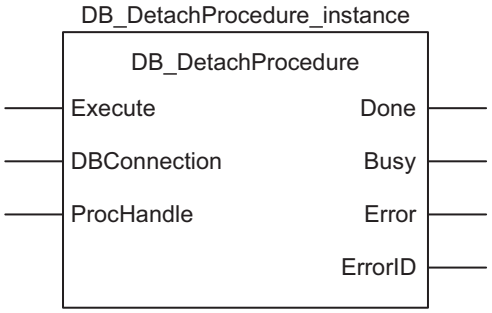
```
//ストアドプロシージャ「Proc1」のプロシージャハンドルを解放します。
//DB コネクション「MyDB1」との DB コネクションを切断します。
IF (Operating_Close=TRUE) THEN
        //プロシージャハンドルを解放
        DB_DetachProcedure_instance(Execute:=TRUE, DBConnection:=MyDB1, ProcHandle:=Handle_Proc1 );

        IF (DB_DetachProcedure_instance.Done=TRUE) THEN
                    //DB コネクション切断
                    DB_Close_instance( Execute:=TRUE, DBConnection:=MyDB1 );
        ELSIF (DB_DetachProcedure_instance.Error=TRUE) THEN
                    //異常終了処理
                    //DB コネクション切断
                    DB_Close_instance( Execute:=TRUE, DBConnection:=MyDB1 );
        END_IF;

        IF (DB_Close_instance.Done=TRUE) THEN
                    Operating_Close := FALSE; // 正常終了
        ELSIF (DB_Close_instance.Error=TRUE) THEN
                    //異常終了処理
                    Operating_Close := FALSE;
        END_IF;
END_IF;
```

DB_DetachProcedure（DB ストア ドプロシージャ ハンドル解放）

DB_AttachProcedure で取得したプロシージャハンドルを解放します。

命令	名称	FB/FUN	グラフィック表現	ST 表現
DB_DetachProcedure	DB ストアドプロシージャハンドル解放	FB		DB_DetachProcedure_instance (Execute, DBConnection, ProcHandle, Done, Busy, Error, ErrorID);

（注）DB_DetachProcedure 命令のインスタンスとして、インスタンス名 DB_DetachProcedure_instance で変数宣言されている場合の例です。

変数

入力変数

入力変数	名称	データ型	有効範囲	単位	初期値	内容
Execute	起動	BOOL	FALSE, TRUE	-	FALSE	実行条件を指定します
DBConnection	DB コネクション	DWORD	16#0 ～ FFFFFFFF	-	16#0	DB_Connect で取得した DB コネクション
ProcHandle	プロシージャハンドル	DWORD	16#0 ～ FFFFFFFF	-	16#0	DB_AttachProcedure 命令で取得したプロシージャハンドル

出力変数

出力変数	名称	データ型	有効範囲	単位	内容
Done	完了	BOOL	FALSE, TRUE	-	正常終了を表します
Busy	実行中	BOOL	FALSE, TRUE	-	処理中を表します
Error	エラー	BOOL	FALSE, TRUE	-	異常終了を表します
ErrorID	エラーコード	WORD	16#0 ～ FFFF	-	エラーコードを表します

関連するシステム定義変数

「DB 接続サービス関連のシステム定義変数 (P.7-5)」を参照してください。

関連するデバイスパラメータ

NX シリーズに接続した X バスユニットの Ethernet ポートを使用する場合は、X バスユニットのデバイスパラメータを参照してください。

関連するエラーコード一覧

エラーコード	名称	内容
041DHex	命令同時実行リソースオーバ	DB 接続命令を同時に 32 個以上実行した場合
3000Hex	DB 接続サービス未開始	DB 接続サービスが動作中でないときに、本命令を実行した場合
3002Hex	DB 接続サービス終了／終了処理中	DB 接続サービスが終了後／終了処理中に、本命令を実行した場合
3008Hex	DB コネクション不正	「DBConnection」が不正、または既に切断している場合
3013Hex	DB 接続サービス異常停止中	DB 接続サービスが異常停止中に、本命令を実行した場合
3015Hex	DB 接続サービス初期処理中	DB 接続サービスが初期処理中に、本命令を実行した場合
3018Hex	ストアドプロシージャハンドル不正	DB_AttachProcedure 命令にて取得した「ProcHandle」を指定していない場合 DB_DetachProcedure 命令にて解放済みの「ProcHandle」を指定した場合 該当コネクション以外で取得した「ProcHandle」を指定した場合
3019Hex	対象外データベース種別への命令実行	本命令の対象となっていないデータベース種別を指定して、命令を実行した場合
301EHex	DB 接続サービス未起動	DB 接続サービスを［使用しない］に設定した状態で、本命令を実行した場合

機能説明

DB_AttachProcedure で取得したプロシージャハンドルを解放します。

使用上の注意

- 本命令は、「Execute」の値が FALSE になっても、また実行時間がタスク周期を超えても、処理を最後まで続けます。処理が正常終了したかどうかは、「Done」の値が TRUE になることで確認してください。
- 「Execute」、「Done」、「Busy」、「Error」のタイミングチャートは、『NJ/NX シリーズコマンドリファレンスマニュアル　基本編』（SBCA-468）の“本章の見方”を参照してください。
- 本命令は、イベントタスク上で使うことができません。コンパイル時にエラーになります。
- DB 接続サービスをテストモードで開始している場合、本命令は正常終了します。
- 本命令は、以下の場合、異常となります。「Error」は TRUE となります。
  a）DB 接続サービスが動作中でないときに、本命令を実行した場合
  b）DB 接続サービスが初期処理中に、本命令を実行した場合
  c）DB 接続サービスが異常停止中に、本命令を実行した場合
  d）DB 接続サービスが終了後／終了処理中に、本命令を実行した場合
  e）「DBConnection」が不正、または既に切断している場合
  f）「ProcHandle」が不正である場合
  g）DB 接続命令を同時に 32 個を超えて実行した場合

サンプルプログラム

サンプルプログラムは、DB_ExecuteProcedure 命令のサンプルプログラム「サンプルプログラム (P.7-106)」を参照してください。

8

異常の確認と処置

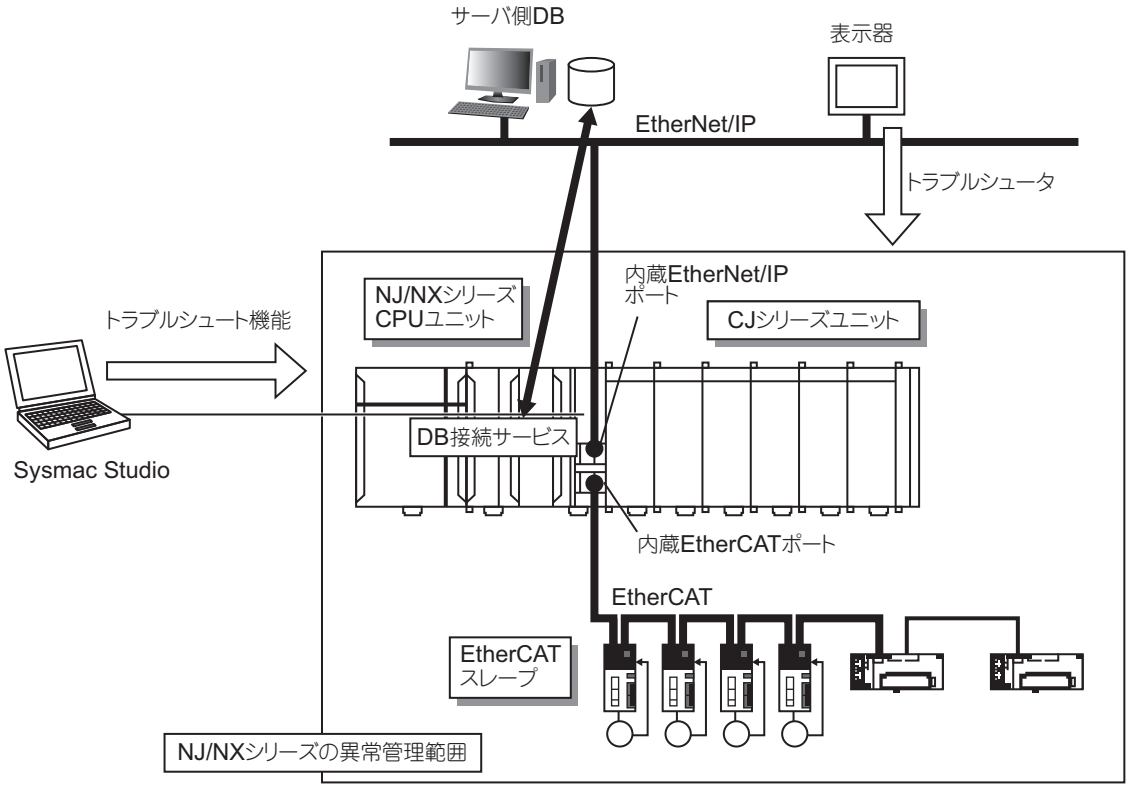
この章では、DB 接続サービスでの異常の確認と処置について説明しています。

8-1　異常の概要 8-2
8-1-1　異常の確認方法 8-2
8-1-2　DB 接続サービスに関する異常 8-5
8-2　トラブルシューティング 8-7
8-2-1　異常一覧 8-7
8-2-2　異常詳細 8-14

8-1 異常の概要

NJ/NX シリーズは、コントローラで発生した異常をイベントとして共通の方法で一元的に管理します。これにより、管理範囲（CPU ユニット、EtherCAT スレーブ*1、CJ シリーズユニット）の異常に対して、共通の方法で異常の内容と対処方法を確認できます。

*1. Sysmac デバイスであることが必要です。



Sysmac Studio のトラブルシュート機能または表示器のトラブルシュータを使用すると、異常の内容から対処方法までを確認できるため、迅速な対応が可能です。

本マニュアルでは、DB 接続サービスが発生源となる異常について説明しています。

異常発生時の具体的な対処方法、および NJ/NX シリーズ全体のトラブルシュートについては、『NJ/NX シリーズ トラブルシューティングマニュアル』（SBCA-469）を参照してください。

DB 接続サービス命令の起動時に発生する異常については、各命令の詳細は、「第 7 章 DB 接続命令 (P.7-1)」を参照してください。

8-1-1 異常の確認方法

以下の方法で異常の発生状態を確認できます。

確認方法	確認できること
LED による確認	CPU ユニットの動作状態
Sysmac Studio のトラブルシュート機能による確認	発生中のコントローラ異常、過去に発生したコントローラ異常の履歴、異常の発生源、異常の発生要因と処置対策、CJ シリーズ高機能ユニットの異常履歴*1
表示器のトラブルシュータによる確認*2	発生中のコントローラ異常、過去に発生したコントローラ異常の履歴、異常の発生源、発生要因と処置対策
機能モジュールごとの異常ステータス取得命令による確認	発生中のコントローラ異常の最重要のステータスと最重要のイベントコード
システム定義変数による確認	各機能モジュールにおける発生中のコントローラ異常のステータス

*1. 異常の発生要因や処置対策などの詳細な情報は表示されません。

*2. 表示器のトラブルシュータを使用する場合は、表示器を CPU ユニットの内蔵 EtherNet/IP ポート、または CPU ユニットに接続した NX シリーズ EtherNet/IP ユニットの EtherNet/IP ポートに接続してください。

以下に各確認方法について説明します。

LED による確認

電源ユニットの PWR LED と CPU ユニットの RUN、ERROR LED の表示により、異常の重要度を確認することができます。LED の表示と異常の重要度の関係は以下のとおりです。

LED			CPU ユニットの動作状態	Sysmac Studio／表示器
PWR	RUN	RUN		
●	●	●	電源部異常	不可： 『NJ/NX シリーズ トラブルシューティングマニュアル』（SBCA-469）を参照してください
○	●	●	CPU リセット*1	
○	◎	○	電源部接続不正	
○	●	○	CPU 異常（WDT 異常）*2	
○	●	○	全停止フォールトレベル*2	可能： Sysmac Studio／表示器を接続し、トラブルシュート機能／トラブルシュータにより原因と処置を確認してください
○	○	◎	部分停止フォールトレベル	
○	○	◎	軽度フォールトレベル	
○	○	●	監視情報	
○	○	●	正常：運転モード中	－
○	●	●	正常：プログラムモード中*1	－
○	◎	●	正常：起動中	－

*1. Sysmac Studio を“USB- 直接接続”で接続し、オンライン接続できれば CPU ユニットは“プログラムモード中”、オンライン接続できなければ CPU ユニットは“CPU リセット”となります。*3

*2. Sysmac Studio を“USB- 直接接続”で接続し、オンライン接続できれば CPU ユニットは“全停止フォールトレベル”の異常、オンライン接続できなければ“CPU 異常（WDT 異常）”となります。*3

*3. Sysmac Studio がオンライン接続できない場合でも、USB ケーブルの不良や Sysmac Studio の“ネットワーク種別”が“USB- 直接接続”に設定されていない可能性があります。オンライン接続できない場合については、『NJ/NX シリーズ トラブルシューティングマニュアル』（SBCA-469）を参照してください。

使用上の注意

形 NX701-□□20 のハードウェアリビジョン A 以降、形 NX502-1□00 CPU ユニットおよび形 NX102-□□20CPU ユニットには USB ポートがないため、“USB-直接接続” で Sysmac Studio を接続することはできません。

Sysmac Studio のトラブルシュート機能による確認

異常発生時、Sysmac Studio をコントローラにオンライン接続することで、発生中の異常、過去に発生した異常の履歴を確認することができます。
また、異常の発生要因と処置対策を確認することができます。

Sysmac Studio を使った異常の確認方法と対処方法については、『NJ/NX シリーズ トラブルシューティングマニュアル』（SBCA-469）を参照してください。

表示器のトラブルシュータによる確認

異常発生時、表示器とコントローラとの通信が確立していれば、発生中の異常、過去に発生した異常の履歴を確認することができます。
また、異常の発生要因と処置対策を確認することができます。

表示器を使った異常の確認方法と対処方法については、『NJ/NX シリーズ トラブルシューティングマニュアル』（SBCA-469）を参照してください。

異常ステータス取得命令による異常の確認

ユーザプログラムでは、機能モジュールごとの異常ステータス取得命令により異常ステータスを取得できます。
DB 接続サービスの異常ステータス取得命令は以下のとおりです。

命令	名称	機能
GetPLCError	PLC 異常ステータス取得	PLC 機能モジュールで発生中のコントローラ異常の、最重要のステータス（部分停止フォールト、軽度フォールト）と、最重要のイベントコードを取得します

異常ステータス取得命令については、『NJ/NX シリーズ コマンドリファレンスマニュアル 基本編』（SBCA-468）を参照してください。

システム定義変数による確認

DB 接続サービスで発生した異常は、システム定義変数の“異常ステータス変数”、および“ステータス変数”により確認できます。

● 異常ステータス変数

NJ/NX シリーズで発生している異常は、各機能モジュールの異常ステータス変数として確認できます。
PLC 機能モジュールの異常ステータス変数は下記のとおりです。

変数名	データ型	名称	機能
_PLC_ErrSta	WORD	PLC 機能モジュール異常ステータス	PLC 機能モジュールで発生している全ての異常の状態を集約して表します

● ステータス変数

変数名	データ型	名称	機能
_DBC_Status	_sDBC_STATUS	DB 接続サービスステータス	DB 接続サービスの状態を表します

変数名	データ型	名称	機能
Run	BOOL	動作中フラグ	DB 接続サービスが動作中のとき TRUE となります。 DB 接続サービスが動作中でないとき FALSE となります。
Test	BOOL	テストモード	DB 接続サービスがテストモードで動作中のとき TRUE となります。 DB 接続サービスがテストモードで動作中でないとき FALSE となります。
Idle	BOOL	アイドル	DB 接続サービスがアイドル中のとき TRUE となります。 DB 接続サービスがアイドル中でないとき FALSE となります。
Error	BOOL	異常発生中フラグ	DB 接続サービスで異常発生中のとき TRUE となります。 DB 接続サービスで異常発生中でないとき FALSE となります。
Shutdown	BOOL	終了	DB 接続サービスが終了しているとき TRUE となります。 DB 接続サービスが終了していないとき FALSE となります。

8-1-2 DB 接続サービスに関する異常

分類

DB 接続サービスの異常は以下の 2 つの発生源に分類しています。

検出場所	イベント発生源	イベント発生源詳細	格納先イベントログのカテゴリ		
			システムログ	アクセスログ	ユーザイベントログ
DB 接続サービス	PLC 機能モジュール	DB 接続サービス	○	×	×
DB 接続命令	PLC 機能モジュール	命令	○	×	×

異常の重要度

異常の重要度ごとの DB 接続サービスの動作について説明します。

異常の重要度レベル	動作
全停止フォールト	NJ/NX シリーズコントローラ全体の制御が停止する異常です
部分停止フォールト	NJ/NX シリーズコントローラの機能モジュール単位ごとに制御が停止する異常です DB 接続サービスで発生すると、DB 接続サービスが持つ全ての機能が停止します
軽度フォールト	NJ/NX シリーズコントローラの各機能モジュールの一部の制御が停止する異常です
監視情報	NJ/NX シリーズコントローラの制御に影響しない異常です 軽度フォールトレベル以上の異常に発展することを防止する目的で通知します
一般情報	異常以外のユーザに通知する情報です

DB 接続サービスの発生源ごとの異常

発生源ごとに各重要度レベルの異常を記載しています。

DB 接続サービスが発生源の異常

異常の重要度レベル	異常の名称
全停止フォールト	なし
部分停止フォールト	なし
軽度フォールト	• スプール破壊 • 実行ログ保存失敗 • SQL 実行失敗ログ保存失敗 • DB 接続設定エラー • DB コネクション異常切断
監視情報	なし
一般情報	• DB 接続サービス開始 • DB 接続サービス停止 • DB 接続サービス終了 • スプールクリア

DB 接続命令が発生源の異常

異常の重要度レベル	異常の名称
全停止フォールト	なし
部分停止フォールト	なし
軽度フォールト	なし
監視情報	• DB 接続サービス未開始 • DB 接続サービス動作モード変更失敗 • DB 接続サービス終了／終了処理中 • DB コネクション名不正 • DB コネクション接続拒否 • DB コネクション接続失敗 • DB コネクション既接続 • DB コネクション接続数超過 • DB コネクション不正 • DB マップ変数不正 • DB マップ変数未登録 • SQL 実行エラー • スプール容量超過 • 抽出条件不正 • ログコード範囲外 • DB コネクション異常切断発生中 • DB 接続命令実行タイムアウト • DB 接続サービス異常停止中 • スプールデータあり • DB 接続サービス初期処理中 • DB 処理中 • 稼働ログ無効
一般情報	なし

8-2 トラブルシューティング

DB 接続サービスで発生する異常とその処置を説明します。

8-2-1 異常一覧

DB 接続サービス、DB 接続命令で発生する異常（イベント）の一覧を以下に示します。重要度の欄では以下のような略称と記号を使用しています。

略称	名称
全	全停止フォールトレベル
部	部分停止フォールトレベル
軽	軽度フォールトレベル
監	監視情報
般	一般情報

記号	意味
○	システムにより定義されている重要度
◎	ユーザが変更可能な重要度*1

*1. 変更可能な重要度がある場合にだけ記載します。

NJ/NX シリーズのすべてのイベントコードについては、『NJ/NX シリーズトラブルシューティングマニュアル』(SBCA-469)を参照してください。

DB 接続サービス関連

イベントコード	イベント名	内容	発生要因（推定原因）	重要度					参照先
				全	部	軽	監	般	
14D00000Hex	スプール破壊	スプールが破壊されています	• ユーザアプリケーションからのスプールへの不正な書込み • ［DB 接続サービス設定］の［運転時のサービス開始］を［使用しない］以外から［使用しない］に変更し、再度［使用しない］以外に変更した			○			P.8-15
14D20000Hex	実行ログ保存失敗	実行ログの SD メモリカードへの保存に失敗しました	• SD メモリカードが未装着 • SD メモリカードが種別不正 • SD メモリカードのフォーマット不正 • SD メモリカードがライトプロテクトされている • SD メモリカードの容量不足 • SD メモリカードの故障			○	◎		P.8-16
14D30000Hex	SQL 実行失敗ログ保存失敗	SQL 実行失敗ログの SD メモリカードへの保存に失敗しました	• SD メモリカードが未装着 • SD メモリカードが種別不正 • SD メモリカードのフォーマット不正 • SD メモリカードがライトプロテクトされている • SD メモリカードの容量不足 • SD メモリカードの故障			○	◎		P.8-17

イベントコード	イベント名	内容	発生要因（推定原因）	重要度					参照先
				全	部	軽	監	般	
35300000Hex	DB 接続設定エラー	DB 接続設定が不正です	• DB 接続設定をダウンロード中にコントローラを電断した • メモリオールクリア中にコントローラを電断したため DB 接続設定が不正 • リストア中にコントローラを電断したため DB 接続設定が不正 • 不揮発性メモリの故障			○			P.8-18
35310000Hex	DB サーバ証明書エラー	DB サーバ証明書のフォーマットが不正です。	• フォーマット(X.509)が不正な DB サーバ証明書をダウンロードした • DB 接続設定転送中に CPU ユニットを電断した • メモリオールクリア中にコントローラを電断したため DB 接続設定が不正 • リストア中にコントローラを電断したため DB 接続設定が不正 • 不揮発性メモリの故障			○			P.8-19
441C0000Hex	DB 接続システム処理異常	DB 接続サービスで致命的な異常を検知しました	• ソフトウェアエラーが発生した			○			P.8-19
85100000Hex	DB コネクション異常切断	DB コネクションが異常切断しました	• サーバの電源 OFF • サーバの DB の停止 • Ethernet ケーブルのコネクタのはずれ • Ethernet ケーブルの断線 • ノイズ			○			P.8-20
95300000Hex	DB 接続サービス開始	DB 接続サービスを開始しました	• DB 接続サービスの開始に成功した					○	P.8-20
95310000Hex	DB 接続サービス停止	DB 接続サービスを停止しました	• DB 接続サービスを停止した					○	P.8-21
95320000Hex	DB 接続サービス終了	DB 接続サービスを終了しました	• DB 接続サービスを終了した					○	P.8-21
95330000Hex	スプールクリア	スプールに保留した SQL 文をクリアしました	• スプールに保留した SQL 文をクリアした					○	P.8-21
95340000Hex	DB 接続サービス操作（開始）	DB 接続サービスの開始操作をしました	• SysmacStudio を使用して DB 接続サービスの開始操作をした					○	P.8-22
95350000Hex	DB 接続サービス操作（停止）	DB 接続サービスの停止操作をしました	• SysmacStudio を使用して DB 接続サービスの停止操作をした					○	P.8-22
95360000Hex	DB 接続サービス操作（終了）	DB 接続サービスの終了操作をしました	• SysmacStudio を使用して DB 接続サービスの終了操作をした					○	P.8-23
95370000Hex	スプールクリア操作	スプールに保留した SQL 文のクリア操作をしました	• SysmacStudio を使用してスプールに保留した SQL 文のクリア操作をした					○	P.8-23

イベントコード	イベント名	内容	発生要因（推定原因）	重要度					参照先
				全	部	軽	監	般	
95380000Hex	稼働ログのクリア操作	稼働ログのクリア操作をしました	• SysmacStudio を使用して稼働ログのクリア操作をした					○	P.8-24
95390000Hex	デバッグログ操作（開始）	デバッグログの開始操作をしました	• SysmacStudio を使用してデバッグログの開始操作をした					○	P.8-24
953A0000Hex	デバッグログ操作（停止）	デバッグログの停止操作をしました	• SysmacStudio を使用してデバッグログの停止操作をした					○	P.8-25

DB 接続命令関連

イベントコードの下位 4 けたが命令のエラーコードを表します。各エラーコードの説明は、対応するイベントコードの説明を参照してください。たとえば、命令のエラーコードが 16#3000 の場合は、54013000Hex のイベントコードの説明を参照してください。

イベントコード	イベント名	内容	発生要因（推定原因）	重要度					参照先
				全	部	軽	監	般	
54013000Hex	DB 接続サービス未開始	DB 接続サービスが開始していません	• 該当命令を実行する前に、DB 接続サービスの開始指示をしていない • 該当命令を実行する前に、DB 接続サービスの停止指示をした				○		P.8-26
54013001Hex	DB 接続サービス動作モード変更失敗	DB 接続サービスの動作モードを変更できませんでした	• “稼働モード”で動作中に、該当命令で“テストモード”への動作モード変更を実行した • “テストモード”で動作中に、該当命令で“稼働モード”への動作モード変更を実行した • DB 接続サービスが停止処理中に、DB 接続サービスの開始を指示した • DB 接続サービスが停止処理中に DB 接続サービスのシャットダウンを指示した				○		P.8-27
54013002Hex	DB 接続サービス終了／終了処理中	DB 接続サービスが既に終了しています、または、終了処理中です	• DB 接続サービスが終了している状態で、該当命令を実行した • DB 接続サービスの終了処理を実行中に該当命令を実行した				○		P.8-28
54013003Hex	DB コネクション名不正	指定した DB コネクション名が DB コネクション設定に設定されていません	• 該当命令の入力変数「DBConnectionName」に指定した DB コネクション名が誤っている • DB コネクション設定の DB コネクション名が誤っている				○		P.8-28
54013004Hex	DB コネクション接続拒否	DB が接続を拒否しました	• DB コネクション設定で設定したユーザ名、または、パスワードが間違っている				○		P.8-29

イベントコード	イベント名	内容	発生要因（推定原因）	重要度					参照先
				全	部	軽	監	般	
54013005Hex	DB コネクション接続失敗	DB に接続できませんでした	• 指定した IP アドレス、または、指定したホスト名のサーバが存在しない • サーバの電源 OFF • サーバの DB の停止 • Ethernet ケーブルのコネクタのはずれ • Ethernet ケーブルの断線				○		P.8-29
54013006Hex	DB コネクション既接続	同一の DB コネクション名で既に DB コネクションを接続しています	• 同一の DB コネクション名で既に DB コネクションを接続している状態で、該当命令を実行した				○		P.8-30
54013007Hex	DB コネクション接続数超過	同時に接続可能な DB コネクション数を超えています	• 同時に接続可能な DB コネクション数が最大の状態で、該当命令を実行した				○		P.8-30
54013008Hex	DB コネクション不正	指定した DB コネクションが不正、または、既に DB コネクションが未接続です	• 該当命令の入力変数「DBConnection」に指定した DB コネクションが誤っている • 該当命令の入力変数「DBConnection」に指定した DB コネクションが未接続である				○		P.8-31
54013009Hex	DB マップ変数不正	指定した DB マップ変数が不正です	• DB マップ変数として、メンバに派生データ型を含む構造体型の変数を指定した • DB マップ変数として構造体型以外の変数を指定した • INSERT 用、UPDATE 用の DB マップ変数として構造体配列変数を指定した				○		P.8-31
5401300AHex	DB マップ変数未登録	指定した DB マップ変数は登録されていません	• DB_CreateMapping 命令で DB マップ変数を作成していない • DB マップ変数として登録した変数でない変数を「MapVar」に指定した • DB_CreateMapping 命令実行時に指定した DB コネクションと異なる DB コネクションを、該当命令で指定した				○		P.8-32

イベントコード	イベント名	内容	発生要因（推定原因）	重要度					参照先
				全	部	軽	監	般	
5401300BHex	SQL 実行エラー	実行した SQL 文が DB でエラーになりました	• DB マップ変数の構造体のメンバ名と同じ名前のカラムが存在しない • DB_CreateMapping 命令で指定したテーブルが DB に存在しない • DB マップ変数の構造体のメンバの値が、カラムのデータ型に変換できない • カラムの値を、DB マップ変数の構造体のメンバのデータ型に変換できない • DB マップ変数の構造体のメンバの値が、カラムのデータ型の範囲を超えている • 抽出条件で指定したカラムが DB のレコードに存在しない（DB_Select 命令、DB_Update 命令、DB_Delete 命令） • 抽出条件の構文の誤り（DB_Select 命令、DB_Update 命令、DB_Delete 命令） • ソート条件で指定したカラムが DB のレコードに存在しない（DB_Select 命令） • ソート条件の構文の誤り（DB_Select 命令） • テーブルへのアクセス権限がない				○		P.8-33
5401300CHex	スプール容量超過	スプールの最大容量を超えたため SQL 文を保留できませんでした	• ネットワーク障害などにより DB に接続できない状態が続いている • スプールに保留した SQL 文の再送処理を実行していない（スプールの再送設定が“手動再送”のとき）				○		P.8-34
5401300EHex	抽出条件不正	入力した抽出条件が不正です	• 入力変数「Where」に NULL 文字（16#00）だけの文字列を指定した				○		P.8-35
54013010Hex	ログコード範囲外	入力したログコードの値が有効範囲外です	• ログコードに 0～9999 以外を指定した				○		P.8-35
54013011Hex	DB コネクション異常切断発生中	DB コネクションが異常切断しているため実行できませんでした	• サーバの電源 OFF • サーバの DB の停止 • Ethernet ケーブルのコネクタのはずれ • Ethernet ケーブルの断線 • ノイズ				○		P.8-36
54013012Hex	DB 接続命令実行タイムアウト	指定した命令実行タイムアウト時間内に命令が終了しませんでした	• サーバの電源 OFF • Ethernet ケーブルのコネクタのはずれ • Ethernet ケーブルの断線 • サーバの処理時間が長い				○		P.8-36
54013013Hex	DB 接続サービス異常停止中	DB 接続サービスが異常停止中のため実行できませんでした	• DB 接続設定が壊れている				○		P.8-37
54013014Hex	スプールデータあり	既にスプールに保留した SQL 文が存在します	• 既にスプールに保留した SQL 文が存在する状態で、DB_Insert 命令、DB_Update 命令を実行した • 既にスプールに保留した SQL 文が存在する状態で、DB_Select 命令、DB_Delete 命令を実行した				○		P.8-37

イベントコード	イベント名	内容	発生要因（推定原因）	重要度					参照先
				全	部	軽	監	般	
54013015Hex	DB 接続サービス初期処理中	DB 接続サービスが初期処理中のため実行できませんでした	• DB 接続サービスの初期処理中に該当命令を実行した				○		P.8-38
54013016Hex	DB 処理中	サーバ側 DB が処理中のため実行できませんでした	• 前回実行した命令で DB 接続命令実行タイムアウトが発生したが、サーバ側 DB の処理が完了していない状態で該当命令を実行した				○		P.8-38
54013017Hex	稼働ログ無効	指定した稼働ログが無効のためログを記録できませんでした	• 入力変数「LogType」で実行ログを指定したが、実行ログが無効である • 入力変数「LogType」でデバッグログを指定したが、デバッグログが記録停止中である				○		P.8-39
54013018Hex	プロシージャハンドル不正	指定したプロシージャハンドルが不正です	• 該当命令の入力変数「ProcHandle」に指定したプロシージャハンドルが誤っている				○		P.8-39
54013019Hex	対象外データベース種別への命令実行	本命令の対象となっていないデータベース種別を指定して命令が実行されました	• DB コネクション設定で設定したデータベース種別が該当命令ではサポート対象外				○		P.8-40
5401301AHex	ストアドプロシージャ名不正	指定したストアドプロシージャ名が存在しません	• 該当命令の入力変数「ProcName」で指定された名前のストアドプロシージャがサーバ側 DB に存在しない （注）指定したストアドプロシージャ名が DB 上に見つからない場合も含みます。ストアドプロシージャが存在する場合でも、ストアドプロシージャへのアクセス権限がないなどの理由により、DB 接続サービスがストアドプロシージャ名を見つけられないことがあります				○		P.8-40
5401301BHex	ストアドプロシージャ引数不正	ストアドプロシージャの引数とアタッチした引数の情報が一致しません	• 該当命令の入力変数「ArgIn」、「ArgOut」、「ArgInOut」とサーバ側 DB から取得するストアドプロシージャの引数情報の名前、数、型が一致しない				○		P.8-41
5401301CHex	ストアドプロシージャ結果セットカラム数不正	ストアドプロシージャの結果セットのカラム数が格納先の構造体型変数のメンバ数と一致しません	• 該当命令により取得された結果セットのカラム数と格納先の構造体型変数のメンバ数が一致しない				○		P.8-41

イベントコード	イベント名	内容	発生要因（推定原因）	重要度					参照先
				全	部	軽	監	般	
5401301EHex	DB 接続サービス未起動	DB 接続サービスが［使用しない］設定です	• DB 接続サービスを［使用しない］設定で該当命令を実行した				○		P.8-42

8-2-2 異常詳細

コントローラ異常説明の見方

各異常の説明で使用する表の各項目の意味を［ ］内に示します。

<table>
<tr><td>イベント名</td><td colspan="2">［異常の名称です］</td><td>イベントコード</td><td colspan="2">［異常のコードです］</td></tr>
<tr><td>内容</td><td colspan="5">［異常の内容です］</td></tr>
<tr><td>発生源</td><td colspan="2">［異常が発生した箇所です］</td><td>発生源詳細</td><td>［発生源の詳細です］</td><td>［異常検出のタイミングです］</td></tr>
<tr><td rowspan="2">異常の属性</td><td>重要度</td><td>［制御に対する影響の大きさです］*1</td><td>復旧方法</td><td>［異常の要因を除去した後に、状態を正常に戻すための方法です。］</td><td>ログカテゴリ</td><td>［保存されるログの種類です］*2</td></tr>
<tr><td>発生の影響</td><td>ユーザプログラム</td><td>［ユーザプログラムの実行状態です］*3</td><td>動作</td><td colspan="2">［異常（イベント）発生時の動作に関する特記事項です］</td></tr>
<tr><td rowspan="2">システム定義変数</td><td colspan="2">変数名</td><td colspan="2">データ型</td><td>名称</td></tr>
<tr><td colspan="5">［異常を検知するシステム定義変数、異常の影響を受けるシステム定義変数、異常の原因となるシステム定義変数の変数名とデータ型、名称です］</td></tr>
<tr><td rowspan="2">発生要因とその処置</td><td colspan="2">発生要因（推定原因）</td><td colspan="2">処置対策</td><td>再発防止</td></tr>
<tr><td colspan="5">［異常の発生要因とその処置対策、再発防止方法です］</td></tr>
<tr><td>付属情報</td><td colspan="5">［Sysmac Studio／表示器で表示される付属情報内容です］</td></tr>
<tr><td>注意点／備考</td><td colspan="5">［その他の注意点、制約事項、補足説明などです。ユーザが選択可能な重要度がある場合は、選択可能な重要度、復旧方法、動作などの関連事項を記載します］</td></tr>
</table>

*1. 次のいずれかです
全停止フォールト：全停止フォールトレベル
部分停止フォールト：部分停止フォールトレベル
軽度フォールト：軽度フォールトレベル
監視情報
一般情報

*2. 次のいずれかです
システム：システムイベントログ
システム：システムイベントログ

*3. 次のいずれかです
継続：ユーザプログラムの実行を継続します
停止：ユーザプログラムの実行を停止します
開始：ユーザプログラムの実行を開始します

DB 接続サービス関連

イベント名	スプール破壊			**イベントコード**	14D00000Hex	
内容	スプールが破壊されています					
発生源	PLC 機能モジュール		**発生源詳細**	DB 接続サービス	**検出タイミング**	DB 接続サービス開始時
異常の属性	**重要度**	軽度フォールト	**復旧方法**	異常解除	**ログカテゴリ**	システム
発生の影響	**ユーザプログラム**	継続	**動作**	影響なし		
システム定義変数	**変数名**		**データ型**		**名称**	
	なし		−		−	
発生要因とその処置	**発生要因（推定原因）**		**処置対策**		**再発防止**	
	ユーザアプリケーションからのスプールへの不正な書込み		スプールの保存先エリアに、ユーザアプリケーションから書込みをしていないか確認してください		スプールの保存先エリアに、ユーザアプリケーションから書込みをしないでください	
	［DB 接続サービス設定］の［運転時のサービス開始］を［使用しない］以外から［使用しない］に変更し、再度［使用しない］以外に変更した		DB 接続サービスの起動時に前回の DB 接続稼働時のスプール設定、およびスプールを破壊しました。DB 接続サービスのサービス開始設定と DB コネクション設定が意図どおりであることを確認してください		なし	
付属情報	なし					
注意点／備考	なし					

イベント名	実行ログ保存失敗			**イベントコード**	14D20000Hex	
内容	実行ログの SD メモリカードへの保存に失敗しました					
発生源	PLC 機能モジュール		**発生源詳細**	DB 接続サービス	**検出タイミング**	常時
異常の属性	**重要度**	軽度フォールト	**復旧方法**	異常解除	**ログカテゴリ**	システム
発生の影響	**ユーザプログラム**	継続	**動作**	影響なし		

システム定義変数	**変数名**	**データ型**	**名称**
	なし	―	―
発生要因とその処置	**発生要因（推定原因）**	**処置対策**	**再発防止**
	SD メモリカードが未装着	SD メモリカードを装着してください	SD メモリカードを装着してください
	SD メモリカードが種別不正	SD または SDHC の SD メモリカードに交換してください	SD または SDHC の SD メモリカードを使用してください
	SD メモリカードのフォーマット不正	SD メモリカードを Sysmac Studio でフォーマットしてください	フォーマットされた SD メモリカードを使用してください。 また、SD BUSY LED 点灯中に SD メモリカードを抜いたり電源断をしたりしないでください
	SD メモリカードがライトプロテクトされている	SD メモリカードのライトプロテクトを解除してください	SD メモリカードのライトプロテクトは解除した状態で使用してください
	SD メモリカードの容量不足	十分な空き容量のある SD メモリカードに交換してください	十分な空き容量のある SD メモリカードに交換してください
	SD メモリカードの故障	SD メモリカードを交換してください	SD BUSY LED 点灯中に SD メモリカードを抜いたり電源断をしたりしないでください。 SD PWR LED 点灯中に SD メモリカードを抜かないでください。 また、書込回数制限にもとづき定期的に SD メモリカードを交換してください

付属情報	付属情報 1：異常の詳細 0001Hex:SD メモリカード未装着 0002Hex:SD メモリカードの故障、フォーマット不正、種別不正 0003Hex:SD メモリカードがライトプロテクトされている 0302Hex:SD メモリカードの容量不足、または、その他の原因による SD メモリカードへのファイルの保存失敗、SD メモリカードの故障
注意点／備考	重要度を監視情報に変更できます。

イベント名	SQL 実行失敗ログ保存失敗			**イベントコード**	14D30000Hex	
内容	SQL 実行失敗ログの SD メモリカードへの保存に失敗しました					
発生源	PLC 機能モジュール		**発生源詳細**	DB 接続サービス	**検出タイミング**	常時
異常の属性	**重要度**	軽度フォールト	**復旧方法**	異常解除	**ログカテゴリ**	システム
発生の影響	**ユーザプログラム**	継続	**動作**	影響なし		

システム定義変数	**変数名**	**データ型**	**名称**
	なし	－	－

発生要因とその処置	**発生要因（推定原因）**	**処置対策**	**再発防止**
	SD メモリカードが未装着	SD メモリカードを装着してください	SD メモリカードを装着してください
	SD メモリカードが種別不正	SD または SDHC の SD メモリカードに交換してください	SD または SDHC の SD メモリカードを使用してください
	SD メモリカードのフォーマット不正	SD メモリカードを Sysmac Studio でフォーマットしてください	フォーマットされた SD メモリカードを使用してください。 また、SD BUSY LED 点灯中に SD メモリカードを抜いたり電源断をしたりしないでください
	SD メモリカードがライトプロテクトされている	SD メモリカードのライトプロテクトを解除してください	SD メモリカードのライトプロテクトは解除した状態で使用してください
	SD メモリカードの容量不足	十分な空き容量のある SD メモリカードに交換してください	十分な空き容量のある SD メモリカードに交換してください
	SD メモリカードの故障	SD メモリカードを交換してください	SD BUSY LED 点灯中に SD メモリカードを抜いたり電源断をしたりしないでください。 SD PWR LED 点灯中に SD メモリカードを抜かないでください。 また、書込回数制限にもとづき定期的に SD メモリカードを交換してください

付属情報	付属情報 1：異常の詳細 0001Hex:SD メモリカード未装着 0002Hex:SD メモリカードの故障、フォーマット不正、種別不正 0003Hex:SD メモリカードがライトプロテクトされている 0302Hex:SD メモリカードの容量不足、または、その他の原因による SD メモリカードへのファイルの保存失敗、SD メモリカードの故障
注意点／備考	重要度を監視情報に変更できます。

イベント名	DB 接続設定エラー			**イベントコード**	35300000Hex	
内容	DB 接続設定が不正です					
発生源	PLC 機能モジュール		**発生源詳細**	DB 接続サービス	**検出タイミング**	ダウンロード時／電源投入時／コントローラリセット時
異常の属性	**重要度**	軽度フォールト	**復旧方法**	自動復旧	**ログカテゴリ**	システム
発生の影響	**ユーザプログラム**	継続	**動作**	DB 接続サービスを開始できません。 DB 接続サービスの稼働状態は“異常停止中”となります		
システム定義変数	**変数名**		**データ型**		**名称**	
	_DBC_Status		_sDBC_STATUS		DB 接続サービスステータス	
発生要因とその処置	**発生要因（推定原因）**		**処置対策**		**再発防止**	
	DB 接続設定のダウンロード中にコントローラを電断した		Sysmac Studio から DB 接続設定を再転送してください		ユーザプログラムまたはコントローラ構成・設定のダウンロード中にコントローラを電断しないでください	
	メモリオールクリア中にコントローラを電断したため DB 接続設定が不正				メモリオールクリア中にコントローラを電断しないでください	
	リストア中にコントローラを電断したため DB 接続設定が不正				リストア中にコントローラを電断しないでください	
	不揮発性メモリの故障		上記の処置を行っても本異常が頻発する場合は、CPU ユニットを交換してください		なし	
付属情報	なし					
注意点／備考	なし					

イベント名	DB サーバ証明書エラー			イベントコード	35310000Hex	
内容	DB サーバ証明書のフォーマットが不正です					
発生源	PLC 機能モジュール		発生源詳細	DB 接続サービス	検出タイミング	ダウンロード時／電源投入時／コントローラリセット時／リストア時
異常の属性	重要度	軽度フォールト	復旧方法	DB 接続設定を再転送後、DB 接続サービスを開始する	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	DB 接続サービスを開始できません。 DB 接続サービスの稼働状態は“異常停止中”となります		
システム定義変数	変数名		データ型		名称	
	_DBC_Status		_sDBC_STATUS		DB 接続サービスステータス	
発生要因とその処置	発生要因（推定原因）		処置対策		再発防止	
	フォーマット(X.509)が不正な DB サーバ証明書をダウンロードした		正しいフォーマットの DB サーバ証明書を設定して Sysmac Studio から DB 接続設定を再転送してください			
	DB 接続設定のダウンロード中にコントローラを電断した		Sysmac Studio から DB 接続設定を再転送してください		ユーザプログラムまたはコントローラ構成・設定のダウンロード中にコントローラを電断しないでください	
	メモリオールクリア中にコントローラを電断したため DB 接続設定が不正				メモリオールクリア中にコントローラを電断しないでください	
	リストア中にコントローラを電断したため DB 接続設定が不正				リストア中にコントローラを電断しないでください	
	不揮発性メモリの故障		上記の処置を行っても本異常が頻発する場合は、CPU ユニットを交換してください		なし	
付属情報	付属情報 1：DB サーバ証明書ファイル名					
注意点／備考	なし					

イベント名	DB 接続システム処理異常			イベントコード	441C0000Hex	
内容	DB 接続サービスで致命的な異常を検知しました					
発生源	PLC 機能モジュール		発生源詳細	DB 接続サービス	検出タイミング	－
異常の属性	重要度	軽度フォールト	復旧方法	電源再投入	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	影響なし		
システム定義変数	変数名		データ型		名称	
	なし		－		－	
発生要因とその処置	発生要因（推定原因）		処置対策		再発防止	
	ソフトウェアエラーが発生した		電源再投入		なし	
付属情報	なし					
注意点／備考	なし					

イベント名	DB コネクション異常切断			イベントコード	85100000Hex	
内容	DB コネクションが異常切断しました					
発生源	PLC 機能モジュール		発生源詳細	DB 接続サービス	検出タイミング	DB 接続命令実行時、および、スプール再送時
異常の属性	重要度	軽度フォールト	復旧方法	自動復旧	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	影響なし		
システム定義変数	変数名		データ型		名称	
	_DBC_Status		_sDBC_STATUS		DB 接続サービスステータス	
発生要因とその処置	発生要因（推定原因）		処置対策		再発防止	
	サーバの電源 OFF		サーバの状態を確認し、正常に起動してください		サーバの状態を確認し、正常に起動してください	
	サーバの DB の停止					
	Ethernet ケーブルのコネクタのはずれ		コネクタが正しく嵌合しているかを確認し、再接続してください		コネクタを確実に接続してください	
	Ethernet ケーブルの断線		Ethernet ケーブルを交換してください		なし	
	ノイズ		ノイズが多い場合はノイズ対策を行ってください		ノイズが多い場合はノイズ対策を行ってください	
付属情報	付属情報 1：DB コネクション名					
注意点／備考	なし					

イベント名	DB 接続サービス開始			イベントコード	95300000Hex	
内容	DB 接続サービスを開始しました					
発生源	PLC 機能モジュール		発生源詳細	DB 接続サービス	検出タイミング	DB 接続サービス開始時
異常の属性	重要度	一般情報	復旧方法	－	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	影響なし		
システム定義変数	変数名		データ型		名称	
	_DBC_Status		_sDBC_STATUS		DB 接続サービスステータス	
発生要因とその処置	発生要因（推定原因）		処置対策		再発防止	
	DB 接続サービスの開始に成功した		－		－	
付属情報	付属情報 1：開始理由 01Hex：DB_ControlService 命令実行、または、Sysmac Studio による操作 02Hex：コントローラの動作モード変更（プログラム→運転）					
注意点／備考	なし					

イベント名	DB 接続サービス停止			イベントコード	95310000Hex	
内容	DB 接続サービスを停止しました					
発生源	PLC 機能モジュール		発生源詳細	DB 接続サービス	検出タイミング	DB 接続サービス停止時
異常の属性	重要度	一般情報	復旧方法	－	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	影響なし		
システム定義変数	変数名		データ型		名称	
	_DBC_Status		_sDBC_STATUS		DB 接続サービスステータス	
発生要因とその処置	発生要因（推定原因）		処置対策		再発防止	
	DB 接続サービスを停止した		－		－	
付属情報	付属情報 1：停止理由 01Hex：DB_ControlService 命令実行、または、Sysmac Studio による操作 02Hex：コントローラの動作モード変更（運転→プログラム） 03Hex：同期（ダウンロード）／メモリオールクリア／リストアの実行 04Hex：全停止フォールトレベルのコントローラ異常発生					
注意点／備考	なし					

イベント名	DB 接続サービス終了			イベントコード	95320000Hex	
内容	DB 接続サービスを終了しました					
発生源	PLC 機能モジュール		発生源詳細	DB 接続サービス	検出タイミング	DB 接続サービス終了時
異常の属性	重要度	一般情報	復旧方法	－	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	影響なし		
システム定義変数	変数名		データ型		名称	
	_DBC_Status		_sDBC_STATUS		DB 接続サービスステータス	
発生要因とその処置	発生要因（推定原因）		処置対策		再発防止	
	DB 接続サービスを終了した		－		－	
付属情報	付属情報 1：終了理由 01Hex：DB_Shutdown 命令実行、または、Sysmac Studio による操作					
注意点／備考	なし					

イベント名	スプールクリア			イベントコード	95330000Hex	
内容	スプールに保留した SQL 文をクリアしました					
発生源	PLC 機能モジュール		発生源詳細	DB 接続サービス	検出タイミング	スプールデータクリア時
異常の属性	重要度	一般情報	復旧方法	－	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	影響なし		
システム定義変数	変数名		データ型		名称	
	なし		－		－	
発生要因とその処置	発生要因（推定原因）		処置対策		再発防止	
	スプールに保留した SQL 文をクリアした		－		－	
付属情報	付属情報 1：DB コネクション名 スプールが DB コネクションに関係なくすべてクリアされた場合は、"-" 付属情報 2：クリア理由 01Hex：DB_ControlSpool 命令実行、または、Sysmac Studio による操作 02Hex：設定したクリア条件に一致 03Hex：自動的にクリアされる条件に一致					
注意点／備考	なし					

イベント名	DB 接続サービス操作（開始）			イベントコード	95340000Hex	
内容	DB 接続サービスの開始操作をしました					
発生源	PLC 機能モジュール		発生源詳細	DB 接続サービス	検出タイミング	DB 接続サービス開始操作時
異常の属性	重要度	一般情報	復旧方法	－	ログカテゴリ	アクセス
発生の影響	ユーザプログラム	継続	動作	影響なし		
システム定義変数	変数名		データ型		名称	
	_DBC_Status		_sDBC_STATUS		DB 接続サービスステータス	
発生要因とその処置	発生要因（推定原因）		処置対策		再発防止	
	SysmacStudio を使用して DB 接続サービスの開始操作をした		－		－	
付属情報	付属情報 1：接続方法 1：USB 直接接続 2：Ethernet 直接接続 3：USB リモート接続または EthernetHUB 接続 付属情報 2：（付属情報 1 が 2 か 3 のときのみ） 接続元の IP アドレス。Proxy 経由の場合は、Proxy の IP アドレス					
注意点／備考	なし					

イベント名	DB 接続サービス操作（停止）			イベントコード	95350000Hex	
内容	DB 接続サービスの停止操作をしました					
発生源	PLC 機能モジュール		発生源詳細	DB 接続サービス	検出タイミング	DB 接続サービス停止操作時
異常の属性	重要度	一般情報	復旧方法	－	ログカテゴリ	アクセス
発生の影響	ユーザプログラム	継続	動作	影響なし		
システム定義変数	変数名		データ型		名称	
	_DBC_Status		_sDBC_STATUS		DB 接続サービスステータス	
発生要因とその処置	発生要因（推定原因）		処置対策		再発防止	
	SysmacStudio を使用して DB 接続サービスの停止操作をした		－		－	
付属情報	付属情報 1：接続方法 1：USB 直接接続 2：Ethernet 直接接続 3：USB リモート接続または EthernetHUB 接続 付属情報 2：（付属情報 1 が 2 か 3 のときのみ） 接続元の IP アドレス。Proxy 経由の場合は、Proxy の IP アドレス					
注意点／備考	なし					

イベント名	DB 接続サービス操作（終了）			イベントコード	95360000Hex	
内容	DB 接続サービスの終了操作をしました					
発生源	PLC 機能モジュール		発生源詳細	DB 接続サービス	検出タイミング	DB 接続サービス終了操作時
異常の属性	重要度	一般情報	復旧方法	－	ログカテゴリ	アクセス
発生の影響	ユーザプログラム	継続	動作	影響なし		
システム定義変数	変数名		データ型		名称	
	_DBC_Status		_sDBC_STATUS		DB 接続サービスステータス	
発生要因とその処置	発生要因（推定原因）		処置対策		再発防止	
	SysmacStudio を使用して DB 接続サービスの終了操作をした		－		－	
付属情報	付属情報 1：接続方法 1：USB 直接接続 2：Ethernet 直接接続 3：USB リモート接続または EthernetHUB 接続 付属情報 2：（付属情報 1 が 2 か 3 のときのみ） 接続元の IP アドレス。Proxy 経由の場合は、Proxy の IP アドレス					
注意点／備考	なし					

イベント名	スプールクリア操作			イベントコード	95370000Hex	
内容	スプールに保留した SQL 文のクリア操作をしました					
発生源	PLC 機能モジュール		発生源詳細	DB 接続サービス	検出タイミング	スプールクリア操作時
異常の属性	重要度	一般情報	復旧方法	－	ログカテゴリ	アクセス
発生の影響	ユーザプログラム	継続	動作	影響なし		
システム定義変数	変数名		データ型		名称	
	なし		－		－	
発生要因とその処置	発生要因（推定原因）		処置対策		再発防止	
	SysmacStudio を使用してスプールに保留した SQL 文のクリア操作をした。		－		－	
付属情報	付属情報 1：接続方法 1：USB 直接接続 2：Ethernet 直接接続 3：USB リモート接続または EthernetHUB 接続 付属情報 2：（付属情報 1 が 2 か 3 のときのみ） 接続元の IP アドレス。Proxy 経由の場合は、Proxy の IP アドレス					
注意点／備考	なし					

イベント名	稼働ログのクリア操作			イベントコード	95380000Hex	
内容	稼働ログのクリア操作をしました					
発生源	PLC 機能モジュール		発生源詳細	DB 接続サービス	検出タイミング	稼働ログのクリア操作時
異常の属性	重要度	一般情報	復旧方法	－	ログカテゴリ	アクセス
発生の影響	ユーザプログラム	継続	動作	影響なし		
システム定義変数	変数名		データ型		名称	
	なし		－		－	
発生要因とその処置	発生要因（推定原因）		処置対策		再発防止	
	SysmacStudio を使用して稼働ログのクリア操作をした		－		－	
付属情報	付属情報 1：接続方法 1：USB 直接接続 2：Ethernet 直接接続 3：USB リモート接続または EthernetHUB 接続 付属情報 2：（付属情報 1 が 2 か 3 のときのみ） 接続元の IP アドレス。Proxy 経由の場合は、Proxy の IP アドレス					
注意点／備考	なし					

イベント名	デバッグログ操作（開始）			イベントコード	95390000Hex	
内容	デバッグログの開始操作をしました					
発生源	PLC 機能モジュール		発生源詳細	DB 接続サービス	検出タイミング	デバッグログの開始操作時
異常の属性	重要度	一般情報	復旧方法	－	ログカテゴリ	アクセス
発生の影響	ユーザプログラム	継続	動作	影響なし		
システム定義変数	変数名		データ型		名称	
	なし		－		－	
発生要因とその処置	発生要因（推定原因）		処置対策		再発防止	
	SysmacStudio を使用してデバッグログの開始操作をした		－		－	
付属情報	付属情報 1：接続方法 1：USB 直接接続 2：Ethernet 直接接続 3：USB リモート接続または EthernetHUB 接続 付属情報 2：（付属情報 1 が 2 か 3 のときのみ） 接続元の IP アドレス。Proxy 経由の場合は、Proxy の IP アドレス					
注意点／備考	なし					

イベント名	デバッグログ操作（停止）			イベントコード	953A0000Hex	
内容	デバッグログの停止操作をしました					
発生源	PLC 機能モジュール		発生源詳細	DB 接続サービス	検出タイミング	デバッグログの停止操作時
異常の属性	重要度	一般情報	復旧方法	－	ログカテゴリ	アクセス
発生の影響	ユーザプログラム	継続	動作	影響なし		
システム定義変数	変数名		データ型		名称	
	なし		－		－	
発生要因とその処置	発生要因（推定原因）		処置対策		再発防止	
	SysmacStudio を使用してデバッグログの停止操作をした		－		－	
付属情報	付属情報 1：接続方法 1：USB 直接接続 2：Ethernet 直接接続 3：USB リモート接続または EthernetHUB 接続 付属情報 2：（付属情報 1 が 2 か 3 のときのみ） 接続元の IP アドレス。Proxy 経由の場合は、Proxy の IP アドレス					
注意点／備考	なし					

DB 接続命令関連

イベント名	DB 接続サービス未開始			**イベントコード**	54013000Hex	
内容	DB 接続サービスが開始していません					
発生源	PLC 機能モジュール		**発生源詳細**	命令	**検出タイミング**	命令実行時
異常の属性	**重要度**	監視情報	**復旧方法**	－	**ログカテゴリ**	システム
発生の影響	**ユーザプログラム**	継続	**動作**	該当命令を仕様どおりに終了		
システム定義変数	**変数名**		**データ型**		**名称**	
	なし		－		－	
発生要因とその処置	**発生要因（推定原因）**		**処置対策**		**再発防止**	
	該当命令を実行する前に、DB 接続サービスの開始指示をしていない		DB 接続サービスを開始してください。または、該当命令が DB 接続サービス動作中に実行するようにユーザプログラムを修正してください		該当命令が DB 接続サービス動作中に実行するようにユーザプログラムを作成してください	
	該当命令を実行する前に、DB 接続サービスの停止指示をした					
付属情報	付属情報 1：異常の発生箇所 付属情報 2：異常の発生箇所詳細　回路番号。プログラムセクションの場合はセクション先頭からの回路番号。ST の場合は行番号 付属情報 3：異常が発生した命令の命令名、命令インスタンス名。候補が複数ある場合は複数表示。特定できない場合は表示なし 付属情報 4：拡張エラーコード(ErrorIDEx)					
注意点／備考	なし					

<table>
<tr><td>イベント名</td><td colspan="2">DB 接続サービス動作モード変更失敗</td><td>イベントコード</td><td colspan="2">54013001Hex</td></tr>
<tr><td>内容</td><td colspan="5">DB 接続サービスの動作モードを変更できませんでした</td></tr>
<tr><td>発生源</td><td colspan="2">PLC 機能モジュール</td><td>発生源詳細</td><td>命令</td><td>検出タイミング</td><td>命令実行時</td></tr>
<tr><td>異常の属性</td><td>重要度</td><td>監視情報</td><td>復旧方法</td><td>－</td><td>ログカテゴリ</td><td>システム</td></tr>
<tr><td>発生の影響</td><td>ユーザプログラム</td><td>継続</td><td>動作</td><td colspan="3">該当命令を仕様どおりに終了</td></tr>
<tr><td rowspan="2">システム定義変数</td><td colspan="2">変数名</td><td colspan="2">データ型</td><td colspan="2">名称</td></tr>
<tr><td colspan="2">なし</td><td colspan="2">－</td><td colspan="2">－</td></tr>
<tr><td rowspan="5">発生要因とその処置</td><td colspan="2">発生要因（推定原因）</td><td colspan="2">処置対策</td><td colspan="2">再発防止</td></tr>
<tr><td colspan="2">“稼働モード”で動作中に、該当命令で“テストモード”への動作モード変更を実行した</td><td colspan="2" rowspan="2">DB 接続サービスを停止してから、該当命令を実行してください。または、該当命令が DB 接続サービスの状態が“アイドル”のときに実行するようにユーザプログラムを修正してください。</td><td colspan="2" rowspan="2">該当命令が DB 接続サービスの状態が“アイドル”のときに実行するようにユーザプログラムを作成してください</td></tr>
<tr><td colspan="2">“テストモード”で動作中に、該当命令で“稼働モード”への動作モード変更を実行した</td></tr>
<tr><td colspan="2">DB 接続サービスが停止処理中に、DB 接続サービスの開始を指示した</td><td colspan="2" rowspan="2">しばらくしてから、該当命令を実行してください。</td><td colspan="2" rowspan="2">DB_Insert 命令、DB_Update 命令、DB_Select 命令、DB_Delete 命令が実行中の間は、DB 接続サービスの停止を指示しても、DB 接続サービスは停止処理中となります。
DB 接続サービスの停止は、DB_Insert 命令、DB_Update 命令、DB_Select 命令、DB_Delete 命令の実行が完了してから、指示してください。</td></tr>
<tr><td colspan="2">DB 接続サービスの停止処理中に、DB 接続サービスのシャットダウンを指示した</td></tr>
<tr><td>付属情報</td><td colspan="5">付属情報 1：異常の発生箇所
付属情報 2：異常の発生箇所詳細　回路番号。プログラムセクションの場合はセクション先頭からの回路番号。ST の場合は行番号
付属情報 3：異常が発生した命令の命令名、命令インスタンス名。候補が複数ある場合は複数表示。特定できない場合は表示なし
付属情報 4：拡張エラーコード(ErrorIDEx)</td></tr>
<tr><td>注意点／備考</td><td colspan="5">なし</td></tr>
</table>

イベント名	DB 接続サービス終了／終了処理中			イベントコード	54013002Hex	
内容	DB 接続サービスが既に終了しています、または、終了処理中です					
発生源	PLC 機能モジュール		発生源詳細	命令	検出タイミング	命令実行時
異常の属性	重要度	監視情報	復旧方法	－	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	該当命令を仕様どおりに終了		
システム定義変数	変数名		データ型		名称	
	なし		－		－	
発生要因とその処置	発生要因（推定原因）		処置対策		再発防止	
	DB 接続サービスが終了している状態で該当命令を実行した		コントローラの電源を再投入し、DB 接続サービスを開始してから、該当命令を実行してください		DB_Shutdown 命令実行後は、該当命令は実行しないようにユーザプログラムを作成してください。または、Sysmac Studio でシャットダウンを指示した後は、該当命令は実行しないようにユーザプログラムを作成してください。	
	DB 接続サービスの終了処理を実行中に該当命令を実行した					
付属情報	付属情報 1：異常の発生箇所 付属情報 2：異常の発生箇所詳細　回路番号。プログラムセクションの場合はセクション先頭からの回路番号。ST の場合は行番号 付属情報 3：異常が発生した命令の命令名、命令インスタンス名。候補が複数ある場合は複数表示。特定できない場合は表示なし 付属情報 4：拡張エラーコード(ErrorIDEx)					
注意点／備考	なし					

イベント名	DB コネクション名不正			イベントコード	54013003Hex	
内容	指定した DB コネクション名が DB コネクション設定に設定されていません					
発生源	PLC 機能モジュール		発生源詳細	命令	検出タイミング	命令実行時
異常の属性	重要度	監視情報	復旧方法	－	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	該当命令を仕様どおりに終了		
システム定義変数	変数名		データ型		名称	
	なし		－		－	
発生要因とその処置	発生要因（推定原因）		処置対策		再発防止	
	該当命令の入力変数「DBConnectionName」に指定した DB コネクション名が誤っている		該当命令の入力変数「DBConnectionName」に正しい DB コネクション名を指定してください		該当命令の入力変数「DBConnectionName」の DB コネクション名が正しく指定されていることを確認してください	
	DB コネクション設定の DB コネクション名が誤っている		DB コネクション設定に正しい DB コネクション名を指定してください		DB コネクション設定の DB コネクション名が正しく設定されていることを確認してください	
付属情報	付属情報 1：異常の発生箇所 付属情報 2：異常の発生箇所詳細　回路番号。プログラムセクションの場合はセクション先頭からの回路番号。ST の場合は行番号 付属情報 3：異常が発生した命令の命令名、命令インスタンス名。候補が複数ある場合は複数表示。特定できない場合は表示なし 付属情報 4：拡張エラーコード(ErrorIDEx)					
注意点／備考	なし					

イベント名	DB コネクション接続拒否			イベントコード	54013004Hex	
内容	DB が接続を拒否しました					
発生源	PLC 機能モジュール		発生源詳細	命令	検出タイミング	命令実行時
異常の属性	重要度	監視情報	復旧方法	−	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	該当命令を仕様どおりに終了		
システム定義変数	変数名		データ型		名称	
	なし		−		−	
発生要因とその処置	発生要因（推定原因）		処置対策		再発防止	
	DB コネクション設定で設定したユーザ名、または、パスワードが間違っている		DB コネクション設定に正しいユーザ名、パスワードを入力してください		DB コネクション設定に正しいユーザ名、パスワードを入力してください	
付属情報	付属情報 1：異常の発生箇所 付属情報 2：異常の発生箇所詳細　回路番号。プログラムセクションの場合はセクション先頭からの回路番号。ST の場合は行番号 付属情報 3：異常が発生した命令の命令名、命令インスタンス名。候補が複数ある場合は複数表示。特定できない場合は表示なし 付属情報 4：拡張エラーコード(ErrorIDEx)					
注意点／備考	なし					

イベント名	DB コネクション接続失敗			イベントコード	54013005Hex	
内容	DB に接続できませんでした					
発生源	PLC 機能モジュール		発生源詳細	命令	検出タイミング	命令実行時
異常の属性	重要度	監視情報	復旧方法	−	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	該当命令を仕様どおりに終了		
システム定義変数	変数名		データ型		名称	
	なし		−		−	
発生要因とその処置	発生要因（推定原因）		処置対策		再発防止	
	指定した IP アドレス、または、指定したホスト名のサーバが存在しない		DB コネクション設定に正しい IP アドレス、または、ホスト名を入力してください		DB コネクション設定に正しい IP アドレス、または、ホスト名を入力してください	
	サーバの電源 OFF		サーバの状態を確認し、正常に起動してください		サーバの状態を確認し、正常に起動してください	
	サーバの DB の停止					
	Ethernet ケーブルのコネクタのはずれ		コネクタが正しく嵌合しているかを確認し、再接続してください		コネクタを確実に接続してください	
	Ethernet ケーブルの断線		Ethernet ケーブルを交換してください		なし	
付属情報	付属情報 1：異常の発生箇所 付属情報 2：異常の発生箇所詳細　回路番号。プログラムセクションの場合はセクション先頭からの回路番号。ST の場合は行番号 付属情報 3：異常が発生した命令の命令名、命令インスタンス名。候補が複数ある場合は複数表示。特定できない場合は表示なし 付属情報 4：拡張エラーコード(ErrorIDEx)					
注意点／備考	なし					

イベント名	DB コネクション既接続			イベントコード	54013006Hex	
内容	同一の DB コネクション名で既に DB コネクションを接続しています					
発生源	PLC 機能モジュール		発生源詳細	命令	検出タイミング	命令実行時
異常の属性	重要度	監視情報	復旧方法	−	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	該当命令を仕様どおりに終了		
システム定義変数	変数名		データ型		名称	
	なし		−		−	
発生要因とその処置	発生要因（推定原因）		処置対策		再発防止	
	同一の DB コネクション名で既に DB コネクションを接続している状態で、該当命令を実行した		該当命令が DB コネクションが未接続状態で実行するようにユーザプログラムを修正してください		該当命令が DB コネクションが未接続状態で実行するようにユーザプログラムを作成してください	
付属情報	付属情報 1：異常の発生箇所 付属情報 2：異常の発生箇所詳細　回路番号。プログラムセクションの場合はセクション先頭からの回路番号。ST の場合は行番号 付属情報 3：異常が発生した命令の命令名、命令インスタンス名。候補が複数ある場合は複数表示。特定できない場合は表示なし 付属情報 4：拡張エラーコード(ErrorIDEx)					
注意点／備考	なし					

イベント名	DB コネクション接続数超過			イベントコード	54013007Hex	
内容	同時に接続可能な DB コネクション数を超えています					
発生源	PLC 機能モジュール		発生源詳細	命令	検出タイミング	命令実行時
異常の属性	重要度	監視情報	復旧方法	−	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	該当命令を仕様どおりに終了		
システム定義変数	変数名		データ型		名称	
	なし		−		−	
発生要因とその処置	発生要因（推定原因）		処置対策		再発防止	
	同時に接続可能な DB コネクション数が最大の状態で、該当命令を実行した		同時に接続可能な DB コネクション数の最大を超えない範囲で DB コネクションを接続するようにユーザプログラムを修正してください		同時に接続可能な DB コネクション数の最大を超えない範囲で DB コネクションを接続するようにユーザプログラムを作成してください	
付属情報	付属情報 1：異常の発生箇所 付属情報 2：異常の発生箇所詳細　回路番号。プログラムセクションの場合はセクション先頭からの回路番号。ST の場合は行番号 付属情報 3：異常が発生した命令の命令名、命令インスタンス名。候補が複数ある場合は複数表示。特定できない場合は表示なし 付属情報 4：拡張エラーコード(ErrorIDEx)					
注意点／備考	なし					

イベント名	DB コネクション不正			イベントコード	54013008Hex	
内容	指定した DB コネクションが不正、または、既に DB コネクションが未接続です					
発生源	PLC 機能モジュール		発生源詳細	命令	検出タイミング	命令実行時
異常の属性	重要度	監視情報	復旧方法	−	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	該当命令を仕様どおりに終了		
システム定義変数	変数名		データ型		名称	
	なし		−		−	
発生要因とその処置	発生要因（推定原因）		処置対策		再発防止	
	該当命令の入力変数「DBConnection」に指定した DB コネクションが誤っている		該当命令の入力変数「DBConnection」に正しい DB コネクションを指定してください		該当命令の入力変数「DBConnection」に正しい DB コネクションが指定されていることを確認してください	
	該当命令の入力変数「DBConnection」に指定した DB コネクションが未接続である		DB_Connect 命令で DB コネクションを接続してから、該当命令を実行するようにユーザプログラムを修正してください		DB_Connect 命令で DB コネクションを接続してから、該当命令を実行するようにユーザプログラムを作成してください	
付属情報	付属情報 1：異常の発生箇所 付属情報 2：異常の発生箇所詳細　回路番号。プログラムセクションの場合はセクション先頭からの回路番号。ST の場合は行番号 付属情報 3：異常が発生した命令の命令名、命令インスタンス名。候補が複数ある場合は複数表示。特定できない場合は表示なし 付属情報 4：拡張エラーコード(ErrorIDEx)					
注意点／備考	なし					

イベント名	DB マップ変数不正			イベントコード	54013009Hex	
内容	指定した DB マップ変数が不正です					
発生源	PLC 機能モジュール		発生源詳細	命令	検出タイミング	命令実行時
異常の属性	重要度	監視情報	復旧方法	−	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	該当命令を仕様どおりに終了		
システム定義変数	変数名		データ型		名称	
	なし		−		−	
発生要因とその処置	発生要因（推定原因）		処置対策		再発防止	
	DB マップ変数として、メンバに派生データ型を含む構造体型の変数を指定した		DB マップ変数で使用する構造体型のメンバは、基本データ型を指定してください		DB マップ変数として指定する変数のデータ型を確認してから、ユーザプログラムを作成してください	
	DB マップ変数として構造体型以外の変数を指定した		DB マップ変数は構造体型の変数を指定してください			
	INSERT 用、UPDATE 用の DB マップ変数として構造体配列変数を指定した		INSERT 用、UPDATE 用の DB マップ変数は、構造体変数を指定してください			
付属情報	付属情報 1：異常の発生箇所 付属情報 2：異常の発生箇所詳細　回路番号。プログラムセクションの場合はセクション先頭からの回路番号。ST の場合は行番号 付属情報 3：異常が発生した命令の命令名、命令インスタンス名。候補が複数ある場合は複数表示。特定できない場合は表示なし 付属情報 4：拡張エラーコード(ErrorIDEx)					
注意点／備考	なし					

<table>
<tr><td>イベント名</td><td colspan="2">DB マップ変数未登録</td><td>イベントコード</td><td colspan="2">5401300AHex</td></tr>
<tr><td>内容</td><td colspan="5">指定した DB マップ変数は登録されていません</td></tr>
<tr><td>発生源</td><td>PLC 機能モジュール</td><td>発生源詳細</td><td>命令</td><td>検出タイミング</td><td>命令実行時</td></tr>
<tr><td>異常の属性</td><td>重要度</td><td>監視情報</td><td>復旧方法</td><td>－</td><td>ログカテゴリ</td><td>システム</td></tr>
<tr><td>発生の影響</td><td>ユーザプログラム</td><td>継続</td><td>動作</td><td colspan="3">該当命令を仕様どおりに終了</td></tr>
<tr><td>システム定義変数</td><td colspan="2">変数名</td><td colspan="2">データ型</td><td>名称</td></tr>
<tr><td></td><td colspan="2">なし</td><td colspan="2">－</td><td>－</td></tr>
<tr><td>発生要因とその処置</td><td colspan="2">発生要因（推定原因）</td><td colspan="2">処置対策</td><td>再発防止</td></tr>
<tr><td></td><td colspan="2">DB_CreateMapping 命令で DB マップ変数を作成していない</td><td colspan="2">DB_CreateMapping 命令で DB マップ変数を作成してから、該当命令を実行するようにユーザプログラムを修正してください</td><td>DB_CreateMapping 命令で DB マップ変数を作成してから、該当命令を実行するようにユーザプログラムを作成してください</td></tr>
<tr><td></td><td colspan="2">DB マップ変数として登録した変数でない変数を「MapVar」に指定した</td><td colspan="2" rowspan="2">該当命令の入力パラメータを確認し、ユーザプログラムを修正してください</td><td rowspan="2">該当命令の入力パラメータには、DB_CreateMapping 命令で指定した DB コネクション、および、DB_CreateMapping 命令で作成した DB マップ変数を指定してください</td></tr>
<tr><td></td><td colspan="2">DB_CreateMapping 命令実行時に指定した DB コネクションと異なる DB コネクションを、該当命令で指定した</td></tr>
<tr><td>付属情報</td><td colspan="5">付属情報 1：異常の発生箇所
付属情報 2：異常の発生箇所詳細　回路番号。プログラムセクションの場合はセクション先頭からの回路番号。ST の場合は行番号
付属情報 3：異常が発生した命令の命令名、命令インスタンス名。候補が複数ある場合は複数表示。特定できない場合は表示なし
付属情報 4：拡張エラーコード(ErrorIDEx)</td></tr>
<tr><td>注意点／備考</td><td colspan="5">なし</td></tr>
</table>

イベント名	SQL 実行エラー			**イベントコード**	5401300BHex	
内容	実行した SQL 文が DB でエラーになりました					
発生源	PLC 機能モジュール		**発生源詳細**	命令	**検出タイミング**	命令実行時
異常の属性	**重要度**	監視情報	**復旧方法**	－	**ログカテゴリ**	システム
発生の影響	**ユーザプログラム**	継続	**動作**	該当命令を仕様どおりに終了		

システム定義変数	**変数名**	**データ型**	**名称**
	なし	－	－
発生要因とその処置	**発生要因（推定原因）**	**処置対策**	**再発防止**
	DB マップ変数の構造体のメンバ名と同じ名前のカラムが存在しない	DB マップ変数の構造体のメンバ名とカラムの名称が合っているかを確認してください	DB マップ変数の構造体のメンバ名とカラムの名称が合っているかを確認してください
	DB_CreateMapping 命令で指定したテーブルが DB に存在しない	DB_CreateMapping 命令で指定したテーブル名が正しいかを確認してください	DB_CreateMapping 命令で指定したテーブル名が正しいかを確認してください
	DB マップ変数の構造体のメンバの値が、カラムのデータ型に変換できない	DB マップ変数の構造体のメンバのデータ型と、カラムのデータ型が変換可能なデータ型かを確認してください	DB マップ変数の構造体のメンバのデータ型と、カラムのデータ型が変換可能なデータ型かを確認してください
	カラムの値を、DB マップ変数の構造体のメンバのデータ型に変換できない	DB マップ変数の構造体のメンバのデータ型と、カラムのデータ型が変換可能なデータ型かを確認してください。または、マッピングされているカラムの値が NULL でないことを確認してください。	DB マップ変数の構造体のメンバのデータ型と、カラムのデータ型が変換可能なデータ型かを確認してください。または、値が NULL となるカラムをマッピングしないよう構造体のメンバを定義してください。
	DB マップ変数の構造体のメンバの値が、カラムのデータ型の範囲を超えている	DB マップ変数の構造体のメンバの値を確認してください	DB マップ変数の構造体のメンバの値がカラムのデータ型の範囲内になるようにユーザプログラムを作成してください
	抽出条件で指定したカラムが DB のレコードに存在しない（DB_Select 命令、DB_Update 命令、DB_Delete 命令）	抽出条件で指定したカラム名が正しいかを確認してください。または、抽出条件の構文が正しいかを確認してください	抽出条件で指定したカラム名が正しいかを確認してください。または、抽出条件の構文が正しいかを確認してください。
	抽出条件の構文の誤り（DB_Select 命令、DB_Update 命令、DB_Delete 命令）		
	ソート条件で指定したカラムが DB のレコードに存在しない（DB_Select 命令）	ソート条件で指定したカラム名が正しいかを確認してください。または、ソート条件の構文が正しいかを確認してください	ソート条件で指定したカラム名が正しいかを確認してください。または、ソート条件の構文が正しいかを確認してください。
	ソート条件の構文の誤り（DB_Select 命令）		
	テーブルへのアクセス権限がない	テーブルのアクセス権限を確認してください	テーブルのアクセス権限を確認してください
付属情報	付属情報 1：異常の発生箇所 付属情報 2：異常の発生箇所詳細　回路番号。プログラムセクションの場合はセクション先頭からの回路番号。ST の場合は行番号 付属情報 3：異常が発生した命令の命令名、命令インスタンス名。候補が複数ある場合は複数表示。特定できない場合は表示なし 付属情報 4：拡張エラーコード(ErrorIDEx)					
注意点／備考	なし					

イベント名	スプール容量超過			**イベントコード**	5401300CHex	
内容	スプールの最大容量を超えたため SQL 文を保留できませんでした					
発生源	PLC 機能モジュール		**発生源詳細**	命令	**検出タイミング**	命令実行時
異常の属性	**重要度**	監視情報	**復旧方法**	－	**ログカテゴリ**	システム
発生の影響	**ユーザプログラム**	継続	**動作**	該当命令を仕様どおりに終了		
システム定義変数	**変数名**		**データ型**		**名称**	
	なし		－		－	
発生要因とその処置	**発生要因（推定原因）**		**処置対策**		**再発防止**	
	ネットワーク障害などにより DB に接続できない状態が続いている		ネットワーク障害を取り除いてください		DB_GetConnectionStatus 命令でスプール使用率を参照し、スプール使用率が一定量を超えた場合、DB_INSERT 命令、および、DB_UPDATE 命令は実行しないように、ユーザプログラムで制御してください。 または、 DB_GetConnectionStatus 命令で DB コネクション状態を参照し、“接続中”に遷移した場合、DB_ControlSpool 命令でスプールに保留した SQL 文を再送してください	
	スプールに保留した SQL 文の再送処理を実行していない（スプールの再送設定が“手動再送”のとき）		DB コネクションが再接続した状態で DB_ControlSpool 命令でスプールに保留した SQL 文を再送してください		DB_GetConnectionStatus 命令で DB コネクション状態を参照し、“接続中”に遷移した場合、DB_ControlSpool 命令でスプールに保留した SQL 文を再送してください	
付属情報	付属情報 1：異常の発生箇所 付属情報 2：異常の発生箇所詳細　回路番号。プログラムセクションの場合はセクション先頭からの回路番号。ST の場合は行番号 付属情報 3：異常が発生した命令の命令名、命令インスタンス名。候補が複数ある場合は複数表示。特定できない場合は表示なし 付属情報 4：拡張エラーコード(ErrorIDEx)					
注意点／備考	なし					

イベント名	抽出条件不正			イベントコード	5401300EHex	
内容	入力した抽出条件が不正です					
発生源	PLC 機能モジュール		発生源詳細	命令	検出タイミング	命令実行時
異常の属性	重要度	監視情報	復旧方法	−	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	該当命令を仕様どおりに終了		
システム定義変数	変数名		データ型		名称	
	なし		−		−	
発生要因とその処置	発生要因（推定原因）		処置対策		再発防止	
	入力変数「Where」に NULL 文字（16#00）だけの文字列を指定した		入力変数「Where」に抽出条件を表す文字列を指定してください		入力変数「Where」に抽出条件を表す文字列を指定してください	
付属情報	付属情報 1：異常の発生箇所 付属情報 2：異常の発生箇所詳細　回路番号。プログラムセクションの場合はセクション先頭からの回路番号。ST の場合は行番号 付属情報 3：異常が発生した命令の命令名、命令インスタンス名。候補が複数ある場合は複数表示。特定できない場合は表示なし 付属情報 4：拡張エラーコード(ErrorIDEx)					
注意点／備考	なし					

イベント名	ログコード範囲外			イベントコード	54013010Hex	
内容	入力したログコードの値が有効範囲外です					
発生源	PLC 機能モジュール		発生源詳細	命令	検出タイミング	命令実行時
異常の属性	重要度	監視情報	復旧方法	−	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	該当命令を仕様どおりに終了		
システム定義変数	変数名		データ型		名称	
	なし		−		−	
発生要因とその処置	発生要因（推定原因）		処置対策		再発防止	
	ログコードに 0～9999 以外を指定した		ログコードが 0～9999 の範囲になるようにユーザプログラムを修正してください		ログコードが 0～9999 の範囲になるようにユーザプログラムを作成してください	
付属情報	付属情報 1：異常の発生箇所 付属情報 2：異常の発生箇所詳細　回路番号。プログラムセクションの場合はセクション先頭からの回路番号。ST の場合は行番号 付属情報 3：異常が発生した命令の命令名、命令インスタンス名。候補が複数ある場合は複数表示。特定できない場合は表示なし 付属情報 4：拡張エラーコード(ErrorIDEx)					
注意点／備考	なし					

イベント名	DB コネクション異常切断発生中			イベントコード	54013011Hex	
内容	DB コネクションが異常切断しているため実行できませんでした					
発生源	PLC 機能モジュール		発生源詳細	命令	検出タイミング	命令実行時
異常の属性	重要度	命令実行時	復旧方法	−	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	該当命令を仕様どおりに終了		

システム定義変数	変数名	データ型	名称
	なし	−	−

発生要因とその処置	発生要因（推定原因）	処置対策	再発防止
	サーバの電源 OFF	サーバの状態を確認し、正常に起動してください	サーバの状態を確認し、正常に起動してください
	サーバの DB の停止		
	Ethernet ケーブルのコネクタのはずれ	コネクタが正しく嵌合しているかを確認し、再接続してください	コネクタを確実に接続してください
	Ethernet ケーブルの断線	Ethernet ケーブルを交換してください	なし
	ノイズ	ノイズが多い場合はノイズ対策を行ってください	ノイズが多い場合はノイズ対策を行ってください

付属情報	付属情報 1：異常の発生箇所 付属情報 2：異常の発生箇所詳細　回路番号。プログラムセクションの場合はセクション先頭からの回路番号。ST の場合は行番号 付属情報 3：異常が発生した命令の命令名、命令インスタンス名。候補が複数ある場合は複数表示。特定できない場合は表示なし 付属情報 4：拡張エラーコード(ErrorIDEx)
注意点／備考	なし

イベント名	DB 接続命令実行タイムアウト			イベントコード	54013012Hex	
内容	指定した命令実行タイムアウト時間内に命令が終了しませんでした					
発生源	PLC 機能モジュール		発生源詳細	命令	検出タイミング	命令実行時
異常の属性	重要度	監視情報	復旧方法	−	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	該当命令を仕様どおりに終了		

システム定義変数	変数名	データ型	名称
	なし	−	−

発生要因とその処置	発生要因（推定原因）	処置対策	再発防止
	サーバの電源 OFF	サーバの状態を確認し、正常に起動してください	サーバの状態を確認し、正常に起動してください
	Ethernet ケーブルのコネクタのはずれ	コネクタが正しく嵌合しているかを確認し、再接続してください	コネクタを確実に接続してください
	Ethernet ケーブルの断線	Ethernet ケーブルを交換してください	なし
	サーバの処理時間が長い	デバッグログでサーバの応答時間を確認し、適切なタイムアウト時間に変更してください	デバッグログでサーバの応答時間を確認し、適切なタイムアウト時間を指定してください

付属情報	付属情報 1：異常の発生箇所 付属情報 2：異常の発生箇所詳細　回路番号。プログラムセクションの場合はセクション先頭からの回路番号。ST の場合は行番号 付属情報 3：異常が発生した命令の命令名、命令インスタンス名。候補が複数ある場合は複数表示。特定できない場合は表示なし 付属情報 4：拡張エラーコード(ErrorIDEx)
注意点／備考	なし

イベント名	DB 接続サービス異常停止中			イベントコード	54013013Hex	
内容	DB 接続サービスが異常停止中のため実行できませんでした					
発生源	PLC 機能モジュール		発生源詳細	命令	検出タイミング	命令実行時
異常の属性	重要度	監視情報	復旧方法	－	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	該当命令を仕様どおりに終了		
システム定義変数	変数名		データ型		名称	
	なし		－		－	
発生要因とその処置	発生要因（推定原因）		処置対策		再発防止	
	DB 接続設定が壊れている		Sysmac Studio の同期機能で DB 接続設定を再転送してください		DB 接続設定のダウンロード中のコントローラを電断しないでください	
付属情報	付属情報 1：異常の発生箇所 付属情報 2：異常の発生箇所詳細　回路番号。プログラムセクションの場合はセクション先頭からの回路番号。ST の場合は行番号 付属情報 3：異常が発生した命令の命令名、命令インスタンス名。候補が複数ある場合は複数表示。特定できない場合は表示なし 付属情報 4：拡張エラーコード(ErrorIDEx)					
注意点／備考	なし					

イベント名	スプールデータあり			イベントコード	54013014Hex	
内容	既にスプールに保留した SQL 文が存在します					
発生源	PLC 機能モジュール		発生源詳細	命令	検出タイミング	命令実行時
異常の属性	重要度	監視情報	復旧方法	－	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	該当命令を仕様どおりに終了		
システム定義変数	変数名		データ型		名称	
	なし		－		－	
発生要因とその処置	発生要因（推定原因）		処置対策		再発防止	
	既にスプールに保留した SQL 文が存在する状態で、DB_Insert 命令、DB_Update 命令を実行した		なし		なし	
	既にスプールに保留した SQL 文が存在する状態で、DB_Select 命令、DB_Delete 命令を実行した		スプールに保留されている SQL 文の再送処理が完了してから、再度命令を実行してください		スプールに SQL 文が保留されていない状態で、該当命令を実行してください	
付属情報	付属情報 1：異常の発生箇所 付属情報 2：異常の発生箇所詳細　回路番号。プログラムセクションの場合はセクション先頭からの回路番号。ST の場合は行番号 付属情報 3：異常が発生した命令の命令名、命令インスタンス名。候補が複数ある場合は複数表示。特定できない場合は表示なし 付属情報 4：拡張エラーコード(ErrorIDEx)					
注意点／備考	なし					

イベント名	DB 接続サービス初期処理中			イベントコード	54013015Hex	
内容	DB 接続サービスが初期処理中のため実行できませんでした					
発生源	PLC 機能モジュール		発生源詳細	命令	検出タイミング	命令実行時
異常の属性	重要度	監視情報	復旧方法	－	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	該当命令を仕様どおりに終了		
システム定義変数	変数名		データ型		名称	
	_DBC_Status		_sDBC_STATUS		DB 接続サービスステータス	
発生要因とその処置	発生要因（推定原因）		処置対策		再発防止	
	DB 接続サービスの初期処理中に該当命令を実行した		DB 接続サービスの動作状態が動作中、または、アイドルになってから該当命令を実行してください		システム定義変数「_DBC_Status」で、DB 接続サービスの動作状態を確認してから命令を実行してください	
付属情報	付属情報 1：異常の発生箇所 付属情報 2：異常の発生箇所詳細　回路番号。プログラムセクションの場合はセクション先頭からの回路番号。ST の場合は行番号 付属情報 3：異常が発生した命令の命令名、命令インスタンス名。候補が複数ある場合は複数表示。特定できない場合は表示なし 付属情報 4：拡張エラーコード(ErrorIDEx)					
注意点／備考	なし					

イベント名	DB 処理中			イベントコード	54013016Hex	
内容	サーバ側 DB が処理中のため実行できませんでした					
発生源	PLC 機能モジュール		発生源詳細	命令	検出タイミング	命令実行時
異常の属性	重要度	監視情報	復旧方法	－	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	該当命令を仕様どおりに終了		
システム定義変数	変数名		データ型		名称	
	なし		－		－	
発生要因とその処置	発生要因（推定原因）		処置対策		再発防止	
	前回実行した命令で DB 接続命令実行タイムアウトが発生したが、サーバ側 DB の処理が完了していない状態で該当命令を実行した		ユーザプログラムから該当命令を再実行してください。ただし、DB_Insert 命令、または、DB_Update 命令の実行で、かつ、スプール機能を有効にしている場合は、SQL 文はスプールに保留されますので、該当命令を再実行する必要はありません。		サーバ側 DB の処理時間の試算を行い、DB 接続命令の実行タイミングを適切な頻度に調整してください	
付属情報	付属情報 1：異常の発生箇所 付属情報 2：異常の発生箇所詳細　回路番号。プログラムセクションの場合はセクション先頭からの回路番号。ST の場合は行番号 付属情報 3：異常が発生した命令の命令名、命令インスタンス名。候補が複数ある場合は複数表示。特定できない場合は表示なし 付属情報 4：拡張エラーコード(ErrorIDEx)					
注意点／備考	なし					

イベント名	稼働ログ無効			イベントコード	54013017Hex	
内容	指定した稼働ログが無効のためログを記録できませんでした					
発生源	PLC 機能モジュール		発生源詳細	命令	検出タイミング	命令実行時
異常の属性	重要度	監視情報	復旧方法	－	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	該当命令を仕様どおりに終了		
システム定義変数	変数名		データ型		名称	
	なし		－		－	
発生要因とその処置	発生要因（推定原因）		処置対策		再発防止	
	入力変数「LogType」で実行ログを指定したが、実行ログが無効である		DB 接続サービス設定で実行ログを有効にしてください		実行ログを有効にした状態で命令を実行してください	
	入力変数「LogType」でデバッグログを指定したが、デバッグログが記録停止中である		DB_ControlService 命令でデバッグログの記録を開始してください。または、Sysmac Studio からデバッグログの記録を開始してください		デバッグログの記録を開始した状態で命令を実行してください	
付属情報	付属情報 1：異常の発生箇所 付属情報 2：異常の発生箇所詳細　回路番号。プログラムセクションの場合はセクション先頭からの回路番号。ST の場合は行番号 付属情報 3：異常が発生した命令の命令名、命令インスタンス名。候補が複数ある場合は複数表示。特定できない場合は表示なし 付属情報 4：拡張エラーコード(ErrorIDEx)					
注意点／備考	なし					

イベント名	プロシージャハンドル不正			イベントコード	54013018Hex	
内容	指定したプロシージャハンドルが不正です					
発生源	PLC 機能モジュール		発生源詳細	命令	検出タイミング	命令実行時
異常の属性	重要度	監視情報	復旧方法	－	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	該当命令を仕様どおりに終了		
システム定義変数	変数名		データ型		名称	
	なし		－		－	
発生要因とその処置	発生要因（推定原因）		処置対策		再発防止	
	該当命令の入力変数「ProcHandle」に指定したプロシージャハンドルが誤っている		該当命令の入力変数「ProcHandle」に正しいプロシージャハンドルを指定してください		該当命令の入力変数「ProcHandle」に正しいプロシージャハンドルが指定されていることを確認してください	
付属情報	付属情報 1：異常の発生箇所 付属情報 2：異常の発生箇所詳細　回路番号。プログラムセクションの場合はセクション先頭からの回路番号。ST の場合は行番号 付属情報 3：異常が発生した命令の命令名、命令インスタンス名。候補が複数ある場合は複数表示。特定できない場合は表示なし 付属情報 4：拡張エラーコード(ErrorIDEx)					
注意点／備考	なし					

イベント名	対象外データベース種別への命令実行			イベントコード	54013019Hex	
内容	本命令の対象となっていないデータベース種別を指定して命令が実行されました					
発生源	PLC 機能モジュール		発生源詳細	命令	検出タイミング	命令実行時
異常の属性	重要度	監視情報	復旧方法	−	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	該当命令を仕様どおりに終了		
システム定義変数	変数名		データ型		名称	
	なし		−		−	
発生要因とその処置	発生要因（推定原因）		処置対策		再発防止	
	DB コネクション設定で設定したデータベース種別が該当命令ではサポート対象外		DB コネクション設定で設定したデータベース種別をサポートしているデータベース種別に修正してください		DB コネクション設定で設定したデータベース種別がサポートしているデータベース種別であることを確認してください	
付属情報	付属情報 1：異常の発生箇所 付属情報 2：異常の発生箇所詳細　回路番号。プログラムセクションの場合はセクション先頭からの回路番号。ST の場合は行番号 付属情報 3：異常が発生した命令の命令名、命令インスタンス名。候補が複数ある場合は複数表示。特定できない場合は表示なし 付属情報 4：拡張エラーコード(ErrorIDEx)					
注意点／備考	なし					

イベント名	ストアドプロシージャ名不正			イベントコード	5401301AHex	
内容	指定したストアドプロシージャ名が存在しません					
発生源	PLC 機能モジュール		発生源詳細	命令	検出タイミング	命令実行時
異常の属性	重要度	監視情報	復旧方法	−	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	該当命令を仕様どおりに終了		
システム定義変数	変数名		データ型		名称	
	なし		−		−	
発生要因とその処置	発生要因（推定原因）		処置対策		再発防止	
	該当命令の入力変数「ProcName」で指定された名前のストアドプロシージャがサーバ側 DB に存在しない （注）指定したストアドプロシージャ名が DB 上に見つからない場合も含みます。ストアドプロシージャが存在する場合でも、ストアドプロシージャへのアクセス権限がないなどの理由により、DB 接続サービスがストアドプロシージャ名を見つけられないことがあります		該当命令の入力変数「ProcName」に存在するストアドプロシージャ名を指定してください		該当命令の入力変数「ProcName」に存在するストアドプロシージャ名が指定されていることを確認してください	
付属情報	付属情報 1：異常の発生箇所 付属情報 2：異常の発生箇所詳細　回路番号。プログラムセクションの場合はセクション先頭からの回路番号。ST の場合は行番号 付属情報 3：異常が発生した命令の命令名、命令インスタンス名。候補が複数ある場合は複数表示。特定できない場合は表示なし 付属情報 4：拡張エラーコード(ErrorIDEx)					
注意点／備考	なし					

イベント名	ストアドプロシージャ引数不正			イベントコード	5401301BHex	
内容	ストアドプロシージャの引数とアタッチした引数の情報が一致しません					
発生源	PLC 機能モジュール		発生源詳細	命令	検出タイミング	命令実行時
異常の属性	重要度	監視情報	復旧方法	－	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	該当命令を仕様どおりに終了		
システム定義変数	変数名		データ型		名称	
	なし		－		－	
発生要因とその処置	発生要因（推定原因）		処置対策		再発防止	
	該当命令の入力変数「ArgIn」、「ArgOut」、「ArgInOut」とサーバ側 DB から取得するストアドプロシージャの引数情報の名前、数、型が一致しない		該当命令の入力変数「ArgIn」、「ArgOut」、「ArgInOut」とサーバ側 DB のストアドプロシージャの引数の名前、数、型を一致させてください		該当命令の入力変数「ArgIn」、「ArgOut」、「ArgInOut」とサーバ側 DB のストアドプロシージャの引数の名前、数、型が一致することを確認してください	
付属情報	付属情報 1：異常の発生箇所 付属情報 2：異常の発生箇所詳細　回路番号。プログラムセクションの場合はセクション先頭からの回路番号。ST の場合は行番号 付属情報 3：異常が発生した命令の命令名、命令インスタンス名。候補が複数ある場合は複数表示。特定できない場合は表示なし 付属情報 4：拡張エラーコード(ErrorIDEx)					
注意点／備考	なし					

イベント名	ストアドプロシージャ結果セットカラム数不正			イベントコード	5401301CHex	
内容	ストアドプロシージャの結果セットのカラム数が格納先の構造体型変数のメンバ数と一致しません					
発生源	PLC 機能モジュール		発生源詳細	命令	検出タイミング	命令実行時
異常の属性	重要度	監視情報	復旧方法	－	ログカテゴリ	システム
発生の影響	ユーザプログラム	継続	動作	該当命令を仕様どおりに終了		
システム定義変数	変数名		データ型		名称	
	なし		－		－	
発生要因とその処置	発生要因（推定原因）		処置対策		再発防止	
	該当命令により取得された結果セットのカラム数と格納先の構造体型変数のメンバ数が一致しない		該当命令により取得された結果セットのカラム数と格納先の構造体型変数のメンバ数を一致させてください		該当命令により取得された結果セットのカラム数と格納先の構造体型変数のメンバ数が一致することを確認してください	
付属情報	付属情報 1：異常の発生箇所 付属情報 2：異常の発生箇所詳細　回路番号。プログラムセクションの場合はセクション先頭からの回路番号。ST の場合は行番号 付属情報 3：異常が発生した命令の命令名、命令インスタンス名。候補が複数ある場合は複数表示。特定できない場合は表示なし 付属情報 4：拡張エラーコード(ErrorIDEx)					
注意点／備考	なし					

イベント名	DB 接続サービス未起動			**イベントコード**	5401301EHex	
内容	DB 接続サービスが［使用しない］設定です					
発生源	PLC 機能モジュール		**発生源詳細**	命令	**検出タイミング**	命令実行時
異常の属性	**重要度**	監視情報	**復旧方法**	－	**ログカテゴリ**	システム
発生の影響	**ユーザプログラム**	継続	**動作**	該当命令を仕様どおりに終了		
システム定義変数	**変数名**		**データ型**		**名称**	
	なし		－		－	
発生要因とその処置	**発生要因（推定原因）**		**処置対策**		**再発防止**	
	DB 接続サービスを［使用しない］設定で該当命令を実行した		DB 接続サービスを開始してください		－	
付属情報	付属情報 1：異常の発生箇所 付属情報 2：異常の発生箇所詳細　回路番号。プログラムセクションの場合はセクション先頭からの回路番号。ST の場合は行番号 付属情報 3：異常が発生した命令の命令名、命令インスタンス名。候補が複数ある場合は複数表示。特定できない場合は表示なし 付属情報 4：拡張エラーコード(ErrorIDEx)					
注意点／備考	なし					

A

付録

A-1　タスク設計方法 A-2
A-1-1　DB 接続サービスの起動時間 A-2
A-1-2　DB 接続命令の実行時間の参考値 A-4
A-1-3　DB 接続命令の実行時間の測定方法 A-12
A-1-4　システムサービス実行時間割合の目安 A-12
A-1-5　システムサービス実行時間割合の確認 A-15

A-2　DB 接続命令の実行時間 A-17
A-2-1　DB 接続命令の実行時間の制約 A-17
A-2-2　稼働ログ記録の DB 接続命令実行時間への影響 A-22
A-2-3　DB 応答時間の測定方法 A-23
A-2-4　命令実行タイムアウト監視による装置性能（タクト時間）の確保 A-24

A-3　仕様 A-25
A-3-1　一般仕様 A-25
A-3-2　性能仕様 A-25
A-3-3　機能仕様 A-25

A-4　バージョン関連情報 A-26
A-4-1　ユニットバージョンと DB 接続サービスのバージョン A-26
A-4-2　ユニットバージョンによる DB 接続機能の追加と変更 A-27
A-4-3　ユニットバージョン／DB 接続サービスのバージョンと Sysmac Studio の設定ユニットバージョン A-28
A-4-4　DB 接続サービスのバージョンとデバイス変更後の接続データベース種別 A-31
A-4-5　DB 接続サービスのバージョンと接続データベースの種別およびバージョン A-31

A-1　タスク設計方法

ここでは、DB 接続機能を使用する上での、タスク設計方法について説明します。

NJ/NX シリーズコントローラのタスクの動作仕様、およびシステムサービスの動作仕様については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル ソフトウェア編（SBCA-467)』を参照してください。

A-1-1　DB 接続サービスの起動時間

CPU ユニットの電源投入から、DB 接続サービスが利用できる状態（システム定義変数「_DBC_Status.Run」が True）になるまでの時間（以下、起動時間）は、接続するデータベース種別やタスク実行時間割合で変わります。
以下に、いくつか組み合わせたときの参考値を掲載します
これらの値を参考に、お客様のシステムを設計してください。

・形 NX701-1620

DB の種別	DB 接続サービス起動時間の参考値（平均）
Oracle	9.49 秒
SQL Server	8.33 秒
DB2	8.84 秒
MySQL	8.32 秒
Firebird	8.32 秒
PostgreSQL	8.32 秒

・形 NX502-1500

DB の種別	DB 接続サービス起動時間の参考値（平均）
Oracle	42.14 秒
SQL Server	37.30 秒
MySQL	36.18 秒
PostgreSQL	38.62 秒

・形 NX102-1220

DB の種別	タスク実行時間割合*1	DB 接続サービス起動時間の参考値（平均）
Oracle	50%	70.85 秒
	80%	74.37 秒
SQL Server	50%	58.41 秒
	80%	61.16 秒
DB2	50%	70.77 秒
	80%	72.56 秒
MySQL	50%	56.52 秒
	80%	59.52 秒
FireBird	50%	60.72 秒
	80%	61.77 秒
PostgreSQL	50%	57.30 秒
	80%	61.55 秒

*1.　Sysmac Studio のタスク実行時間モニタで見たタスク実行時間の割合。タスク実行時の負荷は、擬似的に加えたものです。

• 形 NJ501-1520

DB の種別	タスク実行時間割合*1	DB 接続サービス起動時間の参考値（平均）
Oracle	50%	58.43 秒
	80%	124.95 秒
SQL Server	50%	54.02 秒
	80%	120.95 秒
DB2	50%	56.26 秒
	80%	128.49 秒
MySQL	50%	57.41 秒
	80%	131.33 秒
Firebird	50%	56.65 秒
	80%	129.07 秒
PostgreSQL	50%	59.06 秒
	80%	124.26 秒

*1. Sysmac Studio のタスク実行時間モニタで見たタスク実行時間の割合。タスク実行時の負荷は、擬似的に加えたものです。

• 形 NJ101-1020

DB の種別	タスク実行時間割合*1	DB 接続サービス起動時間の参考値（平均）
Oracle	50%	75.59 秒
	60%	89.31 秒
SQL Server	50%	56.36 秒
	60%	67.17 秒
DB2	50%	61.90 秒
	60%	73.35 秒
MySQL	50%	54.46 秒
	60%	66.83 秒
Firebird	50%	57.61 秒
	60%	70.98 秒
PostgreSQL	50%	63.61 秒
	60%	76.63 秒

*1. Sysmac Studio のタスク実行時間モニタで見たタスク実行時間の割合。タスク実行時の負荷は、擬似的に加えたものです。

測定条件と内容は以下のとおりです。

測定条件		内容
大項目	小項目	
CPU ユニット	タスク構成	プライマリ定周期タスクのみ タスク周期：1ms
システム構成	基本構成	• EtherCAT ネットワーク構成なし • CJ ユニット構成なし • Sysmac Studio と USB で接続*1
	ネットワーク構成	• 他のコントローラとの接続なし • 表示器との接続なし

*1. 形 NX102-1220 は、内蔵 EtherNet/IP ポート 1 と HUB 接続

使用上の注意

- DB 接続サービスは、システムサービス内で実行します。
  そのため、DB 接続サービスの起動処理と、システムサービスの他の処理が同時に実行される場合、各処理の実行時間が長くなることがあります。
- システムサービス実行時間割合の目安が異なるため、タスク実行時間割合は機種により測定条件が異なります。システムサービス実行時間割合の目安の詳細は、「A-1-4 システムサービス実行時間割合の目安 (P.A-12)」を参照してください。

A-1-2 DB 接続命令の実行時間の参考値

DB 接続命令は、数のタスク周期にまたがって実行されるファンクションブロック型の命令です。
DB 接続命令の実行時間の参考値は以下のとおりです。
DB 接続命令の実行時間の変動要因については、「A-2-1 DB 接続命令の実行時間の制約 (P.A-17)」を参照してください。

- 以降の参考値表記で使用される測定名と対象命令、測定方法は、以下のとおりです。

測定名	対象命令	条件	備考
DB_Insert	DB_Insert	100 カラムのレコードの INSERT 実行時	
DB_Select	DB_Select	10 万件のレコードから 1 件のレコードを検索し、100 カラム取り出したとき	抽出条件に主キーを指定
DB_BatchInsert_100	DB_BatchInsert	100 レコードの BATCHINSERT 実行時	1 レコードは 100 カラム ※INSERT と同等
DB_BatchInsert_500		500 レコードの BATCHINSERT 実行時	

測定条件 1

測定条件と内容は、以下のとおりです。

測定条件		内容
大項目	小項目	
CPU ユニット	タスク構成	プライマリ定周期タスクのみ タスク周期：1ms タスク周期に対するタスク実行時間の割合：80%,50%
DB を配置したサーバ	PC	CPU：Intel Xeon(R) CPU E31220 @ 3.10GHz, Quad-Core メモリ：8.00GB
	OS	Windows Server 2008 Standard 64bit
	DB の種別	Oracle Database Express Edition 12c SQL Server 2012 MySQL Community Edition 5.6 DB2 for Linux and Windows 10.5 Firebird 2.5 PostgreSQL 9.4
実行する SQL 文の内容	レコード構成	INT 型：40 カラム REAL 型：40 カラム STRING[16]型：16 カラム DATE_AND_TIME 型：4 カラム
稼働ログ	実行ログ	記録する（デフォルト）
	デバッグログ	停止（デフォルト）
	SQL 実行失敗ログ	記録しない（デフォルト）

● **形 NX701-1620**

DB の種別	測定名	命令実行時間の参考値 (ms)	
		平均	最大
Oracle Database 12c	DB_Insert	2.4	12
	DB_Select	5.11	22
SQL Server 2012	DB_Insert	2.35	13
	DB_Select	2.62	21
MySQL 5.6 ストレージエンジン InnoDB	DB_Insert	15.76	74
	DB_Select	2.21	10
FireBird 2.5	DB_Insert	12.95	32
	DB_Select	21.22	32
DB2 10.5	DB_Insert	3.44	21
	DB_Select	7.39	20
PostgreSQL 9.4	DB_Insert	2.93	21
	DB_Select	5.16	14

● **形 NX102-1120**

DB の種別	タスク実行時間割合	測定名	命令実行時間の参考値 (ms)	
			平均	最大
Oracle Database 12c	50%	DB_Insert	16.53	109
		DB_Select	23.02	244
	80%	DB_Insert	20.15	119
		DB_Select	22.56	168
SQL Server 2012	50%	DB_Insert	15.64	114
		DB_Select	16.36	107
	80%	DB_Insert	19.31	117
		DB_Select	18.98	89
MySQL 5.6 ストレージエンジン InnoDB	50%	DB_Insert	31.73	137
		DB_Select	12.77	114
	80%	DB_Insert	35.48	118
		DB_Select	13.58	100
FireBird 2.5	50%	DB_Insert	24.71	124
		DB_Select	56.92	152
	80%	DB_Insert	26.99	115
		DB_Select	60.22	135
DB2 10.5	50%	DB_Insert	20.78	130
		DB_Select	29.26	350
	80%	DB_Insert	24.49	127
		DB_Select	32.98	333
PostgreSQL 9.4	50%	DB_Insert	16.52	119
		DB_Select	22.28	113
	80%	DB_Insert	19.69	117
		DB_Select	23.86	112

測定条件 2

測定条件と内容は、以下のとおりです。

測定条件		内容
大項目	小項目	
CPU ユニット	タスク構成	プライマリ定周期タスクのみ タスク周期：1ms タスク周期に対するタスク実行時間の割合：80%,50%
DB を配置したサーバ	PC	CPU：Intel Xeon(R) CPU E5-2603 @ 1.7GHz, 6 Core メモリ：32.00GB
	OS	Windows Server 2016 Standard 64bit
	DB の種別	Oracle Database Express Edition 12c SQL Server 2016 MySQL Community Edition 5.6 DB2 for Linux and Windows 10.5 Firebird 2.5 PostgreSQL 9.4
実行する SQL 文の内容	レコード構成	INT 型：40 カラム REAL 型：40 カラム STRING[16]型：16 カラム DATE_AND_TIME 型：4 カラム
稼働ログ	実行ログ	記録する（デフォルト）
	デバッグログ	停止（デフォルト）
	SQL 実行失敗ログ	記録しない（デフォルト）

- **形 NX701-1620**

DB の種別	測定名	命令実行時間の参考値 (ms)	
		平均	最大
Oracle Database 12c	DB_Insert	2.21	11
	DB_Select	6.07	14
	DB_BatchInsert_100	16.2	67
	DB_BatchInsert_500	74.6	397
SQL Server 2016	DB_Insert	2.49	8
	DB_Select	2.69	11
	DB_BatchInsert_100	29.47	44
	DB_BatchInsert_500	133	219
MySQL 5.6　ストレージエンジン InnoDB	DB_Insert	3.02	11
	DB_Select	2.09	10
	DB_BatchInsert_100	158	178
	DB_BatchInsert_500	775	840
FireBird 2.5	DB_Insert	5.29	20
	DB_Select	22.62	57
DB2 10.5	DB_Insert	3.11	11
	DB_Select	10.36	22
PostgreSQL 9.4	DB_Insert	2.48	10
	DB_Select	6.75	14
	DB_BatchInsert_100	69.79	105
	DB_BatchInsert_500	338	415

● **形 NX102-1120**

DB の種別	タスク実行時間割合	測定名	命令実行時間の参考値 (ms)	
			平均	最大
Oracle Database 12c	50%	DB_Insert	16.35	119
		DB_Select	20.54	186
		DB_BatchInsert_100	158	224
		DB_BatchInsert_500	745	921
	80%	DB_Insert	19.32	118
		DB_Select	22.55	170
		DB_BatchInsert_100	175	310
		DB_BatchInsert_500	775	964
SQL Server 2016	50%	DB_Insert	15.4	107
		DB_Select	15.84	114
		DB_BatchInsert_100	234	328
		DB_BatchInsert_500	1134	1369
	80%	DB_Insert	19	115
		DB_Select	19.19	118
		DB_BatchInsert_100	255	354
		DB_BatchInsert_500	1162	1347
MySQL 5.6 ストレージエンジン InnoDB	50%	DB_Insert	17.27	113
		DB_Select	12.42	107
		DB_BatchInsert_100	520	619
		DB_BatchInsert_500	2564	2966
	80%	DB_Insert	19.87	110
		DB_Select	13.64	106
		DB_BatchInsert_100	525	610
		DB_BatchInsert_500	2676	3150
FireBird 2.5	50%	DB_Insert	21.6	121
		DB_Select	56.94	402
	80%	DB_Insert	24.98	120
		DB_Select	60.6	580
DB2 10.5	50%	DB_Insert	19.76	124
		DB_Select	31.54	335
	80%	DB_Insert	23.47	120
		DB_Select	35.19	204
PostgreSQL 9.4	50%	DB_Insert	16.61	116
		DB_Select	22.56	120
		DB_BatchInsert_100	198	587
		DB_BatchInsert_500	926	1162
	80%	DB_Insert	19.2	107
		DB_Select	25.16	123
		DB_BatchInsert_100	204	590
		DB_BatchInsert_500	958	1329

測定条件 3

測定条件と内容は、以下のとおりです。

測定条件		内容
大項目	小項目	
CPU ユニット	タスク構成	プライマリ定周期タスクのみ タスク周期：1ms タスク周期に対するタスク実行時間割合：80%, 50%
DB を配置したサーバ	PC	CPU：Intel Xeon(R) CPU E31220 @ 3.10GHz 3.09GHz メモリ：8.00GB
	OS	Windows Server 2008 Standard SP2 64bit
	DB の種別	Oracle Database Express Edition 11g 11.2.0 SQL Server 2012 DB2 for Linux, UNIX and Windows 10.5 MySQL Community Edition 5.6 Firebird 2.5 PostgreSQL 9.4
実行する SQL 文の内容	レコード構成	INT 型：40 カラム REAL 型：40 カラム STRING[16]型：16 カラム DATE_AND_TIME 型：4 カラム
稼働ログ	実行ログ	記録する（デフォルト）
	デバッグログ	停止（デフォルト）
	SQL 実行失敗ログ	記録しない（デフォルト）

● 形 NJ501-1520

DB の種別	タスク実行時間割合	測定名	命令実行時間の参考値(ms)	
			平均値	最大値
Oracle Database 11g	50%	DB_Insert	16.2	65
		DB_Select	37.1	75
	80%	DB_Insert	49.2	184
		DB_Select	101.6	272
SQL Server 2012	50%	DB_Insert	16.1	57
		DB_Select	23.8	98
	80%	DB_Insert	45.5	112
		DB_Select	72.5	236
DB2 10.5	50%	DB_Insert	27.5	115
		DB_Select	37.1	80
	80%	DB_Insert	69.4	176
		DB_Select	99.5	352
MySQL 5.6 ストレージエンジン：InnoDB	50%	DB_Insert	40.3	273
		DB_Select	32.0	41
	80%	DB_Insert	65.0	315
		DB_Select	69.4	164
Firebird 2.5	50%	DB_Insert	23.8	156
		DB_Select	71.7	153
	80%	DB_Insert	52.8	139
		DB_Select	118.4	234
PostgreSQL 9.4	50%	DB_Insert	17.0	78
		DB_Select	30.9	83
	80%	DB_Insert	48.3	175
		DB_Select	89.1	250

測定条件 4

測定条件と内容は、以下のとおりです。

測定条件		内容
大項目	小項目	
CPU ユニット	タスク構成	プライマリ定周期タスクのみ タスク周期：1ms タスク周期に対するタスク実行時間の割合： • 形 NJ501-1520: 80%, 50% • 形 NJ101-9020: 60%, 50%
DB を配置したサーバ	PC	CPU：Intel Xeon(R) CPU E5-2603 @ 1.7GHz, 6 Core メモリ：32.00GB
	OS	Windows Server 2016 Standard 64bit
	DB の種別	Oracle Database Express Edition 12c SQL Server 2016 DB2 for Linux, UNIX and Windows 10.5 MySQL Community Edition 5.6 Firebird 2.5 PostgreSQL 9.4
実行する SQL 文の内容	レコード構成	INT 型：40 カラム REAL 型：40 カラム STRING[16]型：16 カラム DATE_AND_TIME 型：4 カラム
稼働ログ	実行ログ	記録する（デフォルト）
	デバッグログ	停止（デフォルト）
	SQL 実行失敗ログ	記録しない（デフォルト）

● 形 NJ501-1520

DB の種別	タスク実行時間割合	測定名	命令実行時間の参考値(ms)	
			平均値	最大値
Oracle Database 12c	50%	DB_Insert	10.81	158
		DB_Select	103.08	167
		DB_BatchInsert_100	56.78	275
		DB_BatchInsert_500	257.47	519
	80%	DB_Insert	22.29	230
		DB_Select	130.81	272
		DB_BatchInsert_100	110.91	528
		DB_BatchInsert_500	488.97	1007
SQL Server 2016	50%	DB_Insert	10.63	51
		DB_Select	22.00	50
		DB_BatchInsert_100	85.10	200
		DB_BatchInsert_500	380.20	551
	80%	DB_Insert	21.63	93
		DB_Select	42.45	101
		DB_BatchInsert_100	159.80	345
		DB_BatchInsert_500	714.56	1092
DB2 10.5	50%	DB_Insert	12.27	623
		DB_Select	226.75	302
	80%	DB_Insert	25.76	689
		DB_Select	258.00	450
MySQL 5.6 ストレージエンジン：InnoDB	50%	DB_Insert	21.68	120
		DB_Select	22.31	48

DB の種別	タスク実行時間割合	測定名	命令実行時間の参考値(ms)	
			平均値	最大値
		DB_BatchInsert_100	340.09	492
		DB_BatchInsert_500	1529.69	1879
	80%	DB_Insert	36.64	155
		DB_Select	34.21	92
		DB_BatchInsert_100	519.33	838
		DB_BatchInsert_500	2469.83	2993
Firebird 2.5	50%	DB_Insert	15.62	423
		DB_Select	303.62	676
	80%	DB_Insert	26.40	158
		DB_Select	323.63	717
PostgreSQL 9.4	50%	DB_Insert	12.05	107
		DB_Select	56.75	136
		DB_BatchInsert_100	106.61	319
		DB_BatchInsert_500	500.07	870
	80%	DB_Insert	22.58	185
		DB_Select	80.79	226
		DB_BatchInsert_100	154.58	589
		DB_BatchInsert_500	694.98	1499

● 形 NJ101-9020

DB の種別	タスク実行時間割合	測定名	命令実行時間の参考値(ms)	
			平均値	最大値
Oracle Database 12c	50%	DB_Insert	27.8	311
		DB_Select	42.0	311
		DB_BatchInsert_100	130.5	679
		DB_BatchInsert_500	595.3	1235
	60%	DB_Insert	39.0	342
		DB_Select	62.4	369
		DB_BatchInsert_100	157.8	791
		DB_BatchInsert_500	719.9	1479
SQL Server 2016	50%	DB_Insert	26.7	287
		DB_Select	36.2	626
		DB_BatchInsert_100	188.26	414
		DB_BatchInsert_500	861.49	1186
	60%	DB_Insert	37.5	621
		DB_Select	52.1	456
		DB_BatchInsert_100	226.03	524
		DB_BatchInsert_500	1084.66	2857
DB2 10.5	50%	DB_Insert	39.8	544
		DB_Select	59.0	467
	60%	DB_Insert	52.3	397
		DB_Select	81.0	655
MySQL 5.6 ストレージエンジン：InnoDB	50%	DB_Insert	44.2	365
		DB_Select	36.0	599
		DB_BatchInsert_100	480.11	764
		DB_BatchInsert_500	2313.01	2636
	60%	DB_Insert	54.6	834
		DB_Select	52.4	450
		DB_BatchInsert_100	562.12	999

DB の種別	タスク実行時間割合	測定名	命令実行時間の参考値(ms)	
			平均値	最大値
		DB_BatchInsert_500	2700.42	3127
Firebird 2.5	50%	DB_Insert	34.0	314
		DB_Select	78.4	403
	60%	DB_Insert	45.4	388
		DB_Select	101.0	472
PostgreSQL 9.4	50%	DB_Insert	28.7	306
		DB_Select	45.0	291
		DB_BatchInsert_100	174.25	699
		DB_BatchInsert_500	798.04	1525
	60%	DB_Insert	41.3	471
		DB_Select	66.0	433
		DB_BatchInsert_100	199.67	846
		DB_BatchInsert_500	931.63	1913

測定条件 5

測定条件と内容は、以下のとおりです。

測定条件		内容
大項目	小項目	
CPU ユニット	タスク構成	プライマリ定周期タスクのみ タスク周期：1ms タスク周期に対するタスク実行時間の割合：80%
DB を配置したサーバ ハイパフォーマンス	PC	CPU：Intel Xeon(R) CPU E5-2603 @ 1.7GHz, 6 Core メモリ：32GB
	OS	Windows Server 2016 Standard 64bit
	DB の種別	Oracle Database Express Edition 19c SQL Server 2019 MySQL Community Edition 8.0 PostgreSQL 14
実行する SQL 文の内容	レコード構成	INT 型：40 カラム REAL 型：40 カラム STRING[16]型：16 カラム DATE_AND_TIME 型：4 カラム
稼働ログ	実行ログ	記録する（デフォルト）
	デバッグログ	停止（デフォルト）
	SQL 実行失敗ログ	記録しない（デフォルト）

● 形 NX502-1500

DB の種別	測定名	命令実行時間の参考値 (ms)	
		平均	最大
Oracle Database 19c	DB_Insert	3.75	290
	DB_Select	9.78	385
	DB_BatchInsert_100	17.54	584
	DB_BatchInsert_500	72.25	934
SQL Server 2019	DB_Insert	3.60	256
	DB_Select	4.73	280
	DB_BatchInsert_100	40.45	638
	DB_BatchInsert_500	171.53	962

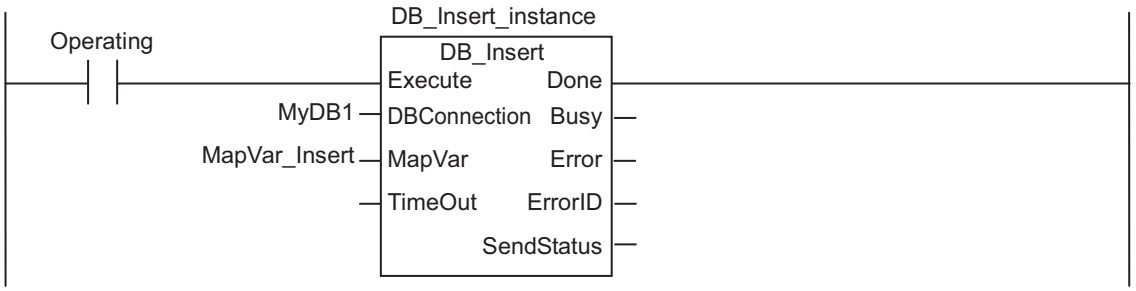
DB の種別	測定名	命令実行時間の参考値 (ms)	
		平均	最大
MySQL 8.0 ストレージエンジン InnoDB	DB_Insert	5.29	217
	DB_Select	3.45	164
	DB_BatchInsert_100	222.41	537
	DB_BatchInsert_500	1067.91	1586
PostgreSQL 14	DB_Insert	4.61	284
	DB_Select	8.68	281
	DB_BatchInsert_100	77.35	440
	DB_BatchInsert_500	361.00	893

A-1-3　DB 接続命令の実行時間の測定方法

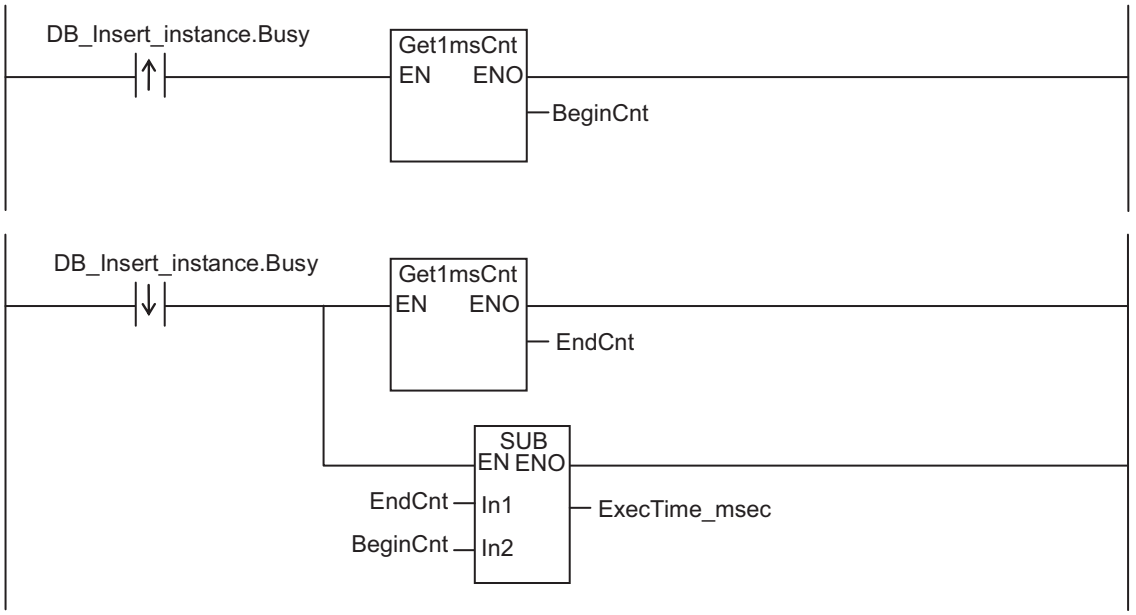
DB 接続命令の実行時間は、Get1msCnt 命令を使用して、出力変数 Busy が TRUE になってから、FALSE になるまでのフリーランカウンタ値を計算することで測定できます。

• DB_Insert 命令の実行時間の測定例

DB コネクション「MyDB1」にレコードを挿入する。



DB_Insert 命令の実行時間を測定し、結果を変数 ExecTime_msec に格納する。



A-1-4　システムサービス実行時間割合の目安

DB 接続サービスは、システムサービス内で実行します。

ユーザプログラムで DB 接続命令を実行すると、システムサービス内で DB 接続サービスが処理を実行します。

システムサービスの実行方法は、CPU ユニットの形式によって異なります。

安全上の要点

システムサービスの実行時間割合（CPU 使用率）は、あくまで目安です。
システムサービス内で実行する他のサービスの使用量によって、システムサービスの実行時間割合（CPU 使用率）は変化します。
最終的には、必ず実機を使用してあらゆる条件で性能確認を行い、DB 接続命令が適切な実行時間で動作することを確認してください。

● 形 NJ501-1□20、形 NJ501-4320、形 NJ101-□□20 の場合

形 NJ501-1□20、形 NJ501-4320、形 NJ101-□□20 の場合、システムサービスの実行時間を十分に確保できない場合は、DB 接続命令の実行時間が長くなる場合があります。
または、システムサービス内で動作する他の処理の実行時間が長くなる場合があります。
DB 接続命令を性能仕様通りに動作させるには、システムサービスの実行時間割合（CPU 使用率）が、以下となるようにタスク設計を行ってください。

CPU ユニットの形式	システムサービスの実行時間割合の目安
形 NJ501-1□20 形 NJ501-4320	20%以上
形 NJ101-□□20	40%以上

使用上の注意

- システムサービスの実行時間割合が少なくなると、Sysmac Studio から各操作を実行したとき、操作の失敗または通信異常が発生することがあります。操作の失敗または通信異常が発生する場合は、以下を実施した後に各操作を再実行してください。
  a）ケーブル接続を確認
  b）通信設定を確認
  c）通信設定のレスポンス監視時間を延ばす
  d）DB 接続サービスの稼働状態が“初期処理中”、“異常停止中”、“終了”以外であることを確認
  DB 接続サービスの稼働状態の詳細は、「4-3-1 DB 接続サービスの稼働状態 (P.4-7)」を参照してください。
- Sysmac Studio がオンライン接続できない場合は、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。
- システムサービス監視設定分のシステムサービスが確保できない場合、“システムサービス時間不足”が発生します。“システムサービス時間不足”は全停止フォールトレベルのコントローラ異常です。エラー発生時に、ユーザプログラムの実行を停止します。システムサービスとタスクの実行時間を十分に確保するために、システムサービス処理の応答性能を満たす範囲で最小の値をシステムサービス監視設定に設定してください。システムサービス監視設定は、設定した時間分のシステムサービス実行時間が確保されているかを監視するための設定です。設定した時間分のシステムサービス実行を保証するものではありません。
- システムサービスの実行時間には、タスク実行時間とタグデータリンクが影響します。タスクの仕様、タグデータリンクサービスとシステムサービスの詳細は、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル ソフトウェア編 （SBCA-467)』を参照してください。
- CJ シリーズ EtherNet/IP ユニットの起動処理は、CPU ユニットの電源投入時のシステムサービス実行時間に依存します。また、形 NJ101-□□20 では、電源投入時のシステムサービス実行時間が不足する場合があります。そのため、形 NJ101-□□20 で CJ シリーズ EtherNet/IP ユニットを使用する場合、起動時の動作をあらかじめ確認してください。
  CPU ユニットの電源投入時のシステムサービス実行時間が不足する場合、［電源投入時の起動待ち時間］の設定を見直してください。
  ［電源投入時の起動待ち時間］の設定については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル ソフトウェア編 （SBCA-467)』を参照してください。
  CJ シリーズ EtherNet/IP ユニットの詳細は、『CJ シリーズ EtherNet/IP ユニット ユーザーズマニュアル NJ シリーズ接続編(SBCD-355)』を参照してください。

● 形 NJ501-R□20 の場合

形 NJ501-R□20 の場合、V+タスクと DB 接続サービスがシステムサービス内で実行します。
システムサービスの実行時間を十分に確保できない場合は、DB 接続命令または V+タスクの実行時間が長くなることがあります。
または、システムサービス内で動作する他の処理の実行時間が長くなる場合があります。
DB 接続命令を性能仕様通りに動作させるには、システムサービスの実行時間割合（CPU 使用率）が、以下となるようにタスク設計を行ってください。

CPU ユニットの形式	システムサービスの実行時間割合の目安
形 NJ501-R□20	35%以上

使用上の注意

- システムサービスの実行時間割合が少なくなると、Sysmac Studio から各操作を実行したとき、操作の失敗または通信異常が発生することがあります。操作の失敗または通信異常が発生する場合は、以下を実施した後に各操作を再実行してください。
  a）ケーブル接続を確認
  b）通信設定を確認
  c）通信設定のレスポンス監視時間を延ばす
  d）DB 接続サービスの稼働状態が“初期処理中”、“異常停止中”、“終了”以外であることを確認
  DB 接続サービスの稼働状態の詳細は、「4-3-1 DB 接続サービスの稼働状態 (P.4-7)」を参照してください。
- Sysmac Studio がオンライン接続できない場合は、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。
- システムサービス監視設定分のシステムサービスが確保できない場合、“システムサービス時間不足”が発生します。“システムサービス時間不足”は全停止フォールトレベルのコントローラ異常です。エラー発生時に、ユーザプログラムの実行を停止します。システムサービスとタスクの実行時間を十分に確保するために、システムサービス処理の応答性能を満たす範囲で最小の値をシステムサービス監視設定に設定してください。システムサービス監視設定は、設定した時間分のシステムサービス実行時間が確保されているかを監視するための設定です。設定した時間分のシステムサービス実行を保証するものではありません。
- システムサービスの実行時間には、タスク実行時間とタグデータリンクが影響します。タスクの仕様、タグデータリンクサービスとシステムサービスの詳細は、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル ソフトウェア編 (SBCA-467)』を参照してください。
- CJ シリーズ EtherNet/IP ユニットの起動処理は、CPU ユニットの電源投入時のシステムサービス実行時間に依存します。
  CPU ユニットの電源投入時のシステムサービス実行時間が不足する場合、［電源投入時の起動待ち時間］の設定を見直してください。
  ［電源投入時の起動待ち時間］の設定については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル ソフトウェア編 (SBCA-467)』を参照してください。
  CJ シリーズ EtherNet/IP ユニットの詳細は、『CJ シリーズ EtherNet/IP ユニット ユーザーズマニュアル NJ シリーズ接続編 (SBCD-355)』を参照してください。

● 形 NX701-□□20 の場合

形 NX701-□□20 の場合、システムサービスは、タスクやタグデータリンクサービスの実行と並列に、いつでも実行されます。
そのため、システムサービスの実行時間が不足することがありません。

● 形 NX502-1□00 の場合

形 NX502-1□00 の場合、システムサービスは、タスクやタグデータリンクサービスの実行と並列に、いつでも実行されます。
そのため、システムサービスの実行時間が不足することがありません。

● 形 NX102-□□20 の場合

形 NX102-□□20 の場合、システムサービスは、タスクの実行と並列に実行されます。
ただし、タグデータリンクサービスの実行中は、システムサービスは実行されません。

A-1-5　システムサービス実行時間割合の確認

形 NJ501-□□20 または形 NJ101-□□20 の場合は、以下の方法で、システムサービスの実行時間が十分に確保できていることを確認して、タスク設計を行ってください。

● 机上での計算の場合

ここでは、プライマリ定周期タスク１つ（シングルタスク）で構成する場合を例とします。
『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル ソフトウェア編 (SBCA-467)』を参照して、机上にて“タスク実行時間の平均値”の概算見積もりを行います。

- 形 NJ501-□□20 の場合
  “タスク実行時間の平均値” ＜ “タスク周期” × 0.8
- 形 NJ101-□□20 の場合
  “タスク実行時間の平均値” ＜ “タスク周期” × 0.6

を目安としてタスク設計してください。

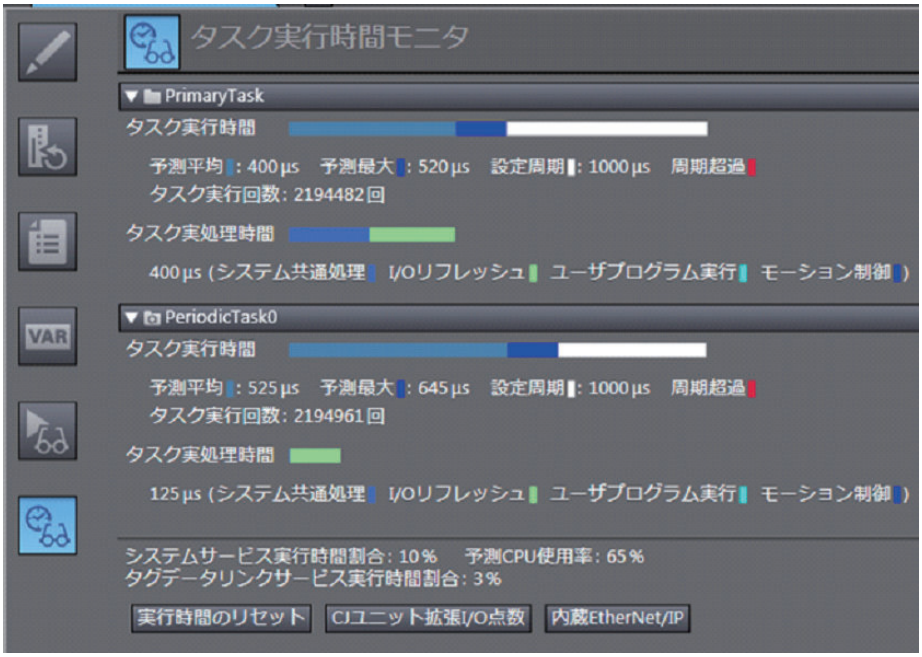
● Sysmac Studio のシミュレータでの計算の場合

Sysmac Studio のシミュレータの“タスク実行時間モニタ”にて、“予測 CPU 使用率”を確認します。
シミュレータ上で動作確認するための手順は、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル ソフトウェア編（SBCA-467)』を参照してください。

- 形 NJ501-□□20 の場合
  “予測 CPU 使用率” - “システムサービス実行時間割合” ＜ 80%
- 形 NJ101-□□20 の場合
  “予測 CPU 使用率” - “システムサービス実行時間割合” ＜ 60%

を目安としてタスク設計してください。

“予測 CPU 使用率”は、タスク処理時間予測最大値、タグデータリンクサービス実行時間割合、システムサービス監視設定に指定された、確保すべきシステムサービス処理時間の合計がタスク周期に対して、どの程度占有しているかを表示します。
“予測 CPU 使用率”から“システムサービス実行時間割合”を引いた値が、システムサービス以外の処理の実行時間の割合となります。



● **実機コントローラでの計算の場合**

プライマリ定周期タスクが 1 つだけの構成の場合は、実機コントローラ接続時、Sysmac Studio の“タスク実行時間モニタ機能”で、“タスク実行時間の平均値”を確認します。

- 形 NJ501-□□20 の場合
  “タスク実行時間の平均値” ＜ “タスク周期” × 0.8
- 形 NJ101-□□20 の場合
  “タスク実行時間の平均値” ＜ “タスク周期” × 0.6

を目安としてタスク設計してください。

マルチタスクの構成の場合は、実機コントローラを使用してあらゆる条件で性能確認を行い、DB 接続命令が適切な実行時間で動作することを確認してください。

A-2 DB 接続命令の実行時間

ここでは、DB 接続命令の実行時間について説明しています。

A-2-1 DB 接続命令の実行時間の制約

DB 接続命令の実行時間は、以下の要因により変動します。
- NJ/NX シリーズ CPU ユニットの状態
- DB の種別
- DB を配置したサーバの処理能力、負荷状況
- サーバ側 DB の処理時間
- 実行する SQL 文の内容
- DB_Select 命令実行時の抽出レコード数
- 暗号化通信機能の使用

上記の要因によって、DB 接続命令の実行時間が「A-1-2 DB 接続命令の実行時間の参考値 (P.A-4)」に記載している参考値より長くなる場合があります。
弊社の測定環境で確認している以下の事象とその対処方法について説明します。

No.	事象
1	CPU ユニット電源投入後、初回の DB 接続命令（DB_Insert 命令、DB_Update 命令、DB_Select 命令、DB_Delete 命令）の実行時間が長くなった
2	DB_CreateMapping 命令実行後、初回の DB_Insert 命令の実行時間が長くなった
3	CPU ユニットで通信処理やメモリカード処理を実行したときに DB 接続命令の実行時間が長くなった
4	定常的に DB 接続命令の実行時間が長くなることがある
5	サーバ側 DB の状況により、DB 接続命令（DB_Insert 命令、DB_Update 命令、DB_Select 命令、DB_Delete 命令）の実行時間が長くなった

測定条件と内容は、「A-1-2 DB 接続命令の実行時間の参考値 (P.A-4)」を参照ください。

事象 1：CPU ユニット電源投入後、初回の DB 接続命令（DB_Insert 命令、DB_Update 命令、DB_Select 命令、DB_Delete 命令）の実行時間が長くなった

● 発生要因（推定原因）

発生要因（推定原因）として、以下の要因が考えられます。
- CPU ユニットの電源投入後、初回のレコード処理命令は、CPU ユニットの処理時間が、通常時よりも長くなる傾向があります。
- DB_CreateMapping 命令実行後の初回の DB_Insert 命令は、DB 側の処理時間が、通常時よりも長くなる傾向があります。

CPU ユニットの電源投入後の初回の DB 接続命令の実行時間の参考値は以下のとおりです。タスク実行時間割合は、50%とします。

- 形 NX701-□□20

DB の種別	命令	命令実行時間の参考値	測定条件
Oracle Database 11g	DB_Insert	99ms	100 カラムのレコードの INSERT 実行時
	DB_Select	72ms	10 万件のレコードから 1 件のレコードを検索し、100 カラム取り出したとき*1
SQL Server 2012	DB_Insert	312ms	100 カラムのレコードの INSERT 実行時
	DB_Select	63ms	10 万件のレコードから 1 件のレコードを検索し、100 カラム取り出したとき*1
DB2 10.5	DB_Insert	145ms	100 カラムのレコードの INSERT 実行時
	DB_Select	395ms	10 万件のレコードから 1 件のレコードを検索し、100 カラム取り出したとき*1
MySQL 5.6 ストレージエンジン：InnoDB	DB_Insert	130ms	100 カラムのレコードの INSERT 実行時
	DB_Select	245ms	10 万件のレコードから 1 件のレコードを検索し、100 カラム取り出したとき*1
Firebird 2.5	DB_Insert	162ms	100 カラムのレコードの INSERT 実行時
	DB_Select	450ms	10 万件のレコードから 1 件のレコードを検索し、100 カラム取り出したとき*1
PostgreSQL 9.4	DB_Insert	277ms	100 カラムのレコードの INSERT 実行時
	DB_Select	379ms	10 万件のレコードから 1 件のレコードを検索し、100 カラム取り出したとき*1

*1. 抽出条件に主キーを指定

- 形 NX502-1□00

DB の種別	命令	命令実行時間の参考値	測定条件
Oracle Database 19c	DB_Insert	73ms	100 カラムのレコードの INSERT 実行時
	DB_Select	83ms	10 万件のレコードから 1 件のレコードを検索し、100 カラム取り出したとき*1
SQL Server 2019	DB_Insert	46ms	100 カラムのレコードの INSERT 実行時
	DB_Select	45ms	10 万件のレコードから 1 件のレコードを検索し、100 カラム取り出したとき*1
MySQL 8.0 ストレージエンジン：InnoDB	DB_Insert	69ms	100 カラムのレコードの INSERT 実行時
	DB_Select	10ms	10 万件のレコードから 1 件のレコードを検索し、100 カラム取り出したとき*1
PostgreSQL 14	DB_Insert	32ms	100 カラムのレコードの INSERT 実行時
	DB_Select	50ms	10 万件のレコードから 1 件のレコードを検索し、100 カラム取り出したとき*1

*1. 抽出条件に主キーを指定

- 形 NX102-□□20

DB の種別	命令	命令実行時間の参考値	測定条件
Oracle Database 11g	DB_Insert	227ms	100 カラムのレコードの INSERT 実行時
	DB_Select	195ms	10 万件のレコードから 1 件のレコードを検索し、100 カラム取り出したとき*1
SQL Server 2012	DB_Insert	218ms	100 カラムのレコードの INSERT 実行時
	DB_Select	163ms	10 万件のレコードから 1 件のレコードを検索し、100 カラム取り出したとき*1

DB の種別	命令	命令実行時間の参考値	測定条件
DB2 10.5	DB_Insert	428ms	100 カラムのレコードの INSERT 実行時
	DB_Select	457ms	10 万件のレコードから 1 件のレコードを検索し、100 カラム取り出したとき*1
MySQL 5.6 ストレージエンジン：InnoDB	DB_Insert	245ms	100 カラムのレコードの INSERT 実行時
	DB_Select	220ms	10 万件のレコードから 1 件のレコードを検索し、100 カラム取り出したとき*1
Firebird 2.5	DB_Insert	202ms	100 カラムのレコードの INSERT 実行時
	DB_Select	318ms	10 万件のレコードから 1 件のレコードを検索し、100 カラム取り出したとき*1
PostgreSQL 9.4	DB_Insert	287ms	100 カラムのレコードの INSERT 実行時
	DB_Select	265ms	10 万件のレコードから 1 件のレコードを検索し、100 カラム取り出したとき*1

*1. 抽出条件に主キーを指定

- 形 NJ501-□□20

DB の種別	命令	命令実行時間の参考値	測定条件
Oracle Database 11g	DB_Insert	124ms	100 カラムのレコードの INSERT 実行時
	DB_Select	175ms	10 万件のレコードから 1 件のレコードを検索し、100 カラム取り出したとき*1
SQL Server 2012	DB_Insert	136ms	100 カラムのレコードの INSERT 実行時
	DB_Select	130ms	10 万件のレコードから 1 件のレコードを検索し、100 カラム取り出したとき*1
DB2 10.5	DB_Insert	315ms	100 カラムのレコードの INSERT 実行時
	DB_Select	839ms	10 万件のレコードから 1 件のレコードを検索し、100 カラム取り出したとき*1
MySQL 5.6 ストレージエンジン：InnoDB	DB_Insert	62ms	100 カラムのレコードの INSERT 実行時
	DB_Select	38ms	10 万件のレコードから 1 件のレコードを検索し、100 カラム取り出したとき*1
Firebird 2.5	DB_Insert	35ms	100 カラムのレコードの INSERT 実行時
	DB_Select	175ms	10 万件のレコードから 1 件のレコードを検索し、100 カラム取り出したとき*1
PostgreSQL 9.4	DB_Insert	87ms	100 カラムのレコードの INSERT 実行時
	DB_Select	111ms	10 万件のレコードから 1 件のレコードを検索し、100 カラム取り出したとき*1

*1. 抽出条件に主キーを指定

- 形 NJ101-□□20

DB の種別	命令	命令実行時間の参考値	測定条件
Oracle Database 11g	DB_Insert	219ms	100 カラムのレコードの INSERT 実行時
	DB_Select	406ms	10 万件のレコードから 1 件のレコードを検索し、100 カラム取り出したとき*1
SQL Server 2012	DB_Insert	213ms	100 カラムのレコードの INSERT 実行時
	DB_Select	248ms	10 万件のレコードから 1 件のレコードを検索し、100 カラム取り出したとき*1
DB2 10.5	DB_Insert	373ms	100 カラムのレコードの INSERT 実行時
	DB_Select	395ms	10 万件のレコードから 1 件のレコードを検索し、100 カラム取り出したとき*1

DB の種別	命令	命令実行時間の参考値	測定条件
MySQL 5.6 ストレージエンジン：InnoDB	DB_Insert	219ms	100 カラムのレコードの INSERT 実行時
	DB_Select	245ms	10 万件のレコードから 1 件のレコードを検索し、100 カラム取り出したとき*1
Firebird 2.5	DB_Insert	162ms	100 カラムのレコードの INSERT 実行時
	DB_Select	450ms	10 万件のレコードから 1 件のレコードを検索し、100 カラム取り出したとき*1
PostgreSQL 9.4	DB_Insert	277ms	100 カラムのレコードの INSERT 実行時
	DB_Select	379ms	10 万件のレコードから 1 件のレコードを検索し、100 カラム取り出したとき*1

*1. 抽出条件に主キーを指定

● 処置対策

「A-1-3 DB 接続命令の実行時間の測定方法 (P.A-12)」を参照し、DB 接続命令の実行時間を測定してください。DB 接続命令の実行時間が、装置の許容範囲を超える場合は、以下の対策を行ってください。

1 **DB 接続命令に命令実行タイムアウトを設定してください。詳細は、「A-2-4 命令実行タイムアウト監視による装置性能（タクト時間）の確保 (P.A-24)」を参照してください。**

2 **DB_CreateMapping 命令実行後に、本稼働前の準備処理として、ダミーの DB_Insert 命令を 1 回実行してください。**

事象 2：DB_CreateMapping 命令実行後、初回の DB_Insert 命令の実行時間が長くなった

● 発生要因（推定原因）

発生要因（推定原因）として、以下の要因が考えられます。

- DB_CreateMapping 命令実行後の初回の DB_Insert 命令は、DB 側の処理時間が、通常時よりも長くなる傾向があります。

DB_CreateMapping 命令実行後の初回の DB_Insert 命令の実行時間の参考値は以下のとおりです。タスク実行時間割合は、50%とします。

- 形 NX701-□□20

DB の種別	命令	命令実行時間の参考値	測定条件
Oracle Database 11g	DB_Insert	3.32ms	100 カラムのレコードの INSERT 実行時
SQL Server 2012	DB_Insert	6.04ms	100 カラムのレコードの INSERT 実行時
DB2 10.5	DB_Insert	86.08ms	100 カラムのレコードの INSERT 実行時
MySQL 5.6 ストレージエンジン：InnoDB	DB_Insert	21.13ms	100 カラムのレコードの INSERT 実行時
Firebird 2.5	DB_Insert	5.31ms	100 カラムのレコードの INSERT 実行時
PostgreSQL 9.4	DB_Insert	8.69ms	100 カラムのレコードの INSERT 実行時

- 形 NX502-1□00

DB の種別	命令	命令実行時間の参考値	測定条件
Oracle Database 19c	DB_Insert	5.8ms	100 カラムのレコードの INSERT 実行時

DB の種別	命令	命令実行時間の参考値	測定条件
SQL Server 2019	DB_Insert	8.1ms	100 カラムのレコードの INSERT 実行時
MySQL 8.0 ストレージエンジン：InnoDB	DB_Insert	6.5ms	100 カラムのレコードの INSERT 実行時
PostgreSQL 14	DB_Insert	9.8ms	100 カラムのレコードの INSERT 実行時

・形 NX102-□□20

DB の種別	命令	命令実行時間の参考値	測定条件
Oracle Database 11g	DB_Insert	18.6ms	100 カラムのレコードの INSERT 実行時
SQL Server 2012	DB_Insert	18.32ms	100 カラムのレコードの INSERT 実行時
DB2 10.5	DB_Insert	24.37ms	100 カラムのレコードの INSERT 実行時
MySQL 5.6 ストレージエンジン：InnoDB	DB_Insert	35.77ms	100 カラムのレコードの INSERT 実行時
Firebird 2.5	DB_Insert	27.66ms	100 カラムのレコードの INSERT 実行時
PostgreSQL 9.4	DB_Insert	22.22ms	100 カラムのレコードの INSERT 実行時

・形 NJ501-□□20

DB の種別	命令	命令実行時間の参考値	測定条件
Oracle Database 11g	DB_Insert	29.9ms	100 カラムのレコードの INSERT 実行時
SQL Server 2012	DB_Insert	17.5ms	100 カラムのレコードの INSERT 実行時
DB2 10.5	DB_Insert	26.4ms	100 カラムのレコードの INSERT 実行時
MySQL 5.6 ストレージエンジン：InnoDB	DB_Insert	41.7ms	100 カラムのレコードの INSERT 実行時
Firebird 2.5	DB_Insert	22.5ms	100 カラムのレコードの INSERT 実行時
PostgreSQL 9.4	DB_Insert	14.1ms	100 カラムのレコードの INSERT 実行時

・形 NJ101-□□20

DB の種別	命令	命令実行時間の参考値	測定条件
Oracle Database 11g	DB_Insert	28.2ms	100 カラムのレコードの INSERT 実行時
SQL Server 2012	DB_Insert	35.6ms	100 カラムのレコードの INSERT 実行時
DB2 10.5	DB_Insert	52.7ms	100 カラムのレコードの INSERT 実行時
MySQL 5.6 ストレージエンジン：InnoDB	DB_Insert	59.3ms	100 カラムのレコードの INSERT 実行時
Firebird 2.5	DB_Insert	32.6ms	100 カラムのレコードの INSERT 実行時
PostgreSQL 9.4	DB_Insert	32.1ms	100 カラムのレコードの INSERT 実行時

● 処置対策

1 「A-1-3 DB 接続命令の実行時間の測定方法 (P.A-12)」を参照し、DB 接続命令の実行時間を測定してください。DB 接続命令の実行時間が、装置の許容範囲を超える場合は、以下の対策を行ってください。

・DB_CreateMapping 命令実行後に、本稼働前の準備処理として、ダミーの DB_Insert 命令を 1 回実行してください。

事象 3：CPU ユニットで通信処理やメモリカード処理を実行したときに DB 接続命令の実行時間が長くなった

● **発生要因（推定原因）**

発生要因（推定原因）として、以下の要因が考えられます。

- 通信処理やメモリカード処理の実行によって、システムサービス内の DB 接続サービスの処理時間が十分に確保できていない可能性があります。

● **処置対策**

1 **「A-1-4 システムサービス実行時間割合の目安 (P.A-12)」を参照して、システムサービス実行時間が十分に確保できるようにタスク設計の見直しを実施してください。**

事象 4：定常的に DB 接続命令の実行時間が長くなることがある

● **発生要因（推定原因）**

発生要因（推定原因）として、以下の要因が考えられます。

- システムサービスの実行時間が十分に確保できていない可能性があります。

● **処置対策**

1 **「A-1-4 システムサービス実行時間割合の目安 (P.A-12)」を参照して、システムサービス実行時間が十分に確保できるようにタスク設計の見直しを実施してください。**

事象 5：サーバ側 DB の状況により、DB 接続命令（DB_Insert 命令、DB_Update 命令、DB_Select 命令、DB_Delete 命令）の実行時間が長くなった

● **発生要因（推定原因）**

発生要因（推定原因）として、以下の要因が考えられます。

- サーバの負荷が一時的に上がった。
- サーバ側 DB の対象のテーブルに格納されているレコード数が多い。
- サーバ側 DB の対象のテーブルに対して、テーブルのクリアを実行した。
- サーバ側 DB の対象のテーブルが一時的にロックされた。

● **処置対策**

1 **「A-2-3 DB 応答時間の測定方法 (P.A-23)」を参照して、サーバ側 DB の処理時間を測定してください。**

2 **サーバ側 DB の処理時間が長くなったタイミングから、要因を推定して、サーバ側で対策を行ってください。**

A-2-2　稼働ログ記録の DB 接続命令実行時間への影響

稼働ログを記録する場合、DB 接続命令（DB_Insert 命令、DB_Update 命令、DB_Select 命令、DB_Delete 命令）の実行時間が長くなります。
稼働ログ記録中の DB 接続命令の実行時間の増加分の参考値は以下の通りです。タスク実行時間割合は、50%とします。

稼働ログの記録は、設備に影響がないかを確認したうえで行ってください。

• 形 NX701-□□20

ログの種別	命令	命令実行時間の増加分の参考値	測定条件
実行ログ	DB_Insert	+0.1ms	100 カラムのレコードの INSERT 実行時
デバッグログ	DB_Insert	+0.5ms	100 カラムのレコードの INSERT 実行時

• 形 NX502-1□00

ログの種別	命令	命令実行時間の増加分の参考値	測定条件
実行ログ	DB_Insert	+0.2ms	100 カラムのレコードの INSERT 実行時
デバッグログ	DB_Insert	+0.3ms	100 カラムのレコードの INSERT 実行時

• 形 NX102-□□20

ログの種別	命令	命令実行時間の増加分の参考値	測定条件
実行ログ	DB_Insert	+1.0ms	100 カラムのレコードの INSERT 実行時
デバッグログ	DB_Insert	+5.7ms	100 カラムのレコードの INSERT 実行時

• 形 NJ501-□□20

ログの種別	命令	命令実行時間の増加分の参考値	測定条件
実行ログ	DB_Insert	+1.4ms	100 カラムのレコードの INSERT 実行時
デバッグログ	DB_Insert	+3.3ms	100 カラムのレコードの INSERT 実行時

• 形 NJ101-□□20

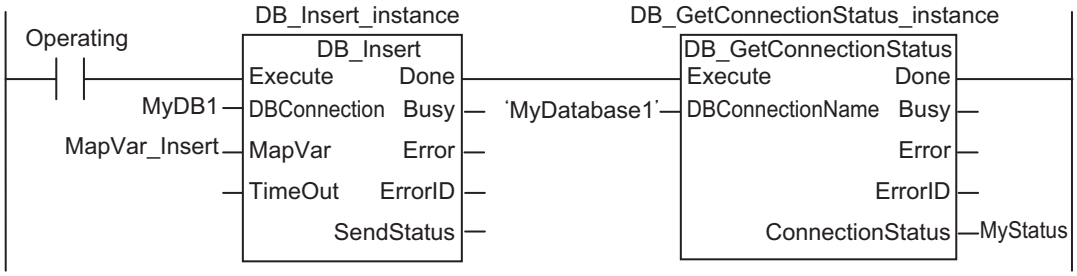
ログの種別	命令	命令実行時間の増加分の参考値	測定条件
実行ログ	DB_Insert	+2.0ms	100 カラムのレコードの INSERT 実行時
デバッグログ	DB_Insert	+7.6ms	100 カラムのレコードの INSERT 実行時

A-2-3　DB 応答時間の測定方法

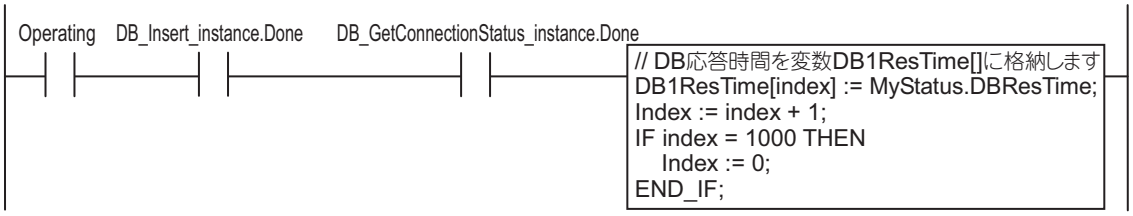
DB 応答時間とは、CPU ユニットから DB に SQL 文を送信してから、CPU ユニットに SQL 実行結果が返ってくるまでの時間です。SQL 文を送信する命令を実行した後に DB_GetConnectionStatus 命令を実行することで、DB 応答時間を取得できます。

ユーザプログラムの例を以下に示します。

• DB_Insert 命令の DB 応答時間の測定例
DB_Insert 命令の DB 応答時間を取得します



正常終了時の処理

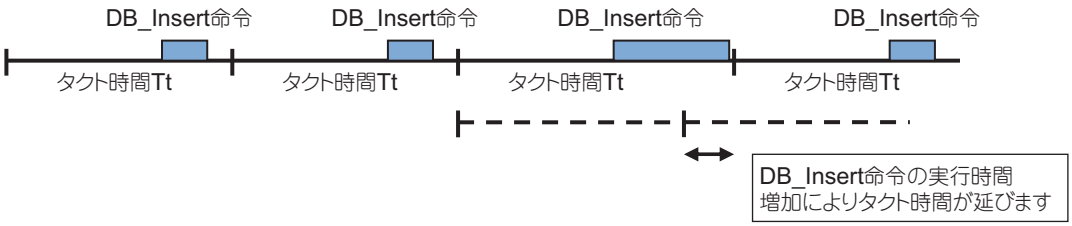


また、実行ログおよびデバッグログでも DB 応答時間を確認できます。

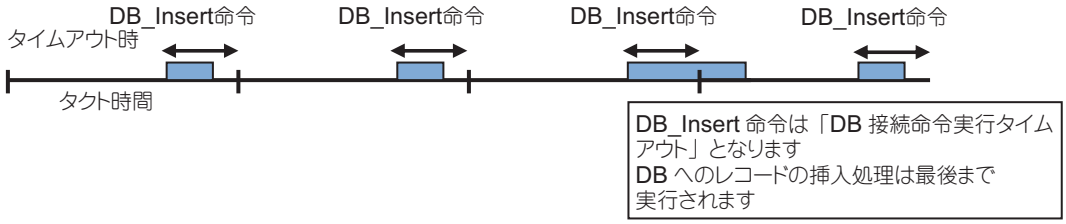
A-2-4　命令実行タイムアウト監視による装置性能（タクト時間）の確保

DB 接続命令の実行時間が増加したときにも、装置性能（タクト時間）を延ばしたくない場合は、命令に命令実行タイムアウト時間を設定します。
命令の命令実行タイムアウト時間は、DB_Insert 命令、DB_Update 命令、DB_Select 命令、DB_Delete 命令の入力変数 TimeOut で指定します。
命令の命令実行タイムアウト時間には、タクト時間のうち、DB アクセスに使用できる最長の時間を指定します。
例えば、タクトの最後に、DB_Insert 命令で DB に生産データを格納するような装置では、DB_Insert 命令に命令実行タイムアウト時間を設定することで、DB へのレコードの挿入処理がタクト時間内に完了しない場合でも、DB_Insert 命令は DB 接続命令実行タイムアウトとなり実行を完了します。このとき、DB へのレコードの挿入処理は最後まで実行します。
命令の命令実行タイムアウト時間を設定することで、一時的に DB 接続命令の実行時間が増加しても、装置性能（タクト時間）を延ばすことなく稼働し続けることが可能となります。

- 命令実行タイムアウト時間指定なし



- 命令実行タイムアウト時間指定あり



使用上の注意

- DB_Select 命令で DB 接続命令実行タイムアウトが発生した場合は、入出力変数 MapVar に取得したレコードの値は格納されません。
- DB 接続命令実行タイムアウトが定常的に発生する場合は、タスク設計の見直し、DB を配置しているサーバ環境の見直しを実施してください。

A-3 仕様

ここでは、データベース接続 CPU ユニットの仕様について説明しています。

A-3-1 一般仕様

以下のマニュアルを参照してください。
・『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル ソフトウェア編（SBCA-467)』

A-3-2 性能仕様

以下のマニュアルを参照してください。
・『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル ソフトウェア編（SBCA-467)』

A-3-3 機能仕様

下記以外の機能仕様については、以下のマニュアルを参照してください。
・『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル ソフトウェア編 （SBCA-467)』

・ NJ/NX シリーズ CPU ユニット共通の機能

項目			形 NX701-□□20	形 NX502-1□□00	形 NX102-□□20	形 NJ501-□□20		形 NJ501-4320	形 NJ101-□□20
						Ver.1.07 以前	Ver.1.08 以降		
デバッグ機能	データトレース	同時起動最大数	4	4	2*1	4	2*1	2*1	2*1

*1. データトレース関連の命令実行時、トレース番号に 2 以上を指定した場合、命令は異常（領域範囲外指定）となります。ENO は FALSE となります。

・ DB 接続サービスの機能
DB 接続サービスの仕様の詳細は、「1-2-1 DB 接続サービスの仕様 (P.1-4)」を参照してください。

A-4 バージョン関連情報

CPU ユニットと Sysmac Studio とのバージョンの対応、およびユニットバージョンごとの DB 接続機能の追加と変更について説明します。

A-4-1 ユニットバージョンと DB 接続サービスのバージョン

CPU ユニットのユニットバージョンと DB 接続サービスのバージョンの関係を以下に示します。

● 形 NX701-□□20 の場合

CPU ユニットのユニットバージョン	DB 接続サービスのバージョン
Ver.1.35 以降	Ver.2.04
Ver.1.29 以降	Ver.2.02
Ver.1.26 以降	Ver.2.01
Ver.1.21 以降	Ver.2.00
Ver.1.16 以降	Ver.1.03

● 形 NX502-1□00 の場合

CPU ユニットのユニットバージョン	DB 接続サービスのバージョン
Ver.1.65 以降	Ver.2.04
Ver.1.60 以降	Ver.2.03

● 形 NX102-□□20 の場合

CPU ユニットのユニットバージョン	DB 接続サービスのバージョン
Ver.1.68 以降	Ver.2.04
Ver.1.50 以降	Ver.2.02
Ver.1.37 以降	Ver.2.01
Ver.1.33 以降	Ver.2.00
Ver.1.30 以降	Ver.1.04

● 形 NJ501-1□20、形 NJ501-4□20、または形 NJ101-□□20 の場合

CPU ユニットのユニットバージョン	DB 接続サービスのバージョン
Ver.1.69 以降	Ver.2.04
Ver.1.50 以降	Ver.2.02
Ver.1.26 以降	Ver.2.01
Ver.1.23 以降	Ver.2.00
Ver.1.10 以降*1	Ver.1.02
Ver.1.10*1	Ver.1.01
Ver.1.09	
Ver.1.08	
Ver.1.07 以前	Ver.1.00

*1. CPU ユニットのユニットバージョンが Ver.1.10 の場合、DB 接続サービスのバージョン Ver.1.01 または Ver.1.02 を搭載しています。Sysmac Studio でオンライン時に［生産情報］ダイアログボックスで確認できます。CPU ユニットのバージョン、および DB 接続サービスのバージョンの確認方法については、「バージョン (P.21)」を参照してください。

● **形 NJ501-R□20 の場合**

CPU ユニットのユニットバージョン	DB 接続サービスのバージョン
Ver.1.46 以降	Ver.2.01
Ver.1.43 以降	Ver.2.00

A-4-2　ユニットバージョンによる DB 接続機能の追加と変更

CPU ユニットのバージョンアップにより、追加または変更された DB 接続機能を紹介します。

機能仕様の追加と変更

追加または変更された機能仕様ごとに、CPU ユニットのユニットバージョンと Sysmac Studio のバージョンを示します。
下記以外の機能仕様については、以下のマニュアルを参照してください。
・『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル ソフトウェア編（SBCA-467）』

● **形 NX701-□□20 の場合**

機能	追加／変更	ユニットバージョン	Sysmac Studio のバージョン	参照先
ストアドプロシージャ	追加	Ver.1.21	Ver.1.29 以降	「5-3 ストアドプロシージャ呼出機能 (P.5-15)」
バッチインサート	追加			「5-4 バッチインサート機能 (P.5-23)」

● **形 NX102-□□20 の場合**

機能	追加／変更	ユニットバージョン	Sysmac Studio のバージョン	参照先
ストアドプロシージャ	追加	Ver.1.33	Ver.1.29 以降	「5-3 ストアドプロシージャ呼出機能 (P.5-15)」
バッチインサート	追加			「5-4 バッチインサート機能 (P.5-23)」

● **形 NJ501-□□20 または形 NJ101-□□20 の場合**

機能		追加／変更	ユニットバージョン	Sysmac Studio のバージョン	参照先
DB コネクション設定	データベース種別	変更	Ver.1.08	Ver.1.09	「2-2-2 DB コネクション設定 (P.2-7)」
		変更	Ver.1.10	Ver.1.14	
DB コネクションの状態	SQL ステータス *1	変更	Ver.1.08	ー	「4-3-4 各 DB コネクションの状態の確認 (P.4-11)」
	エラーコード*1				
	エラーメッセージ*1				

*1.　SQLServer 接続時のエラー情報が変更されています。

A-4-3 ユニットバージョン／DB 接続サービスのバージョンと Sysmac Studio の設定ユニットバージョン

CPU ユニットのユニットバージョン／DB 接続サービスのバージョンと Sysmac Studio のバージョンには、対応関係があります。

ユニットバージョンと Sysmac Studio の対応バージョン

CPU ユニットのユニットバージョン／DB 接続サービスのバージョンと、設定可能な Sysmac Studio のバージョンの表を、DB 接続サービスのバージョンが変更されている場合について、以下に示します。CPU ユニットのユニットバージョンと設定可能な Sysmac Studio のバージョンのすべての組み合わせについては、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル ソフトウェア編 (SBCA-467)』を参照してください。

● 形 NX701-□□20 の場合

CPU ユニットのユニットバージョン	DB 接続サービスのバージョン	設定可能な Sysmac Studio のバージョン
Ver.1.35 以降	Ver.2.04	Ver.1.62 以降
Ver.1.29 以降	Ver.2.02	Ver.1.52 以降
Ver.1.26 以降	Ver.2.01	Ver.1.46 以降
Ver.1.21 以降	Ver.2.00	Ver.1.29 以降
Ver.1.16 以降	Ver.1.03	Ver.1.21 以降

● 形 NX502-1□00 の場合

CPU ユニットのユニットバージョン	DB 接続サービスのバージョン	設定可能な Sysmac Studio のバージョン
Ver.1.65 以降	Ver.2.04	Ver.1.57 以降
Ver.1.60 以降	Ver.2.03	Ver.1.54 以降

● 形 NX102-□□20 の場合

CPU ユニットのユニットバージョン	DB 接続サービスのバージョン	設定可能な Sysmac Studio のバージョン
Ver.1.68 以降	Ver.2.04	Ver.1.62 以降
Ver.1.60 以降	Ver.2.02	Ver.1.50 以降
Ver.1.37 以降	Ver.2.01	Ver.1.46 以降
Ver.1.33 以降	Ver.2.00	Ver.1.29 以降
Ver.1.30 以降	Ver.1.04	Ver.1.24 以降

● 形 NJ501-1□20、形 NJ501-4□20、または形 NJ101-□□20 の場合

CPU ユニットのユニットバージョン	DB 接続サービスのバージョン	設定可能な Sysmac Studio のバージョン
Ver.1.69 以降	Ver.2.04	Ver.1.62 以降 *1
Ver.1.50 以降	Ver.2.02	Ver.1.52 以降
Ver.1.26 以降	Ver.2.01	Ver.1.46 以降
Ver.1.23 以降	Ver.2.00	Ver.1.41 以降
Ver.1.10 以降	Ver.1.02	Ver.1.14 以降 *2*3
	Ver.1.01	Ver.1.13 以降 *2*3
Ver.1.08		Ver.1.09 以降

CPU ユニットのユニットバージョン	DB 接続サービスのバージョン	設定可能な Sysmac Studio のバージョン
Ver.1.05	Ver.1.00	Ver.1.06 以降

*1. CPU ユニット形 NJ101-□□20 には、Sysmac Studio Ver.1.63 以降を使用してください。
*2. Sysmac Studio でユニットバージョンを設定する場合、同一ユニットバージョンに複数の DB 接続サービスのバージョンが存在するため、ユニットバージョンと DB 接続サービスのバージョンが表示されます。 たとえば、CPU のユニットバージョンが Ver.1.10 で、DB 接続サービスのバージョンが Ver.1.02 のプロジェクトとする場合は、“バージョン”に“1.10(DBCon 1.02)”を設定してください。



*3. CPU ユニット形 NJ101-□□20 には、Sysmac Studio Ver.1.14 以降を使用してください。Sysmac Studio Ver.1.13 以前では形 NJ101-□□20 を使用できません。

● 形 NJ501-R□20 の場合

CPU ユニットのユニットバージョン	DB 接続サービスのバージョン	設定可能な Sysmac Studio のバージョン
Ver.1.46 以降	Ver.2.01	Ver.1.46 以降
Ver.1.43 以降	Ver.2.00	Ver.1.44 以降

DB 接続サービスのバージョンと Sysmac Studio での設定ユニットバージョンの関係

CPU ユニットの DB 接続サービスのバージョンと Sysmac Studio プロジェクトの DB 接続サービスのバージョンの組み合わせでの仕様の差異は、以下のとおりです。
以下のケース以外では、仕様の差異を考慮する必要がありません。

● 接続可能なデータベース種別

・ 形 NJ501-1□20 または形 NJ101-□□20 の場合

CPU ユニットの DB 接続サービスのバージョン	Sysmac Studio プロジェクトの DB 接続サービスのバージョン		
	Ver.1.00*1	Ver.1.01	Ver.1.02 以降
Ver.1.02 以降	Oracle SQL Server	Oracle SQL Server DB2 MySQL Firebird	Oracle SQL Server DB2 MySQL Firebird PostgreSQL
Ver.1.01	Oracle SQL Server	Oracle SQL Server DB2 MySQL Firebird	転送不可
Ver.1.00*1	Oracle SQL Server	転送不可	転送不可

*1. 形 NJ501-1□20 の場合だけ

● マッピング可能な DB マップ変数の最大数

• 形 NX701-1□20 の場合

CPU ユニットの DB 接続サービスのバージョン	Sysmac Studio プロジェクトの DB 接続サービスのバージョン	
	Ver.1.03	Ver.2.00 以降
Ver.2.00 以降	SQL Server と接続する場合：60 Oracle と接続する場合: 30 DB2 と接続する場合: 30 MySQL と接続する場合: 30 Firebird と接続する場合: 15 PostgreSQL と接続する場合: 30	同左
Ver.1.03		転送不可

• 形 NX102-□□20 の場合

CPU ユニットの DB 接続サービスのバージョン	Sysmac Studio プロジェクトの DB 接続サービスのバージョン	
	Ver.1.04	Ver.2.00 以降
Ver.2.00 以降	15	SQL Server と接続する場合: 30 Oracle と接続する場合: 20 DB2 と接続する場合: 20 MySQL と接続する場合: 20 Firebird と接続する場合: 15 PostgreSQL と接続する場合: 20
Ver.1.04		転送不可

• 形 NJ501-□□20 の場合

CPU ユニットの DB 接続サービスのバージョン	Sysmac Studio プロジェクトの DB 接続サービスのバージョン			
	Ver.1.00	Ver.1.01	Ver.1.02 以降	Ver.2.00 以降
Ver.2.00 以降	15	SQL Server と接続する場合: 60 Oracle と接続する場合: 30 DB2 と接続する場合: 30 MySQL と接続する場合: 30 Firebird と接続する場合: 15	SQL Server と接続する場合: 60 Oracle と接続する場合: 30 DB2 と接続する場合: 30 MySQL と接続する場合: 30 Firebird と接続する場合: 15 PostgreSQL と接続する場合: 30	同左
Ver.1.02 以降				転送不可
Ver.1.01			転送不可	
Ver.1.00		転送不可	転送不可	

• 形 NJ501-4320 の場合

CPU ユニットの DB 接続サービスのバージョン	Sysmac Studio プロジェクトの DB 接続サービスのバージョン	
	Ver.1.00 以降	Ver.2.00 以降
Ver.2.00 以降	SQL Server と接続する場合: 15 Oracle と接続する場合: 15 MySQL と接続する場合: 15	SQL Server と接続する場合: 60 Oracle と接続する場合: 30 MySQL と接続する場合: 30
Ver.1.00 以降		転送不可

• 形 NJ101-1□20 の場合

CPU ユニットの DB 接続サービスのバージョン	Sysmac Studio プロジェクトの DB 接続サービスのバージョン		
	Ver.1.01	Ver.1.02 以降	Ver.2.00 以降
Ver.2.00 以降	15	15	同左
Ver.1.02 以降			転送不可
Ver.1.01		転送不可	

● DB コネクション数

• 形 NX102-□□20 の場合

CPU ユニットの DB 接続サービスのバージョン	Sysmac Studio プロジェクトの DB 接続サービスのバージョン	
	Ver.1.04	Ver.2.00 以降
Ver.2.00 以降	1	2
Ver.1.04		転送不可

A-4-4 DB 接続サービスのバージョンとデバイス変更後の接続データベース種別

Sysmac Studio の［デバイス変更］を実行して、CPU ユニットのユニットバージョンを変更した場合、それに伴って DB 接続サービスのバージョンが下がることがあります。DB 接続サービスのバージョンが下がると、接続データベース種別が自動的に変わる可能性があります。
デバイス変更後の接続データベース種別は、デバイス変更後の CPU ユニットの DB 接続サービスのバージョンと、デバイス変更後の CPU ユニット形式、およびデバイス変更前の接続データベース種別によって、つぎのように決まります。

デバイス変更後の CPU ユニットの、DB 接続サービスのバージョン	デバイス変更後の CPU ユニット形式	デバイス変更前の接続データベース種別	デバイス変更後の接続データベース種別
Ver.1.02 以降	形 NJ501-4□20	Oracle SQL Server MySQL	左記と同じ
		上記以外	SQL Server
	形 NJ501-4□20 以外	全種類	左記と同じ
Ver.1.01	形 NJ501-4□20	Oracle SQL Server MySQL	左記と同じ
		上記以外	SQL Server
	形 NJ501-4□20 以外	Oracle SQL Server DB2 MySQL Firebird	左記と同じ
		上記以外	SQL Server
Ver.1.00	全形式	Oracle SQL Server	左記と同じ
		上記以外	SQL Server

A-4-5 DB 接続サービスのバージョンと接続データベースの種別およびバージョン

DB 接続サービスのバージョンと接続データベースの種別およびバージョンの関係は、以下のとおりです。

● 形 NX701-□□20 の場合

DB 接続サービスのバージョン (DBcon)	SQL Server	Oracle Database	DB2 for Linux, UNIX and Windows	MySQL Community Edition	Firebird	PostgreSQL
Ver.1.03	2008, 2008R2, 2012, 2014, 2016	10g, 11g, 12c	9.5, 9.7, 10.1, 10.5, 11.1	5.1, 5.5, 5.6, 5.7	2.1, 2.5	9.2, 9.3, 9.4, 9.5, 9.6

DB 接続サービスのバージョン (DBcon)	SQL Server	Oracle Database	DB2 for Linux, UNIX and Windows	MySQL Community Edition	Firebird	PostgreSQL
Ver.2.00	2012, 2014, 2016, 2017	11g, 12c*1, 18c	9.7, 10.1, 10.5, 11.1	5.6, 5.7, 8.0	2.5	9.4, 9.5, 9.6, 10
Ver.2.01, Ver.2.02	2012, 2014, 2016, 2017, 2019	11g, 12c*1, 18c, 19c	9.7, 10.1, 10.5, 11.1	5.6, 5.7, 8.0	2.5	9.4, 9.5, 9.6, 10, 11, 12, 13
Ver.2.04 以降	2012, 2014, 2016, 2017, 2019, 2022	11g, 12c*1, 18c, 19c, 21c, 23ai(23c)	9.7, 10.1, 10.5, 11.1	5.6, 5.7, 8.0	2.5	9.4, 9.5, 9.6, 10, 11, 12, 13, 14, 15, 16

*1. Release2 を含みます。

● 形 NX502-□□□□の場合

DB 接続サービスのバージョン (DBcon)	SQL Server	Oracle Database	DB2 for Linux, UNIX and Windows	MySQL Community Edition	Firebird	PostgreSQL
Ver.2.03	2019	19c, 21c	非対応	8.0	非対応	14
Ver.2.04 以降	2014, 2016, 2017, 2019, 2022	19c, 21c, 23ai(23c)	非対応	8.0	非対応	11, 12, 13, 14, 15, 16

● 形 NX102-□□20 の場合

DB 接続サービスのバージョン (DBcon)	SQL Server	Oracle Database	DB2 for Linux, UNIX and Windows	MySQL Community Edition	Firebird	PostgreSQL
Ver.1.04	2008, 2008R2, 2012, 2014, 2016, 2017	10g, 11g, 12c	9.5, 9.7, 10.1, 10.5, 11.1	5.1, 5.5, 5.6, 5.7	2.1, 2.5	9.2, 9.3, 9.4, 9.5, 9.6
Ver.2.00	2012, 2014, 2016, 2017	11g, 12c*1, 18c	9.7, 10.1, 10.5, 11.1	5.6, 5.7, 8.0	2.5	9.4, 9.5, 9.6, 10
Ver.2.01, Ver.2.02	2012, 2014, 2016, 2017, 2019	11g, 12c*1, 18c, 19c	9.7, 10.1, 10.5, 11.1	5.6, 5.7, 8.0	2.5	9.4, 9.5, 9.6, 10, 11, 12, 13
Ver.2.04 以降	2012, 2014, 2016, 2017, 2019, 2022	11g, 12c*1, 18c, 19c, 21c, 23ai(23c)	9.7, 10.1, 10.5, 11.1	5.6, 5.7, 8.0	2.5	9.4, 9.5, 9.6, 10, 11, 12, 13, 14, 15, 16

*1. Release2 を含みます。

● 形 NJ501-1□20、形 NJ501-4□20、または形 NJ101-□□20 の場合

DB 接続サービスのバージョン (DBcon)	SQL Server	Oracle Database	DB2 for Linux, UNIX and Windows*2	MySQL Community Edition	Firebird*2	PostgreSQL*2
Ver.1.00	2008, 2008R2, 2012	10g, 11g	−	−	−	−

DB 接続サービスのバージョン (DBcon)	SQL Server	Oracle Database	DB2 for Linux, UNIX and Windows*2	MySQL Community Edition	Firebird*2	PostgreSQL*2
Ver.1.01	2008, 2008R2, 2012	10g, 11g	9.5, 9.7, 10.1, 10.5	5.1, 5.5, 5.6	2.1, 2.5	−
Ver.1.02	2008, 2008R2, 2012, 2014	10g, 11g, 12c	9.5, 9.7, 10.1, 10.5	5.1, 5.5, 5.6	2.1, 2.5	9.2, 9.3, 9.4
Ver.2.00	2012, 2014, 2016, 2017	11g, 12c*1, 18c	9.7, 10.1, 10.5, 11.1	5.6, 5.7, 8.0	2.5	9.4, 9.5, 9.6, 10
Ver.2.01, Ver.2.02	2012, 2014, 2016, 2017, 2019	11g, 12c*1, 18c, 19c	9.7, 10.1, 10.5, 11.1	5.6, 5.7, 8.0	2.5	9.4, 9.5, 9.6, 10, 11, 12, 13
Ver.2.04 以降	2012, 2014, 2016, 2017, 2019, 2022	11g, 12c*1, 18c, 19c, 21c, 23ai(23c)	9.7, 10.1, 10.5, 11.1	5.6, 5.7, 8.0	2.5	9.4, 9.5, 9.6, 10, 11, 12, 13, 14, 15, 16

*1. Release2 を含みます。
*2. 形 NJ501-4320 は非対応です。

● 形 NJ501-R□20 の場合

DB 接続サービスのバージョン (DBcon)	SQL Server	Oracle Database	DB2 for Linux, UNIX and Windows	MySQL Community Edition	Firebird	PostgreSQL
Ver.2.00	2012, 2014, 2016, 2017	11g, 12c*1, 18c	9.7, 10.1, 10.5, 11.1	5.6, 5.7, 8.0	2.5	9.4, 9.5, 9.6, 10
Ver.2.01 以降	2012, 2014, 2016, 2017, 2019	11g, 12c*1, 18c, 19c	9.7, 10.1, 10.5, 11.1	5.6, 5.7, 8.0	2.5	9.4, 9.5, 9.6, 10, 11, 12, 13

*1. Release2 を含みます。

I

索引

索引

C

CA....26

D

DB....26
DB_AttachProcedure（DB ストアドプロシージャ ハンドル生成）命令....3-23, 7-99
DB_BatchInsert（DB レコード一括挿入）命令3-22, 7-87
DB_Close(DB コネクション切断）....3-22, 7-9
DB_Connect（DB コネクション接続）命令....3-22, 7-6
DB_ControlService（DB 接続サービスの制御）命令....3-23, 4-4, 7-54
DB_ControlSpool（スプールの再送・クリア)命令....3-23, 5-9, 5-10, 7-71
DB_CreateMapping....3-18
DB_CreateMapping（DB マップ生成)命令....3-2, 3-15, 3-22, 7-11
DB_Delete（DB レコード削除）命令....3-22, 7-40
DB_DetachProcedure（DB ストアドプロシージャ ハンドル解放）命令....3-23, 7-115
DB_ExecuteProcedure（DB ストアドプロシージャ 実行）命令....3-23, 7-103
DB_GetConnectionStatus（DB コネクションの状態取得）命令....3-23, 4-13, 7-65
DB_GetServiceStatus（DB 接続サービスの制御）命令....3-23, 7-60
DB_Insert（DB レコード挿入）命令....3-22, 7-15
DB_Insert（DB レコード挿入)命令....5-5
DB_PutLog（稼働ログ記録)命令....3-23, 7-78
DB_Select（DB レコード取得）命令....3-22, 7-35
DB_Shutdown（DB 接続サービスのシャットダウン）命令....3-23, 7-83
DB_Update（DB レコード更新）命令....3-22, 5-5, 7-19
DB 応答時間....4-12
DB コネクション異常切断....8-20
DB コネクション....26
DB コネクション異常切断発生中....8-36
DB コネクション既接続....8-30
DB コネクション接続拒否....8-29
DB コネクション接続失敗....8-29
DB コネクション接続数超過....8-30
DB コネクション設定....2-7
DB コネクションの状態の確認....4-11
DB コネクションの接続／切断....4-6
DB コネクションの追加....2-7
DB コネクション不正....8-31
DB コネクション名の変更....2-8
DB コネクション名不正....8-28
DB サーバ証明書エラー....8-19
DB 処理中....8-38
DB 接続機能....26
DB 接続サービス....26
DB 接続サービス異常停止中....8-37
DB 接続サービス開始....8-20
DB 接続サービスシャットダウン機能....26, 5-25
DB 接続サービス終了....8-21
DB 接続サービス終了／終了処理中....8-28
DB 接続サービス初期処理中....8-38
DB 接続サービスステータス（システム定義変数_DBC_Status）....3-23
DB 接続サービス設定....2-5
DB 接続サービス操作（開始）....8-22
DB 接続サービス操作（終了）....8-23
DB 接続サービス操作（停止）....8-22
DB 接続サービス停止....8-21
DB 接続サービス動作モード変更失敗....8-27
DB 接続サービスの状態の確認....4-8
DB 接続サービスのしくみ....1-10
DB 接続サービスの動作モード....26, 4-2
DB 接続サービスのプログラミングの概要....3-21
DB 接続サービス未開始....8-26
DB 接続サービス未起動....8-42
DB 接続システム処理異常....8-19
DB 接続設定....2-5
DB 接続設定エラー....8-18
DB 接続命令....26
DB 接続命令実行タイムアウト....8-36
DB 接続命令一覧....3-22
DB 接続命令の変数....7-2
DB 接続命令の命令カテゴリ....3-21
DB 側のテーブルのカラム名側の制約....3-4
DB にアクセスするための構造体データ型....26
DB へのマッピング....3-15
DB マッピング....26, 3-2
DB マッピングの制約....3-18
DB マップ変数....26, 3-15
DB マップ変数の制約事項....3-17
DB マップ変数の登録および属性....3-16
DB マップ変数のマッピングの解除....3-18
DB マップ変数不正....8-31
DB マップ変数未登録....8-32
DB レコード一括挿入命令....26

E

EM エリア....5-7

I

IP アドレス....2-10

K

KeepAlive 監視時間....5-34

M

Microsoft Excel からのコピー....3-12

O

Oracle Data base....1-2

S

SQL....26
SQL Server....1-2
SQL 実行エラー....8-33
SQL 実行失敗ログ....26, 2-7
SQL 実行失敗ログ保存失敗....8-17
SQL 種別....26, 3-15, 3-18
SQL ステータス....4-13
SQL 文のクリア条件....5-10
SQL 文の再送....5-9
SQL 文の手動再送....5-9

あ

暗号化通信....26

い

異常....8-5, 8-6
異常実行回数....4-12
異常切断時間....4-12
異常切断発生日時....4-12
イベントコード....8-7 – 8-13
イベント名....8-7 – 8-13

え

エラーコード（接続エラー）....4-13
エラーメッセージ（接続エラー）....4-13

か

稼働時間....4-9
稼働状況....4-9
稼働モード....2-6, 4-2
稼働ログ....26
稼働ログのクリア操作....8-24
稼働ログ無効....8-39
カラム....26
カラム名の制約....3-4

く

クエリ実行....4-12
クエリ実行タイムアウト時間....2-12

こ

構造体データ型の作成....3-3
構造体データ型の作成方法....3-11
構造体データ型の仕様....3-3
構造体名....3-3
構造体のメンバのデータ型....3-3
構造体メンバ名....3-3
構造体メンバ名の制約....3-4
コネクション状態....4-12
コネクション設定....2-8
コネクション名....2-10

さ

サーバ証明書....26
サービス開始....2-6

し

システム定義変数....3-23, 4-10
実行ログ....26, 2-6
実行ログ保存失敗....8-16

す

推定原因....8-7 – 8-13
ストアドファンクション....26
ストアドプロシージャ....26
ストアドプロシージャ引数不正....8-41
ストアドプロシージャ結果セットカラム数不正....8-41
ストアドプロシージャ名不正....8-40
ストアドプロシージャ呼出....26
スプール....26
スプール機能....26, 5-5
スプール機能専用エリア....5-6
スプール機能の設定....5-8
スプールクリア....8-21
スプールクリア操作....8-23
スプール使用率....4-13
スプール設定....2-13
スプールデータ....26
スプールデータあり....8-37
スプールデータ数....4-13
スプール破壊....8-15
スプール容量超過....8-34

せ

正常実行回数....4-12
接続時間....4-12

そ

操作権限認証....5-36

た

対象外データベース種別への命令実行....8-40

ち

抽出条件不正....8-35

つ

通信タイムアウト....5-33
通信テスト....2-12

て

テストモード..... 2-6, 4-2
デバッグログ..... 27, 2-6
デバッグログ操作（開始）.....8-24
デバッグログ操作（停止）.....8-25
電源断発生時の SQL 文消失の回避策.....5-28
データ型の対応..... 3-4
データベース種別.....2-10
テーブル.....27

と

トラブルシューティング.....8-7

は

バックアップ／リストア機能.....5-35
発生要因（推定原因）.....8-7 - 8-13
バッチインサート..... 27

ふ

プロシージャハンドル不正.....8-39

め

命令実行タイムアウト..... 5-34

れ

レコード処理.....26

ろ

ログインタイムアウト..... 5-33
ログインタイムアウト時間.....2-12
ログコード範囲外..... 8-35

オムロン株式会社 インダストリアルオートメーションビジネスカンパニー

製品に関するお問い合わせ先

お客様相談室

フリー通話 **0120-919-066**（クイック　オムロン）

携帯電話の場合、

055-982-5015（有料）をご利用ください。

受付時間：9:00～17:00（土・日・12/31～1/3を除く）

オムロンFAクイックチャット

www.fa.omron.co.jp/contact/tech/chat/

技術相談員にチャットでお問い合わせいただけます。（I-Webメンバーズ限定）

受付時間：平日9:00～12:00 / 13:00～17:00（土日祝日・年末年始・当社休業日を除く）

※受付時間、営業日は変更の可能性がございます。最新情報はリンク先をご確認ください。

その他のお問い合わせ：納期・価格・サンプル・仕様書は貴社のお取引先、または貴社担当オムロン販売員にご相談ください。オムロン制御機器販売店やオムロン販売拠点は、Webページでご案内しています。

オムロン制御機器の最新情報をご覧いただけます。緊急時のご購入にもご利用ください。 **www.fa.omron.co.jp**

●本誌に記載のない条件や環境での使用、および原子力制御・鉄道・航空・車両・燃焼装置・医療機器・娯楽機械・安全機器、その他人命や財産に大きな影響が予測されるなど、特に安全性が要求される用途に使用される際には、当社の意図した特別な商品用途の場合や特別の合意がある場合を除き、当社は当社商品に対して一切保証をいたしません。

●本製品の内、外国為替及び外国貿易法に定める輸出許可、承認対象貨物（又は技術）に該当するものを輸出（又は非居住者に提供）する場合は同法に基づく輸出許可、承認（又は役務取引許可）が必要です。

オムロン商品のご用命は

Man. No. SBCA-411AC　2025年7月現在

©OMRON Corporation 2013-2025 All Rights Reserved.
お断りなく仕様などを変更することがありますのでご了承ください