

# MESX プロバイダ MESX プロトコル 対応

Version 1.0.0

## ユーザーズ ガイド

March 31, 2014

備考:

## 【改版履歴】

バージョン	日付	内容
1.0.0	2014-3-31	初版.

## 【対応標準化技術】

機種	バージョン	説明
PSLX	PSLX2	PSLX は, APSOM (NPO 法人ものづくり APS 推進機構) で制定され, ISO や IEC といった国際規格として標準化がすすめられる技術です. PSLX の詳細は, APSOM のホームページを確認してください.
MESX	Rev8	MESX は, APSOM 内のプロジェクトであり, PSLX を基盤に MES を中心とした標準プロトコルを制定しています. MESX プロトコルは, IEC62264 の機能階層モデルに基づき, 生産活動を支援するさまざまな機能を 3 階層に配分した上で, それらの機能を実現するシステムコンポーネント間の通信メッセージと応答規則を定めたものです. これらは「MESX プロトコルブック」にまとめられています. MESX の詳細は, APSOM のホームページを確認してください.

## 目次

1. はじめに .....	5
1.1. システム構成 .....	5
1.2. 環境 .....	6
1.3. 業務ドキュメント一覧 .....	6
1.4. 業務オブジェクト一覧 .....	6
1.5. 情報関係図 .....	8
2. プロバイダの概要 .....	9
2.1. インストール .....	9
2.2. 概要 .....	9
2.3. メソッド・プロパティ .....	10
2.3.1. CaoWorkspace::AddController メソッド .....	10
3. コマンドリファレンス .....	11
3.1. Controller クラス .....	11
3.1.1. CaoController::Execute("StatusChanged") コマンド .....	11
3.1.2. CaoController::Execute("GetOrder") コマンド .....	13
3.1.3. CaoController::Execute("SendResult") コマンド .....	14
3.1.4. CaoController::Execute("ValidLot") コマンド .....	15
4. 通知データフォーマット .....	16
4.1. コマンド番号(1 要素目) .....	16
4.2. 設備名称(2 要素目) .....	16
4.3. データ本文(3 要素目以降) .....	16
5. メッセージ一覧 .....	17
5.1.1. GetStatus メッセージ .....	17
5.1.2. AddOrder メッセージ .....	17
5.1.3. SyncStatus メッセージ .....	18
5.1.4. GetResult メッセージ .....	18
6. MESX プロバイダ用 CaoSQLConfig 定義 .....	20
MESX プロバイダ使用例 .....	21

---

付録 A.1. システム構成 .....	21
付録 A.2. 設定 .....	21
付録 B. MESX プロトコル .....	22

## 1. はじめに

MESX プロバイダは、NPO 法人ものづくり APS 推進機構の技術部会である MESX ジョイントプロジェクトで規定される MESX プロトコルに準拠した XML メッセージの送受信、及び、XML の生成、解析を行う ORiN2 CAO プロバイダです。

MESX プロバイダを利用することで ORiN2 アプリケーションは、MESX 対応のシステムとリアルタイムにデータ交換することができます。また、実デバイスが MESX プロトコルに対応していない場合も、ORiN2 を介し、システムと実デバイスのリアルタイムなデータ交換をすることができます。

本ドキュメントでは、MESX プロバイダの概要と、実装されている CAO インタフェース(関数仕様)について説明しています。

### 1.1. システム構成

図 1-1 は、ORiN2 を使用したシステムと各種デバイスのシステム構成を示しています。MESX プロバイダと CaoSQL のリンク機能を使うと、プログラムレスで各種デバイスの I/O やレジスタの値をシステムへ送信したり、逆にシステムからの指示値を PLC やロボットへ送信するというようなことができます。

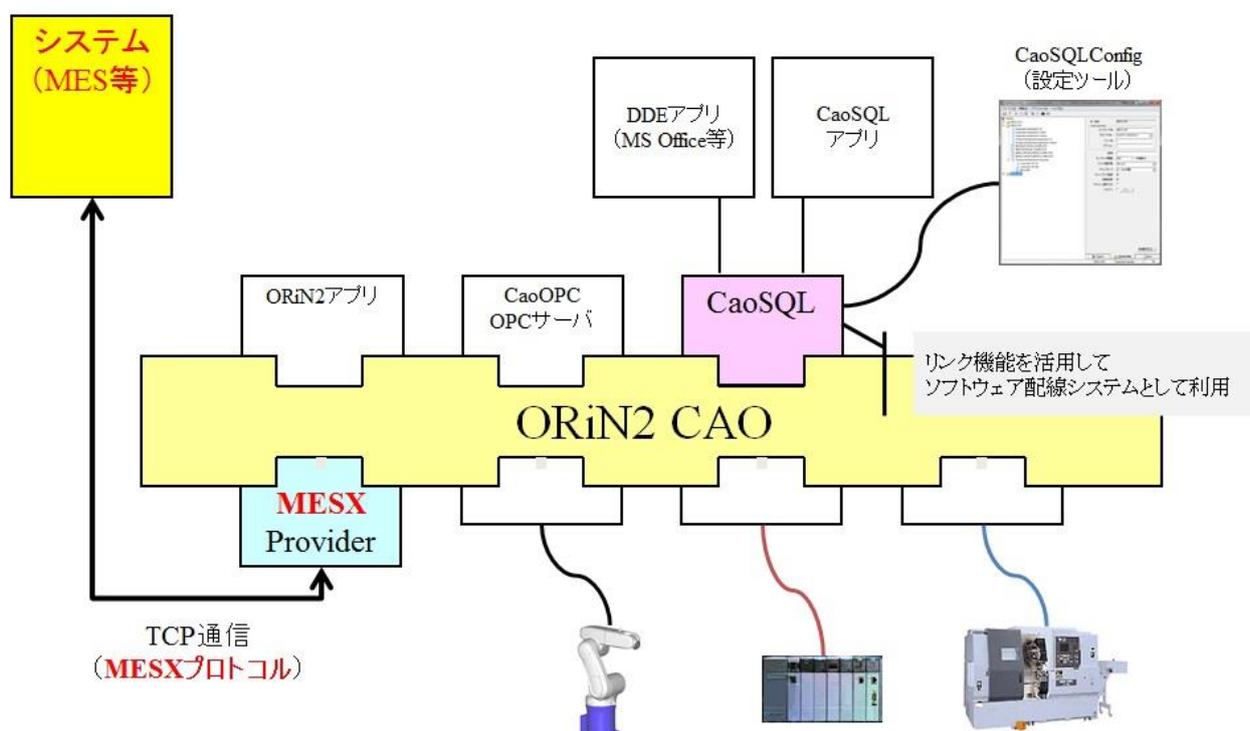


図 1-1MESX プロバイダを利用した接続方法の概要

## 1.2. 環境

本プロバイダは、以下の規約に基づいて作成されています。

表 1-1 環境

PPS スキーマ	pps-schema-1.0.xsd
MESX プロトコル	MESX プロトコルブック rev.8

## 1.3. 業務ドキュメント一覧

MESX プロトコルが対応する PSLX 業務ドキュメントの一覧を表 1-2 に記載します。

表 1-2 業務ドキュメント一覧

区分	業務ドキュメント		説明
工程管理	OperationSchedule	予定作業	生産を実施するための作業指示であり 作業場に送られる情報
	OperationRecord	実績作業	実際に作業を実施した結果として生成される情報
	LotRecord	実績ロット	実際に生産され一度は存在したロットの具体的な情報
	EquipmentState	装置状態	設備を実際に制御するうえで必要となる情報

## 1.4. 業務オブジェクト一覧

MESX プロトコルが対応する PSLX 業務オブジェクトの一覧を表 1-3 に記載します。

表 1-3 業務オブジェクト一覧

業務オブジェクト	項目名	説明
Operation	指示 ID	このオブジェクトのキーとなる識別名
	指示オーダーID	このオブジェクトに関係するオーダーの識別名
	指示資源 ID	このオブジェクトに関係する資源の識別名
	指示プロセス ID	このオブジェクトに関係するプロセスの識別名
	指示品目 ID	生成に関する品目の識別名
	生成種別	生成の種別や方法など
	生成予定数量	生成する品目、資源、作業などの予定数量
	生成予定数量単位	生成する品目、資源、作業などの予定数量単位
	生成数量	生成する品目、資源、作業などの数量
	生成数量単位	生成する品目、資源、作業などの数量単位
	消費品目 ID	消費に関する品目の識別名
	消費予定数量	消費する品目、資源、作業などの予定数量

	消費予定数量単位	消費する品目, 資源, 作業などの予定数量単位
	消費数量	消費する品目, 資源, 作業などの数量
	消費数量単位	消費する品目, 資源, 作業などの数量単位
	消費ロット ID	消費するロットの識別名
	消費ロット品目 ID	消費するロットの品目 ID
	消費ロット予定数量	ロットを消費する予定数量
	消費ロット予定数量単位	ロットを消費する予定数量の単位
	消費ロット数量	ロットを消費する数量
	消費ロット数量単位	ロットを消費する数量の単位
	割当資源 ID	割当に関する資源の識別名
	割当名称	割当る品目, 資源, 作業などの名称
	割当グループ	割当る品目, 資源, 作業などのグループ
	割当開始予定日時	割当が行われる開始予定の日時
	割当開始日時	割当が行われた開始日時
	割当完了予定日時	割当が行われる完了予定の日時
	割当完了日時	割当が行われた完了日時
	開始予定日時	処理の開始を予定する日時
	開始日時	処理の開始日時
	完了予定日時	処理の終了を予定する日時
	完了日時	処理の終了日時
	実行日時	処理の実行日時
Lot	ロット ID	このオブジェクトのキーとなる識別名
	ロットロット ID	このオブジェクトに関するロットの識別名
	消費品目 ID	消費に関する品目の識別名
	消費資源 ID	消費に関する資源の識別名
	消費作業 ID	消費に関する作業やプロセスの識別名
	消費予定数量	消費する品目, 資源, 作業などの予定数量
	消費予定数量単位	消費する品目, 資源, 作業などの予定数量単位
	消費数量	消費する品目, 資源, 作業などの数量
	消費数量単位	消費する品目, 資源, 作業などの数量単位
	結果区分	対象とする処理の実行結果に関する区分
Resource	装置 ID	このオブジェクトのキーとなる識別名
	装置状態	このオブジェクトの状態
	実行予定日時	処理の実行を予定する日時
	実行日時	処理の実行日時

## 1.5. 情報関係図

本ユーザーズガイド内で使用される MESX 用の管理データの関連図を図 1-2 にまとめます。  
ある作業を行う場合、作業を実施する設備(装置)を「割当資源」といいます。「消費品目」とは、作業を行うのに必要な「部材」や「中間品」のことです。作業の結果により生成される品目を「生成品目」といいます。



図 1-2 情報関係図

## 2. プロバイダの概要

### 2.1. インストール

MESX プロバイダモジュールは、下記の DLL で構成されています。ORiN2 SDK のインストーラでインストールした場合は、インストール作業は不要です。手動でインストールする場合は、表 2-1 のように実行してください。

表 2-1 MESX プロバイダ

ファイル名	CaoProvMESX.dll
ProgID	CaoProv.MESX
レジストリ登録	regsvr32 CaoProvMESX.dll
レジストリ登録の抹消	regsvr32 /u CaoProvMESX.dll

### 2.2. 概要

MESX プロバイダは、TCP/IP でシステム(MES 等)と通信を行います。システムへ送信を行うときは、MESX プロバイダは TCP クライアントとして動作します。また、システムから受信を行うときは、TCP サーバーとして動作します。

システムと接続する際、MESX プロバイダは 1 つの TCP ポートを開きます。クライアントアプリケーションから値の書き込みを行った時、システムに対してパケットを送信します。システムからの受信があったときには、OnMessage イベントで、そのクライアントに通知します。

MESX プロバイダで対応する MESX プロトコルは「付録 B. MESX プロトコル」に記載します。

## 2.3. メソッド・プロパティ

### 2.3.1. GaoWorkspace::AddController メソッド



AddController ( <bstrCtrlName:BSTR>, <bstrProvName:BSTR>, <bstrPcName:BSTR>, [**<bstrOption:BSTR>**] )

<bstrCtrlName> : [in] コントローラ名

<bstrProvName> : [in] プロバイダ名. 固定値 =” CaoProv. MESX”

<bstrPcName> : [in] プロバイダの実行マシン名 (未使用)

<bstrOption> : [in] オプション文字列 (未使用)

## 3. コマンドリファレンス

### 3.1. Controller クラス

表 3-1 CaoController::Execute コマンド一覧

コマンド	機能	ページ
StatusChanged	状態変更を MESX に通知します.	11
GetOrder	MESX から最新のオーダーを取得します.	13
SendResult	MESX に対して実績を通知します.	14
ValidLot	MESX にロットの投入可否を問い合わせます.	15

#### 3.1.1. CaoController::Execute(“StatusChanged”) コマンド

状態変更を MESX に通知します.



StatusChanged ( <aryData> )

< aryData > : [in] 通知を行うデータ (VT\_BSTR|VT\_ARRAY)  
データの形式は 4. 通知データフォーマットを参照

処理に使用する CaoSQL の要素は以下の通りです.

EquipmentState.equipment-status  
EquipmentState.execution-time-schedule  
EquipmentState.execution-time-record  
Operation.operation-id  
Operation.operation-order-id  
Operation.operation-resource-id  
Operation.operation-process-id  
Operation.start-time-record  
Operation.end-time-record  
Operation.execution-time-record

---

**使用例**

---

```
Dim param(0 To 11) As String
param(0) = "10"
param(1) = "MESX_PLC"
param(2) = "EquipmentState.equipment-status." & status
param(3) = "EquipmentState.execution-time-schedule, " & CStr(ctrl.item("EquipmentState.execution-time-schedule").value)
param(4) = "EquipmentState.execution-time-record, " & CStr(ctrl.item("EquipmentState.execution-time-record").value)
param(5) = "Operation.operation-id, " & CStr(ctrl.item("Operation.operation-id").value)
param(6) = "Operation.operation-order-id, " & CStr(ctrl.item("Operation.operation-order-id").value)
param(7) = "Operation.operation-resource-id, " & CStr(ctrl.item("Operation.operation-resource-id").value)
param(8) = "Operation.operation-process-id, " & CStr(ctrl.item("Operation.operation-process-id").value)
param(9) = "Operation.start-time-record, " & CStr(ctrl.item("Operation.start-time-record").value)
param(10) = "Operation.end-time-record, " & CStr(ctrl.item("Operation.end-time-record").value)
param(11) = "Operation.execution-time-record, " & CStr(ctrl.item("Operation.execution-time-record").value)
```

---

### 3.1.2. CaoController::Execute(“GetOrder”) コマンド

MESX から最新のオーダーを取得します。



GetOrder ( <aryData> )

< aryData > : [in] 通知を行うデータ (VT\_BSTR|VT\_ARRAY)  
データの形式は 4 通知データフォーマットを参照

処理に使用する CaoSQL の要素は以下の通りです。

Operation..operation-id

Operation.operation-order-id

Operation.operation-resource-id

Operation.operation-process-id

戻り値 : [out]コマンドの戻り値 (VT\_BSTR | VT\_ARRAY)  
データの形式は 4 通知データフォーマットを参照

以下の値が設定されます。

Operation から始まる計画に関する要素および

[Operation.Consume\_Plan] , [Operation.Consume\_Lot\_Plan] ,

[Operation.Assign\_Plan] 以下の各要素。



```
Dim param(0 To 5) As String
param(0) = CStr(ADD_ORDER)
param(1) = CStr(ctrl.item("EquipmentState.equipment-id").value)
param(2) = "Operation.operation-id, " & CStr(ctrl.item("Operation.operation-id").value)
param(3) = "Operation.operation-order-id, " & _
    CStr(ctrl.item("Operation.operation-order-id").value)
param(4) = "Operation.operation-resource-id, " & _
    CStr(ctrl.item("Operation.operation-resource-id").value)
param(5) = "Operation.operation-process-id, " & _
    CStr(ctrl.item("Operation.operation-process-id").value)

Dim varParam As Variant
varParam = param

Dim res As Variant
res = m_caoCtrl.Execute("GetOrder", varParam)
```

### 3.1.3. CaoController::Execute(“SendResult”) コマンド

MESX に対して実績を通知します。



SendResult (<aryData>)

< aryData > : [in] 通知を行うデータ(VT\_BSTR|VT\_ARRAY)  
データの形式は 4 通知データフォーマットを参照

処理に使用される値は以下の通りです。

Operation. から始まる実績に関する要素および

[Operation.Produce\_Result] , [Operation.Consume\_Result] ,  
[Operation.Consume\_Lot\_Result] , [Operation.Assign\_Result] 以下の各要素。

各グループの要素数は一致している必要があります。



```
Dim argNameList As Variant
argNameList = Array( _
    "Operation.operation-id", "Operation.operation-order-id", "Operation.operation-resource-id", _
    "Operation.operation-process-id", "Operation.produce-item-id", "[Operation.Produce_Result]produce-type", _
    "[Operation.Produce_Result]produce-quantity", "[Operation.Produce_Result]produce-quantity-unit", _
    "[Operation.Consume_Result]consume-item-id", "[Operation.Consume_Result]consume-quantity", _
    "[Operation.Consume_Result]consume-quantity-unit", "[Operation.Consume_Lot_Result]consume-lot-id", _
    "[Operation.Consume_Lot_Result]consume-item-id", "[Operation.Consume_Lot_Result]consume-lot-quantity", _
    "[Operation.Consume_Lot_Result]consume-lot-quantity-unit", "[Operation.Assign_Result]assign-resource-id", _
    "[Operation.Assign_Result]assign-name", "[Operation.Assign_Result]assign-group", _
    "[Operation.Assign_Result]assign-start-time-record", "[Operation.Assign_Result]assign-end-time-record", _
    "Operation.start-time-record", "Operation.end-time-record")

Dim count As Integer
count = UBound(argNameList) - LBound(argNameList) + 1

Dim param() As String
ReDim param(0 To count + 1)
param(0) = CStr(command)
param(1) = ctrlName

Dim i As Integer
For i = LBound(argNameList) To UBound(argNameList)
    param(i - LBound(argNameList) + 2) = argNameList(i) & ", " & GetValueString(ctrl.item(argNameList(i)).value)
Next i
Dim res as Variant
res = m_catCtrl.Execute( "SendResult", CVar(param))
```

### 3.1.4. GaoController::Execute(“ValidLot”) コマンド

MESX にロットの投入可否を問い合わせます。

#### 書式

ValidLot (<aryData> )

< aryData > : [in] 通知を行うデータ (VT\_BSTR|VT\_ARRAY)  
データの形式は 4 通知データフォーマットを参照

処理に使用される値は以下の通りです。

[Operation.Consume\_Lot\_Valid]以下の result-status を除く各要素。  
全ての要素数は一致している必要があります。

戻り値 : [out]コマンドの戻り値 (VT\_BSTR | VT\_ARRAY)  
データの形式は 4 通知データフォーマットを参照

以下の値が設定されます。

[Operation.Consume\_Lot\_Valid]以下の各要素。

#### 使用例

```
Dim argNameList As Variant
argNameList = Array( _
    "lot-id", _
    "lot-lot-id", _
    "consume-item-id", _
    "consume-resource-id", _
    "consume-operation-id", _
    "consume-lot-quantity", _
    "consume-lot-quantity-unit" _
)

Dim count As Integer
count = UBound(argNameList) - LBound(argNameList) + 1

Dim param() As String
ReDim param(0 To count + 1)
param(0) = CStr(VALID_LOT)
param(1) = ctrl.Name

Dim i As Integer
For i = LBound(argNameList) To UBound(argNameList)
    param(i - LBound(argNameList) + 2) = _
        "[Operation.Consume_Lot_Valid]" & argNameList(i) & ", " & _
        GetValueString(ctrl.item("[Operation.Consume_Lot_Valid]" & argNameList(i)).value)
Next i

Dim result() As String
result = m_caoCtrl.Execute("ValidLot", CVar(param))
```

## 4. 通知データフォーマット

通知データは BSTR の配列で表します。

各要素は以下の意味を持ちます。

### 4.1. コマンド番号(1 要素目)

コマンドの番号が格納されます。

エラーが発生した場合、戻り値のコマンド番号に負の値を格納すると MESX へエラーが通知されます。

### 4.2. 設備名称(2 要素目)

Execute を実行する場合は Execute を呼び出した設備の名称を、設備へのメッセージ送信要求の場合はメッセージを送信する設備の名称を指定します。

### 4.3. データ本文(3 要素目以降)

データの本文は以下の形式で記述します。

<CaoSQL 変数名>, <データの内容(配列の場合はコンマ区切りで複数記述)>

処理に必要な要素が存在しない場合は空文字列が指定されたものとして処理を行います。

例)

[Operation. Consume\_Lot\_Valid]lot-id, LOTID1, LOTID2

## 5. メッセージ一覧

MESX プロバイダでは MESX からの要求に対して呼び出し元に対して同期メッセージで問い合わせを行うことがあります。

呼び出し元はメッセージを受け取ったら Reply に要求に応じたパラメータ(パラメータ形式は 4 通知データフォーマットを参照)を生成してプロバイダに返してください。

メッセージの一覧は以下の通りです。

### 5.1.1. GetStatus メッセージ

設備の状態を問い合わせます。

メッセージ ID : [in] 7  
呼び出しパラメータ : [in] 通知を行うデータ (VT\_BSTR|VT\_ARRAY)  
データの形式は 4 通知データフォーマットを参照

渡すデータはありません。  
返信 : [out]コマンドの戻り値 (VT\_BSTR | VT\_ARRAY)  
データの形式は 4 通知データフォーマットを参照

以下の値を返してください。

EquipmentState.equipment-status

### 5.1.2. AddOrder メッセージ

MESX からオーダーの追加を要求します。

CaoController::Execute(“GetOrder”) コマンドと同様の処理でオーダーの設定を行ってください。

メッセージ ID : [in] 1  
呼び出しパラメータ : [in] 通知を行うデータ (VT\_BSTR|VT\_ARRAY)  
データの形式は 4 通知データフォーマットを参照

以下の値が設定されます。

Operation. から始まる計画に関する要素および  
[Operation.Consume\_Plan] , [Operation/Consume\_Lot\_Plan] ,  
[Operation.Assign\_Plan] 以下の各要素。

返信 : [out]コマンドの戻り値 (VT\_BSTR | VT\_ARRAY)  
データの形式は 4 通知データフォーマットを参照

データの本文は必要ありません。

### 5.1.3. SyncStatus メッセージ

MESX から状態通知の同期を要求します。

呼び出し元はこのメッセージを受け取るまでは設備の状態変更を MESX に通知する必要はありません。

このメッセージを受け取った後に設備の状態変更を検知したら CaoController::Execute(“StatusChanged”) コマンドで通知を行ってください。

メッセージ ID : [in] 1  
呼び出しパラメータ : [in] 通知を行うデータ (VT\_BSTR|VT\_ARRAY)  
データの形式は 4 通知データフォーマットを参照

EquipmentState の要素が設定されますが、処理の必要はありません。

返信 : [out]コマンドの戻り値 (VT\_BSTR | VT\_ARRAY)  
データの形式は 4 通知データフォーマットを参照

データの本文は必要ありません。

### 5.1.4. GetResult メッセージ

作業の進捗を問い合わせます。

このメッセージを受け取ったら現在の作業進捗を取得して返信データとして返してください。

メッセージ ID : [in] 5  
呼び出しパラメータ : [in] 通知を行うデータ (VT\_BSTR|VT\_ARRAY)  
データの形式は 4 通知データフォーマットを参照

渡すデータはありません。

返信 : [out]コマンドの戻り値 (VT\_BSTR | VT\_ARRAY)  
データの形式は 4 通知データフォーマットを参照

以下の値を返してください。

Operation. から始まる実績に関する要素および

[Operation.Produce\_Result] , [Operation.Consume\_Result] ,  
[Operation.Consume\_Lot\_Result] , [Operation.Assign\_Result] 以下の各要素。

各グループの要素数は一致している必要があります。

## 6. MESX プロバイダ用 CaoSQLConfig 定義

MESX プロバイダを使用する際、CaoSQLController, 及び, CaoSQLItem に MESX 用の定義を行う必要があります。

CaoSQLConfig 定義は、「別紙: MESX\_CaoSQLConfig.pdf」を参照してください。

MESX 用の CaoSQL 定義が設定された CSQ ファイルは、以下に配置されています。

```
<ORiN2>¥CAO¥ProviderLib¥MESX¥Bin¥MESX.csq
```

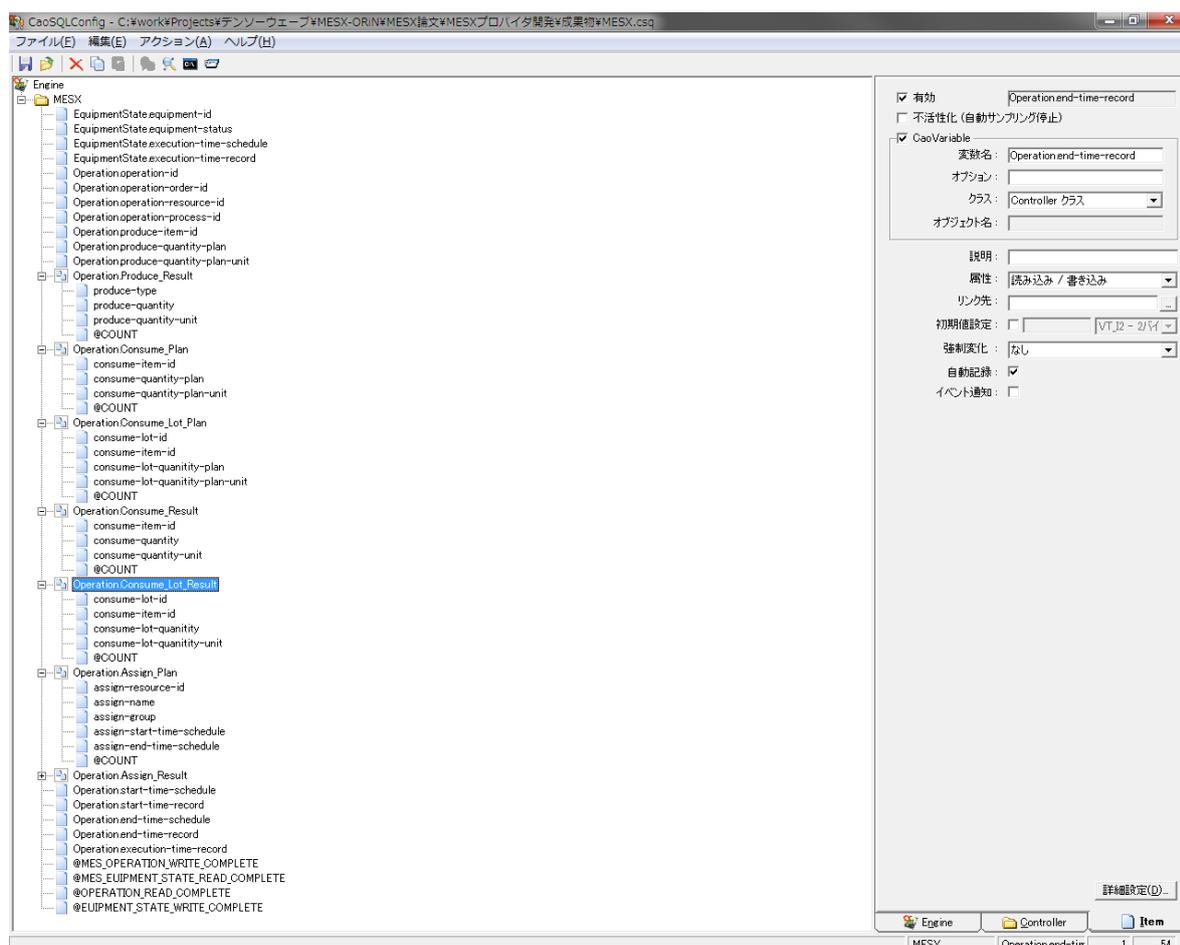


図 6-1 CaoSQLConfig 定義

## MESX プロバイダ使用例

MESX 対応の MES システムである EXPIO MES (ケー・ティー・システム社製) における MESX プロバイダの使用例を記載します。

### 付録A.1. システム構成

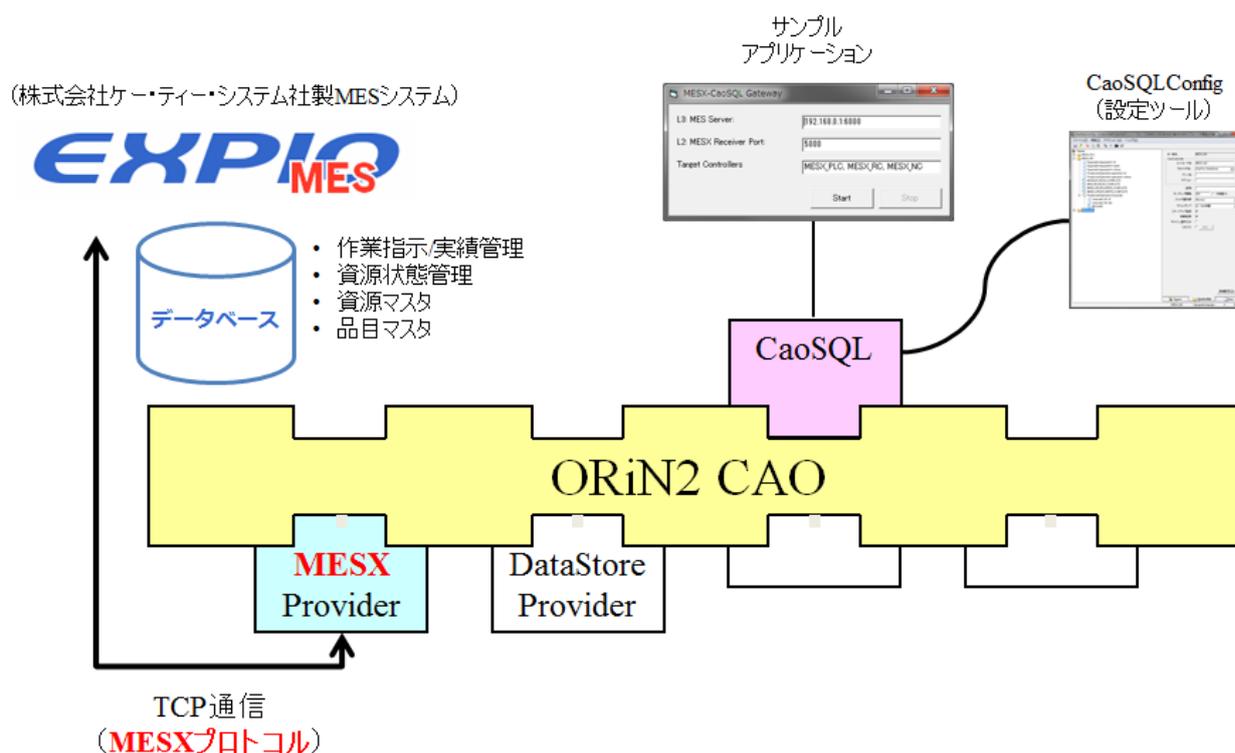


図 6-2EXPIO MES で MESX プロバイダを使用したシステム構成

### 付録A.2. 設定

MESX プロバイダを使用する際、特別な設定は必要ありません。ORiN2 が起動しているマシンのホスト名、TCP 通信用のポート番号を EXPIO MES のコネクタ(通信用モジュール)に登録するだけで、MESX プロバイダを使用した ORiN2 とのデータ交換が可能となります。

## 付録B. MESX プロトコル

MESX プロバイダが対応する各 MESX プロトコルについて以下にまとめます。

詳細については「別紙: MESX\_Message.pdf」を参照してください。

種別	番号	メッセージ名	備考
実行指示	①	MESXaddOperation MESXchangeOperation MESXremoveOperation MESXconfirm	MES から装置に実行指示を行う Add: 追加, Change: 変更, Remove: 取消に対応
	②	MESXisItemValid MESXshowValidity	投入する品目が投入可能かチェックを行う
	③	MESXgetOperation MESXshowOperation	装置が MES から実行指示を取得する
実行実績	④	MESXgetProgress MESXshowProgress	MES が装置から実行実績を取得する
	⑤	MESXaddProgress MESXshowProgress	装置が MES に実行実績を報告する
	⑥	MESXnotifyOperation	装置が MES に実行実績を報告する(非同期)
状態監視	⑦	MESXgetStatus MESXshowStatus	MES が装置の状態を取得する
	⑧	MESXnotifyStatus	装置が MES に状態変化を通知する
	⑨	MESXsyncStatus MESXconfirm	MES が装置からの状態変化通知を要求する