

# IEC 61850対応プロトコルスタック・ソフトウェア

## SISCO MMS Lite

### 製品概要

第1版

2018/10/02

株式会社日新システムズ

<http://www.co-nss.co.jp>

京都本社：600-8482 京都府京都市下京区堀川通四条下ル左側（堀川四条ビル）

東京事務所：101-0024 東京都千代田区神田泉町1番地（神田泉町ビル）

Copyright 2018 Nissin Systems Co.Ltd.

本書についての著作権は、株式会社日新システムズが保有します。

すべての商標、および著作権はそれぞれの所有者が所有しています。

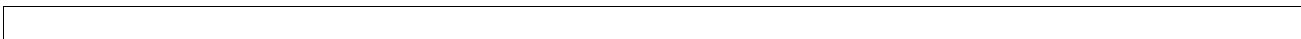
本書に記載された名称等には、必ずしも商標表示（®，TM）を付記していません。

本書の内容は予告なしに変更が行われます。

株式会社日新システムズ（以下NSS）は、この文書または記述されている内容について一切の保障を行いません。NSSはいかなる場合も直接的、付带的、間接的ないかなる損害に対しても、一切責任を負いません。

著作権者により明示された許可なしに、この文書およびこの文書の実質的な修正版を頒布することを禁止します。

著作者から事前の許可を得る事なく、この著作物または派生的著作物を商用目的で標準的な本の形式で出版する事を禁止します。



<b>1.</b>	<b>はじめに</b> .....	<b>5</b>
<b>2.</b>	<b>MMS Lite 製品概要</b> .....	<b>6</b>
2.1	MMS Lite とは.....	6
2.1.1	MMS Lite の特徴.....	7
2.1.2	MMS Lite のリソース.....	8
2.1.3	MMS Lite の対応.....	9
2.1.4	ポーティングについて .....	10
<b>3.</b>	<b>通信プロトコル</b> .....	<b>11</b>
3.1	MMS (Manufacturing Messaging Specification) .....	12
3.1.1	リクエスト-レスポンス型での通信 .....	12
3.1.2	レポート(Reporting).....	13
3.2	GOOSE (Generic Object Oriented Events) .....	15
3.2.1	動作 .....	16
3.3	Sampled Values.....	17
<b>4.</b>	<b>パッケージのフォルダ構成</b> .....	<b>18</b>
4.1	インストール後のフォルダ構成 .....	18
4.2	ビルドを行うフォルダ .....	19
4.3	ユーザ変更可能なファイルが格納されているフォルダ.....	20
<b>5.</b>	<b>アプリケーションのソフトウェア構成</b> .....	<b>21</b>
5.1	Foundry (ファウンダリ).....	22
<b>6.</b>	<b>リーフ関数</b> .....	<b>23</b>
6.1	ライブラリ提供の既存リーフ関数 .....	25
6.2	ユーザリーフ関数 .....	25
<b>7.</b>	<b>サンプルアプリケーション</b> .....	<b>26</b>
7.1	概要 .....	26
7.2	サーバアプリケーション(scl_srvr).....	27
7.2.1	対応機能 .....	27
7.2.2	特徴 .....	27
7.2.3	ファイル内容 .....	28
7.2.4	設定ファイル .....	29
7.2.4.1	SCL ファイルのサンプル .....	29
7.2.4.2	起動時設定ファイル (startup.cfg).....	29
7.2.4.3	プロトコル スタックの設定 (osicfg.xml).....	30
7.2.4.4	ログの設定 (logcfg.xml) .....	30
7.2.4.5	データマップファイル(datamap.cfg と datamapout.cfg).....	31
7.2.5	ビルド方法 .....	32

7.2.5.1	ビルド手順 .....	32
7.2.5.2	ビルドシーケンス.....	33
7.2.5.3	成果物 .....	34
7.2.6	実行方法 .....	35
7.2.7	動作処理概要 .....	36
7.2.7.1	サーバのサンプル scl_srvr_ld.....	36
■	起動時の動作 .....	36
■	起動時のフロー .....	37
■	レポート機能 .....	38
7.3	クライアントアプリケーション(client) .....	41
7.3.1	対応機能 .....	41
7.3.2	特徴 .....	41
7.3.3	ファイル内容 .....	42
7.3.4	ビルド方法 .....	43
7.3.4.1	ビルド手順 .....	43
7.3.4.2	成果物 .....	44
7.3.5	cositcps0 .....	45
7.3.5.1	実行方法 .....	45
7.3.5.2	動作処理概要 .....	46
■	テストモード .....	48

## 1. はじめに

---

MMS Lite をご購入いただき、ありがとうございます。

IEC 61850対応プロトコルスタック・ソフトウェアである SISCO 社の MMS Lite に関する概要を説明します。

また、付属のサンプルアプリケーションのカスタマイズ方法に関しましては、別途プログラマーズ ガイドを参考にして下さい。

## 2. MMS Lite 製品概要

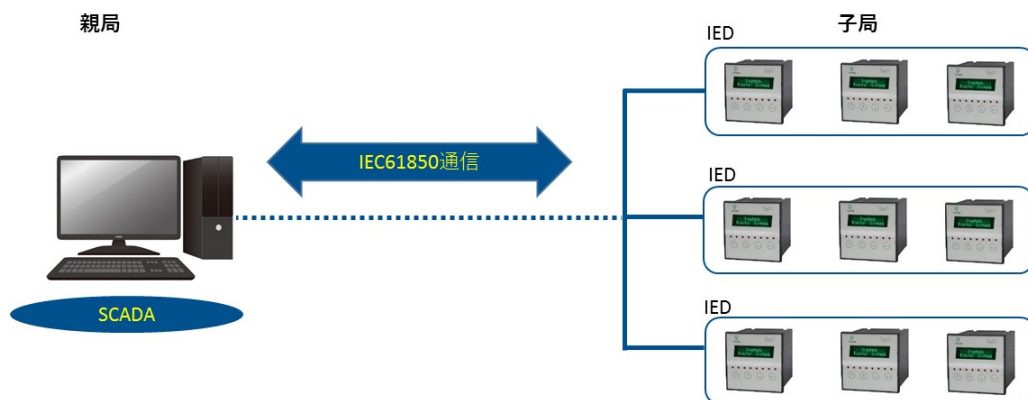
### 2.1 MMS Lite とは

MMS Lite は、変電所構内伝送に関する規格として制定された IEC 61850に沿った変電所オートメーション(既存の変電所の更新・改良)を実現するために組込みシステム用として作成されたプロトコルスタックです。

その為、本製品は遠隔地から変電所の送配電プロセスを監視・制御できるようになり、配電機能の最適化を図る為に、RTU(Remote Terminal Unit)、継電器、再閉路、PLC、変圧器、メータなどリソースに制限のある IEC 61850インテリジェント電子デバイス (IED) の開発に採用されています。

IEC 61850 は、インテリジェントな電子装置(IED : Intelligent Electronic Device)間の情報交換を標準化し、相互運用を達成するために制定された規定です。

IEC 61850 に対応した IED はデジタル化された送電網の状態を表すデータを得ることが可能になります。



### 2.1.1 MMS Lite の特徴

MMS Lite は C 言語で作成された IEC 61850 の通信ミドルウェアです。下表の特徴があります。

- IEC 61850 と MMS プロトコルのサポート
- ANSI C で書かれたソースコード
- ライブラリ、サンプル、ツール、プロトコルスタックの全ての部分をソースコードで提供
- サイズが最少になるように最適化されたコード
- C と C++ のアプリケーションにリンク可能
- サンプルアプリケーションを提供:
  - ✓ IEC 61850 クライアント
  - ✓ IEC 61850 サーバ (MMS)
  - ✓ IEC 61850 サーバ全般 (MMS と GOOSE/SV)
- SCL (変電所構成言語) を使用して IEC 61850 のサーバとクライアントを設定する機能をサポート
- Windows と Linux にはポーティング済みのものを提供  
(但し、これら以外の OS へのポーティングも可能)

## 2.1.2 MMS Lite のリソース

MMS Lite において、使用するメモリ使用量について記載します。

尚、本情報は以下の条件のもと計測したものになります。

- Windows Server 2008 R2のメモリ使用量
- 数値はすべて、ロギングを含むアプリケーションの V6.2000リリース構成に基づく

### ➤ ROM サイズ

アプリケーション	メモリ使用量		ROM サイズ
サンプルサーバ (scl_srvr)	CODE	820,657 bytes	約 1,055 Kbytes
	DATA	258,690 bytes	

※DATA には、プログラムの開始時に使用されるデータのみで、動的に確保されたメモリは含まない。

ROM 及び、プログラムメモリは CODE と DATA の合計と等しいです。

### ➤ RAM サイズ

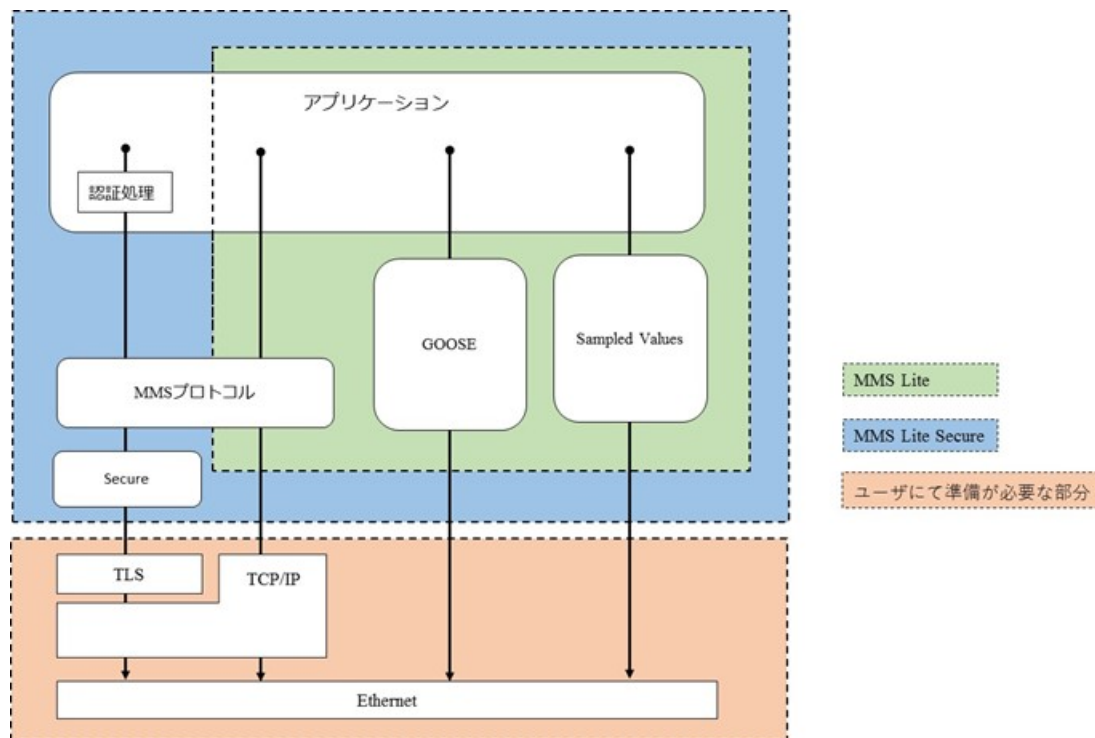
アプリケーション	条件	ROM サイズ
サンプルサーバ(scl_srvr) ※ 計測用に一部変更	"Edition 1"のロジカルノードが5つの場合	約 1,825 Kbytes
	"Edition 1"のロジカルノードが全て(91)の場合	約 7,198 Kbytes

※上記のRAM サイズには、通信バッファまたはバッファされたレポートは含まない。



### 2.1.3 MMS Lite の対応

本製品は MMS Lite だけでなく、セキュリティ拡張オプションの MMS Lite Security Add-On があります。対応範囲としては、通常の MMS Lite に加えて、セキュリティ部分及び、それに関する処理が追加されたものになります。尚、MMS Lite Security Add-On の説明に関しましては本書では省略させていただきます。



尚、MMS Lite に関しては、IEC61850モデルの以下の内容について、全てサポートした製品になります。

IEC 61850 モデル	MMS Lite のサポート状況
論理デバイス (Logical Device)	○
論理ノード (Logical Node)	○
データ (Data)	○
データセット (DataSets)	○
レポート (Reporting)	○
ログ (Logging)	○
制御 (Control)	○
GOOSE	○
GSSE	○
SCL	○

#### 2.1.4 ポーティングについて

MMS Lite は Windows と Linux にポーティングされた状態のコードが提供されています。

これ以外のプラットフォーム (例: 組み込み OS) で使用するには、ポーティングが必要になります。

ポーティング先のハードウェアと OS には、下記のものが必要になります。

- ANSI C のコンパイラと標準ライブラリ
- TCP/IP のプロトコルレイヤー
- BSD ソケット API
- 時計 (データにタイムスタンプを付けるため)
- データモデルを保持するためのメモリ

また、ファイルシステムは SCL ファイルのために必要です。

但し、ファイルシステムが無い OS の場合には、メモリの領域に SCL の XML データを置く事で対応可能です。

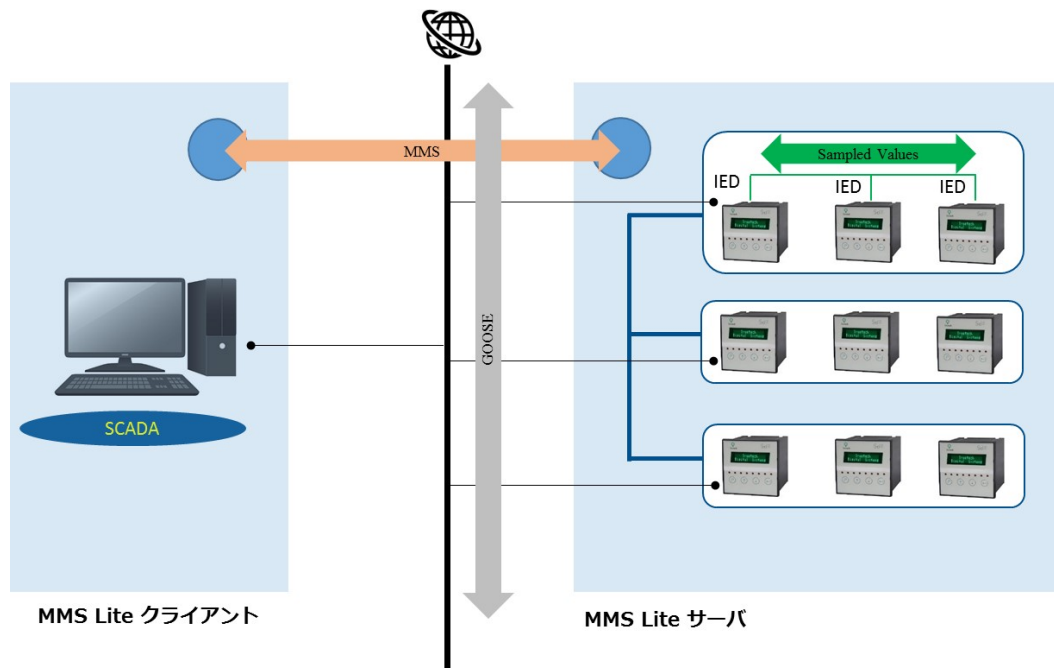
その場合、MMS Lite のファイル読み込み部分をメモリからコピーする様にカスタマイズする必要があります。

ポーティングを行う際には、下記の作業が主に必要になります。

- ターゲット OS 用の Makefile 作成
- イーサネットへのインタフェースの実装 (GOOSE または GSSE 使用時)
- 対象 OS 用のメモリのアライメント ファイル (align.cfg) の作成
- 対象 OS に向けてのビルド

### 3. 通信プロトコル

MMS Lite は、IEC 61850で3種類の通信を利用する下記のプロトコルに対応しています。

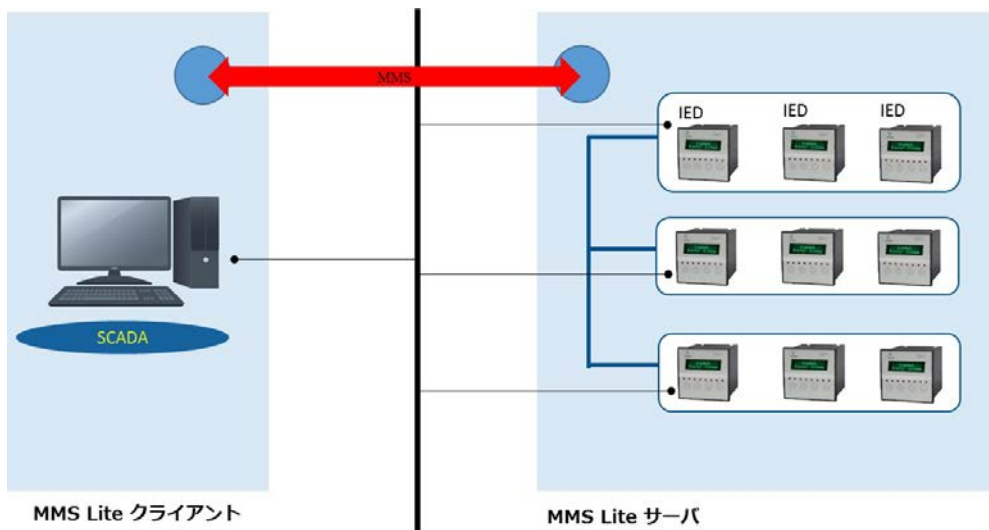


各プロトコルの詳細については、次項にて説明します。

### 3.1 MMS (Manufacturing Messaging Specification)

MMS は以下の特徴があります。

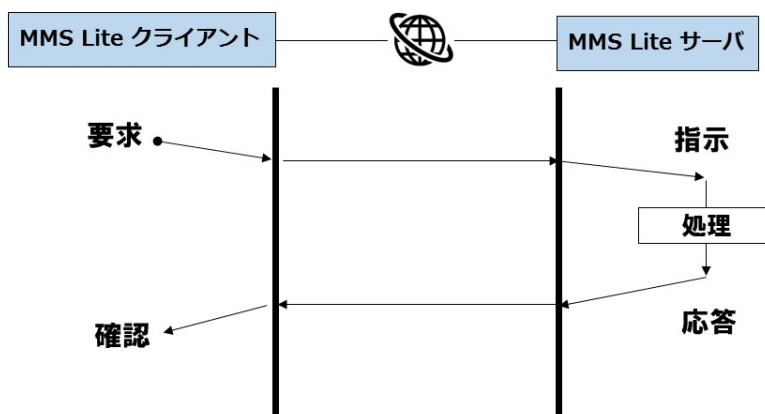
- 監視レベルの SCADA と変電所内の機器に利用
- 各種機器の状態取得や設定値の確認、変更、制御コマンドの発行などを実施する
- TCP/IP 上の通信で行われるクライアント-サーバプロトコル



#### 3.1.1 リクエスト-レスポンス型での通信

クライアント-サーバプロトコルでの MMS では、リクエスト-レスポンス型での通信として以下のシーケンスで動作します。

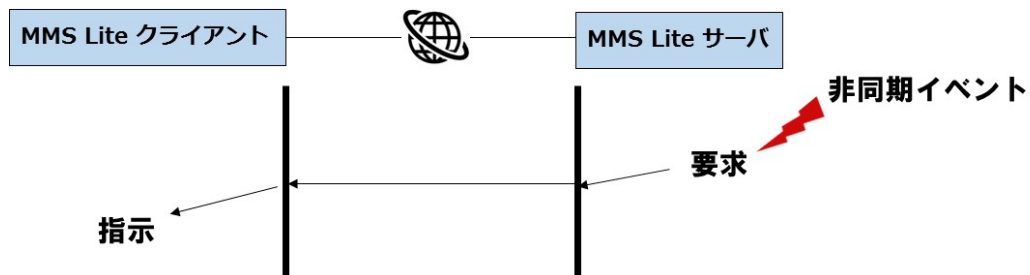
- ◆ リクエスト-レスポンス型での通信



上記は、データ取得、データ設定、制御などの一般的な MMS の動作になります。

### 3.1.2 レポート(Reporting)

レポートとは、サーバが保持している計測値や状態等の変数値を、自動的（一方的）にクライアントに送信する機能になります。



また、レポートを送信するための方法としては以下のものがあり、それぞれに特徴があります。

#### ◆ BUFFERED-REPORT-CONTROL-BLOCK (BRCB)

- ✓ 設定可能なオプション（データ変更、品質変更、データ更新）がトリガーとなり、内部イベントは、レポートの即時送信を発行
- ✓ 送信フロー制御の制約や接続の切断により、データの変更は失われない
- ✓ 送信されないイベントは、後で送信するためにバッファリングされる
- ✓ レポートは、EntryID に従って再送信することができる

#### ◆ UNBUFFERED-REPORT-CONTROL-BLOCK (URCB)

- ✓ 設定可能なオプション（データ変更、品質変更、データ更新）がトリガーとなり、内部イベントは、レポートの即時送信を発行
- ✓ レポートは「ベストエフォート」ベースで送信
- ✓ レポートのバッファリングはされない
- ✓ レポートの再送信はできない

これらのレポートを送信するトリガーとして以下の5種類が存在します。

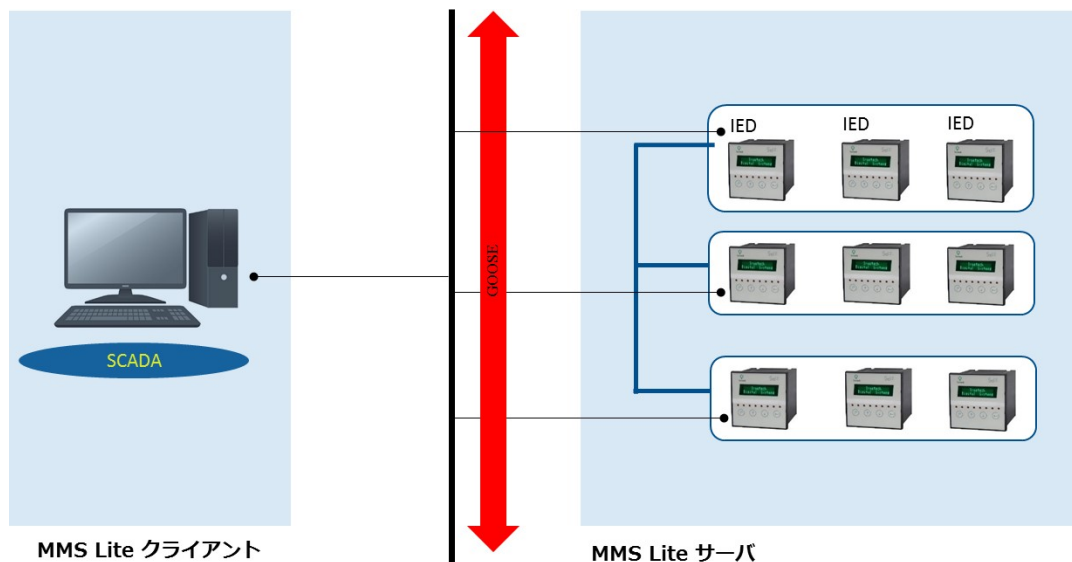
トリガータイプ	内容
data-change (dchg)	データの変更により送信が行われます。
quality-change (qchg)	品質データ属性タイプ (q) の変更により送信されます。
data-update (dupd)	データの更新により送信が行われます。
integrity	タイマー値 (周期的に) が終了したときに、データセットに定義されたすべての値の送信を行う。
general-interrogation	このトリガーは、クライアントによって強制的に送信を促されます。 通常、GI は、クライアントとサーバがセッションを開始または再開するときに要求されます。

### 3.2 GOOSE (Generic Object Oriented Events)

GOOSE は、IEC 61850に準拠して定義された制御モデルであり、電気変電所ネットワーク全体にわたって事象データを高速かつ確実に転送する仕組みを提供します。

このモデルを実装すると、マルチキャストサービスまたはブロードキャストサービスを使用して複数の物理デバイスによって同じイベントメッセージが確実に受信されます。

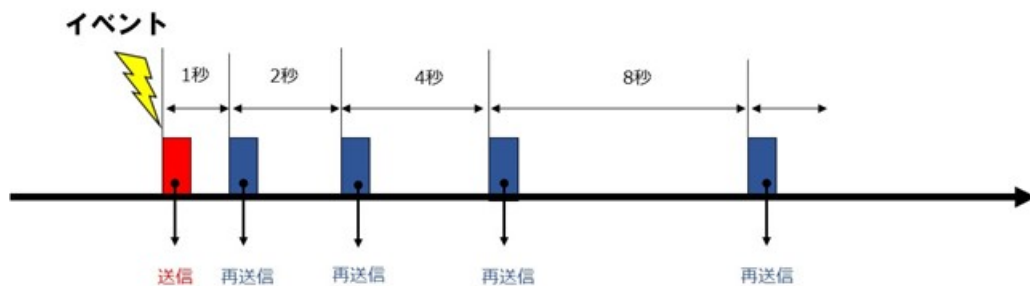
- 変電所内の機器同士の通信に利用
- 高速通信が必要な箇所に使用
- GOOSE データは Ethernet に直接マルチキャスト送信され、publish(送信側)/subscribe(受信側)型の通信



### 3.2.1 動作

GOOSE のメッセージは、規格で繰り返し再送 (retransmit) する事になっています。

再送周期や再送数は特に定められておりませんが、慣習的に時間を倍々にして送信される事が多いです。また、GOOSE はリアルタイム性を確保するため、イーサネットのパイロードに直接データを載せてマルチキャストで送信します。



GOOSE はマルチキャストのイーサネットで送信されるため、MAC アドレスにより自分宛ての GOOSE メッセージかを判断する事ができます。

subscribe(受信側)では MAC チップ(ネットワークカード) で MAC アドレスのフィルタリングを行う事により、ソフトウェアに負荷をかけず効果的になります。

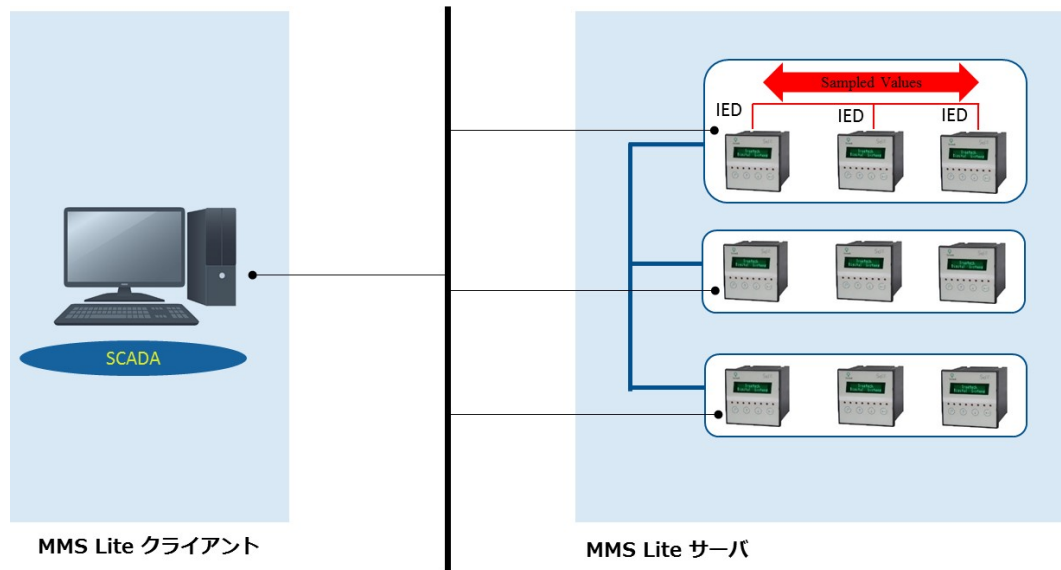
フィルタリングのために受信するアドレスを指定する関数も MMS Lite に用意されています。



### 3.3 Sampled Values

変電所内の機器のアナログデータをサンプリングによりデジタル化して送信するなどの用途に使用されます。

例えば電圧の波形等を一定間隔で計測 (サンプリング) したデータを GOOSE と同様にマルチキャストのイーサパケットで送信します。



サンプルドバリューは、処理の内容がユーザのアプリケーションに大きく依存します。

## 4. パッケージのフォルダ構成

### 4.1 インストール後のフォルダ構成

MMS Lite のフォルダ構成は下記の通りです。

/cmd	コンパイル、ビルドに必要な環境
/gnu	Linux や QNX 等の GNU makefile
/vs2010	Windows (VS2010 以降) 用のプロジェクト
/win32	古い Windows (VS2005) 用のプロジェクト
/src	ライブラリのソースコード
/inc	ヘッダファイル
/mmsop_en	デフォルトの mmsop_en.h ファイルを格納
/uca	プロトコルスタックのコード
/acse	ACSE のソースコード
/goose	GOOSE のソースコード
/slip	SLIP (GOOSE で使用する)
/leant	TP4, TP0, CLNP, ES-IS, subnet, UCA 時間同期
/sn_test	サブネットワークテストツール
/su_target	サブネットワーク (サーバ側)
/sn_test	サブネットワーク (クライアント側)
/bin	ユーティリティの実行ファイル
/mvl	MMS Virtual Lite
/src	MVL ソースコード
/acse	MVL-ACSE ソースコード
/loop	ループバック LLP ファイル
/usr	<b>MVL のサンプルのルート</b>
/client	MVL クライアントのサンプル
/server	(61850 ではない) MVL サーバのサンプル
/iecgoose	IEC GOOSE のサンプル (GSSE を含まない)
/scl_srvr	SCL を使用する IEC 61850 のサンプル
/util	MVL ユーティリティのルート
/foundry	MVL ファウンダリ
/linux	32bit 版 Linux 用アライメントファイル
/linux64	64bit 版 Linux 用アライメントファイル
/win32	Win32 用 makefile
/mbufcalc	MVL バッファ初期化のサポート
/mmslog	MMS PDU のデコードと解析
/gsemtest	グローバルセマフォのテストコード
/doc	製品のマニュアル等 (PFD ドキュメント)
/lib	ライブラリ
/win32	Win32 用ライブラリ

## 4.2 ビルドを行うフォルダ

アプリケーション及び、ライブラリのビルドを行う場合は、各環境にあった以下のファイル内にて移動した後、実施してください。

/cmd	
	/gnu
	/vs2010
	/win32
/src	
/inc	/mmsop_en
/uca	
	/acse
	/goose
	/slip
	/leant
	/sn_test
	/su_target
	/sn_test
/bin	
/mvl	
	/src
	/acse
	/loop
	/usr
	/client
	/server
	/iecgoose
	/scl_srvr
	/util
	/foundry
	/linux
	/linux64
	/win32
	/mbufcalc
	/mmslog
	/gsemtest
/doc	
/lib	
/win32	

尚、各フォルダ内に関しては以下になります。

- gnu : Linux や QNX 等の GNU makefile
- vs2010 : Windows (VS2010 以降) 用のプロジェクト
- win32 : 古い Windows (VS2005) 用のプロジェクト

ビルド時に関しては、ご使用環境にあった上記のフォルダ内にて実施してください。

### 4.3 ユーザ変更可能なファイルが格納されているフォルダ

ユーザが変更可能なファイルが格納されているフォルダ構成は下記の通りです。

/cmd
/gnu
/vs2010
/win32
/src
/inc
/mmsop_en
/uca
/acle
/goose
/slip
/leant
/sn_test
/su_target
/sn_test
/bin
/mvl
/src
/acle
/loop
/usr
/client
/server
/iecgoose
/scl_srvr
/util
/foundry
/linux
/linux64
/win32
/mbufcalc
/mmslog
/gsemtest
/doc
/lib
/win32

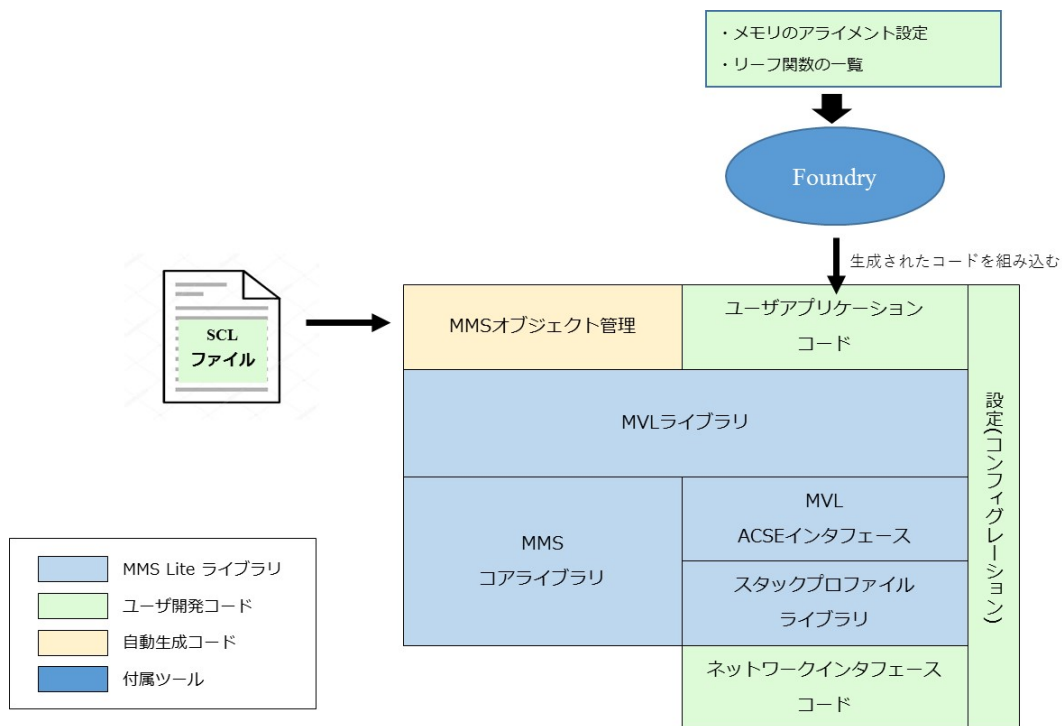
この中でも、ユーザが変更を行うディレクトリは mvl/usr 以下だけになります。これはサンプルアプリケーション及び、コンフィグレーションファイルが含まれております。

IEC 61850 のサーバのアプリケーションを作成する場合は、例えば (NSS のリレーの場合) mvl/usr/NSS-relay のように mvl/usr の下で server ディレクトリをコピーして編集する事が考えられます。

IEC 61850 のクライアントを作成する場合、同様に mvl/usr の下で client ディレクトリをコピーして編集する事が考えられます。しかし、クライアントではアプリケーションにより、サンプルで使用する部分が限られるものです。その場合には新しいディレクトリを mvl/usr の下に作成して、サンプルを参考に新しく作成する方が早い事があります。

## 5. アプリケーションのソフトウェア構成

MMS Lite のアプリケーションの構成は以下の様になっています。



MMS Lite ライブラリ部は、ユーザがコードを変更せずに使用するものです。

MVL (MMS Virtual Lite) とは SISCO 社提供のアプリケーションフレームワークになります。

ユーザ開発コードは、ユーザが作成、またはサンプルを変更して開発するものであり、ユーザのアプリケーション毎に実装が異なります。

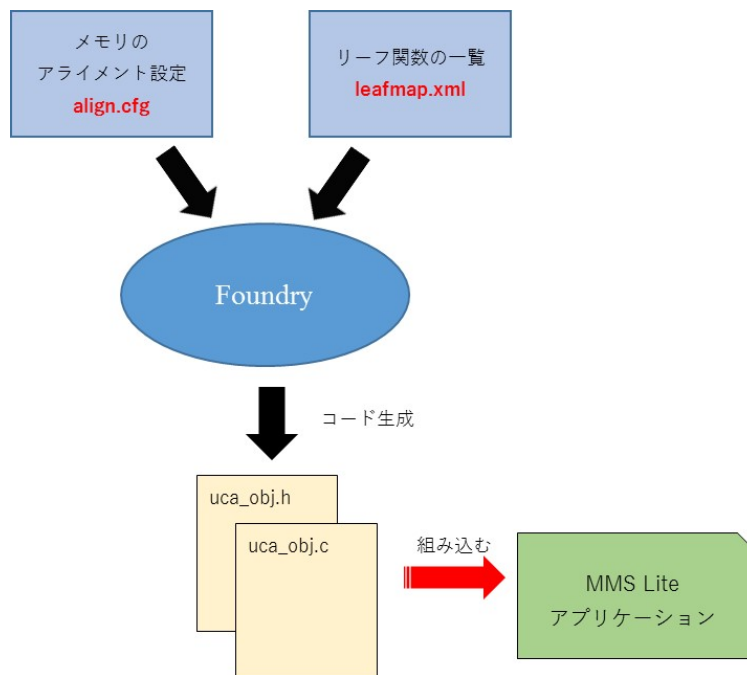
自動生成コード部に関しては、オブジェクトの定義が記載された SCL ファイルを、アプリケーションの実行時に読み込みます。

## 5.1 Foundry (ファウンダリ)

Foundry(ファウンダリ)は、以下の2つの設定ファイルを入力することで、MMS Lite アプリケーションに組み込むためのコードを生成します。

- メモリのアライメント設定 (align.cfg)
- 使用するリーフ関数の名前のリスト (leafmap.xml)

ファウンダリのワークフローの図は次の通りになります。



これは、ビルド時において、makefileにて、自動的に実施する設定になっております。

また、ファウンダリにて、生成された uca\_obj.c と uca\_obj.h はアプリケーションのビルド時においてリンクされ使用されます。

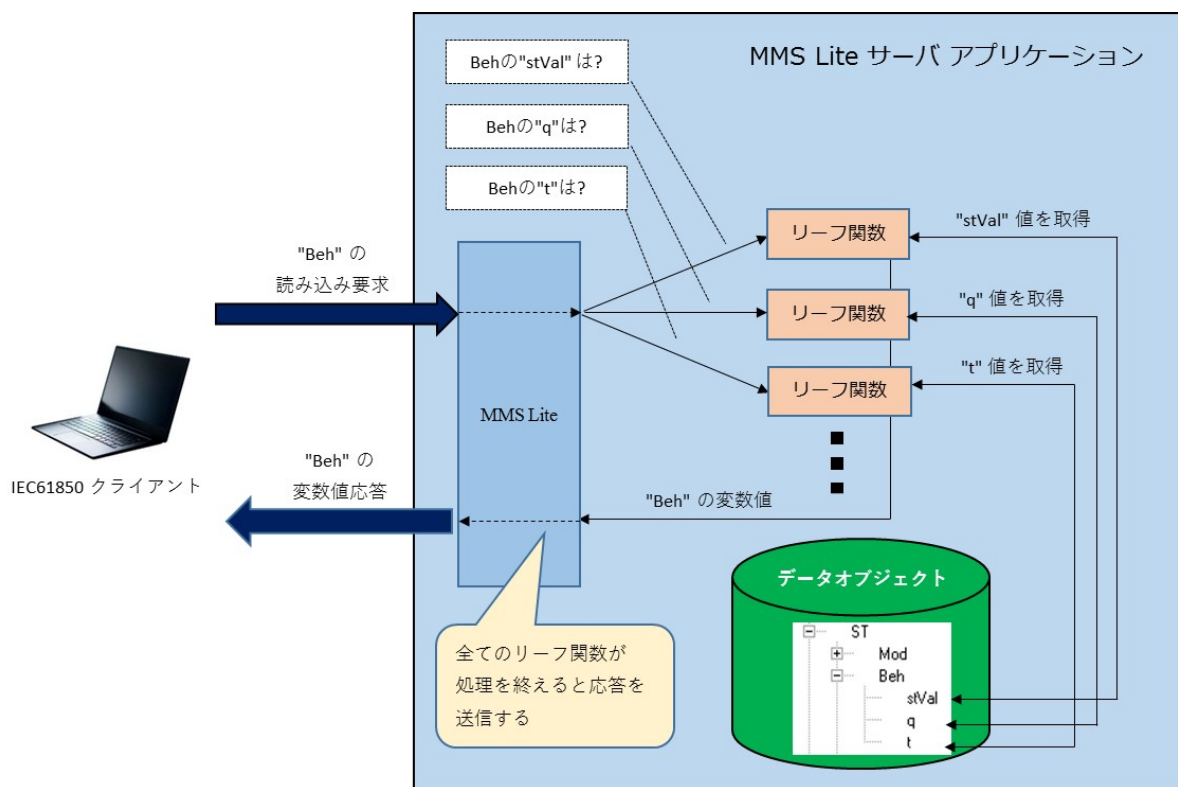
## 6. リーフ関数

「リーフ関数」とは、サーバがクライアントから MMS の要求を受信した(MMS インディケーション) 時、データを処理するために MMS Lite から呼び出されるコールバック関数の名称です。

リーフ関数はその要求で処理を行う必要がある全ての属性 (stVal 等) に対して個々に呼び出されます。

処理を行う属性が MMS データオブジェクトのツリーで末端の「葉」 (リーフ) であるため、「リーフ関数」という名前になっています。

下図は基本的な IEC61850クライアントから MMS Lite のサーバに対して”Beh”の値取得の要求を行った場合のイメージ図になります。



IEC61850クライアントから”Beh”の値取得の要求を受けた場合、サーバ内部では各属性に対する処理をリーフ関数で実施し、データオブジェクトからデータの取得を行います。

その後、全てのリーフ関数の処理が完了した段階で、IEC61850クライアントに変数値の応答を行います。

リーフ関数には、MMS Lite のライブラリで提供されるものと、ユーザがターゲットデバイスの独自処理のために作成する「ユーザリーフ関数」があります。

また、リーフ関数はコールバックで呼び出されるために、MMS Lite は内部でリーフ関数へのポインタの配列を持っています。

アプリケーションが使用するリーフ関数の一覧を leafmap.xml の中で列挙して、ファウンダリにより関数ポインタの配列が uca\_obj.c に自動生成されるようになっています。

なお、アプリケーションでどのリーフ関数を使用するかを設定するため、ライブラリのリーフ関数もファウンダリの leafmap.xml に記載する必要があります。

ファウンダリにて生成された uca\_obj.c と uca\_obj.h の情報を用いて各テーブルにて index 数とともに設定されます。



## 6.1 ライブラリ提供の既存リーフ関数

ライブラリで提供の既存リーフ関数は「MVLU ヘルパー」とも呼ばれます。

レポートの処理、アクセス不可でエラーを返す属性の処理、スタブ関数等の全てのアプリケーションで共通の処理を行うものが用意されています。

それらの処理には、ユーザは SISCO から提供されたリーフ関数を単に使うだけになっています。

ライブラリのリーフ関数には下記の例のものがあります。

例えば、READ のものとしては、u\_custom\_rd\_ind、mvl\_u\_rptena\_rd\_ind\_fun など、WRITE のものとしては u\_custom\_wr\_ind、mvl\_u\_rptena\_wr\_ind\_fun などがあります。これらは、leafmap.xml にて登録されています。

## 6.2 ユーザリーフ関数

「ユーザリーフ関数」は、開発対象の機器で独自の処理をするためにユーザが開発を行うリーフ関数です。

例えば、応答データをクライアントに返信するため、実データをデータオブジェクトに設定する等です。

SCL で動的にオブジェクトを定義するため、leafmap.xml にはリーフ関数名だけを設定すれば良いものになっています。そのため、全エントリのオブジェクト名の Name には、動的な設定を示す \$Dynamic を指定します。

```
<Leaf Name="$dynamic"  
  RdIndFun="u_int32_offset_rd_ind_fun"  WrIndFun="u_int32_offset_wr_ind_fun"  
  Ref=""/>
```

尚、サンプルでは、リーフ関数から MMS オブジェクトへの書き込みを行うのはリーフ関数が呼び出されるタイミング固定になっております。ユーザ独自のタイミングで MMS オブジェクトへ書き込みを行う場合は、別途用意しておりますプログラマーズガイドを参考にしてください。

## 7. サンプルアプリケーション

---

### 7.1 概要

サーバでのサンプルアプリケーションは、/mvl/usr/server 及び、/mvl/usr/scl\_srvr にそれぞれ格納しています。この違いは以下になります。

/mvl/usr/server	一般的な MMS プロトコルを用いたサーバのサンプルアプリケーション
/mvl/usr/scl_srvr	SCL ファイルを用いる IEC61850に対応したサーバのサンプルアプリケーション

上記のことから、基本的使用頂く場合には、scl\_srvr の使用を推奨しております。

クライアントのサンプルアプリケーションは、/mvl/usr/client に格納しています。

/mvl/usr/client	一般的な MMS プロトコルを用いたクライアントのサンプルアプリケーション
-----------------	---------------------------------------

また、その他のサンプルアプリケーションとしては以下のものを用意しております。

/mvl/usr/iecgoose	IEC61850の GOOSE フレームワークのサンプルアプリケーション
/mvl/usr/gse_mgmt	IEC 61850 GSE Management のサンプルアプリケーション

## 7.2 サーバアプリケーション(scl\_srvr)

SCL ファイルを用いる IEC61850に対応したサーバのサンプルアプリケーションに関するファイルを収納

### 7.2.1 対応機能

scl\_srvr 内にあるサンプルアプリケーションの主な対応機能は以下の様になっております。

内容	対応
MMS 通信	○
GOOSE	○
Sample Values	○
Reporting	○

尚、本フォルダ内には複数のアプリケーションが生成されるため、各アプリケーションの詳細機能に関しましては、こちら[成果物](#)を参考にして下さい。

### 7.2.2 特徴

ファイルシステムを用いて、IEC 61850-6で定義されている「サブステーション構成記述言語」に準拠した SCL ファイルを読み込み、全ての MMS オブジェクトを動的に作成します。

### 7.2.3 ファイル内容

以下のフォルダ内のファイルについて説明します。尚、ヘッダファイル(\*.h)に関しましては下記の説明には記載していません。

¥mmslite	root MMS-Lite directory
¥mvl	MMS Virtual Light
¥usr	MVL sample user root
¥scl_srvr	MVL sample server

- Cソースファイル

ファイル名	拡張子	内容
scl_srvr.c	C	main
scl_test.c	C	main
startup.c	C	SCL ファイル読み込み関連のコード
userleaf.c	C	Leaf 関数の書き込み関連のコード
userleaf2.c	C	Leaf 関数の読み込み関連のコード
usermap.c	C	datamap.cfg ファイルを読み込み関連のコード
subnet_scl.c	C	IEEE 802.3 MAC フレームの受信処理関連のコード
userwrite.c	C	IEC 61850オブジェクトの MMS Write 処理関連のコード
db_61850.c	C	IEC 61850における不感帯の設定および計算関連のコード
userleaf_beh.c	C	IEC 61850 "Beh.stVal"の読み取り関連のコード
userleaf_health.c	C	LLN0の IEC 61850 "Health.stVal"、"Health.q"、"Health.t"の読み取り関連のコード

- 設定関連ファイル

ファイル名	拡張子	内容
sisco_sample.cid	CID	SCL サンプルファイル
sisco_sample_420.cid	CID	SCL サンプルファイル
sisco_sample_ed2.scd	SCD	SCL サンプルファイル
sisco_sample_wind.cid	CID	SCL サンプルファイル
logcfg.xml	XML	ロギング機能の設定ファイル
osicfg.xml	XML	TCP/IP および、Network Layer(CLNP/ES-IS)の設定ファイル
leafmap.xml	XML	任意の構造の属性を処理するリーフ関数を設定するファイル
datamap.cfg	CFG	「リーフ名」を「ユーザ定義のテキスト」にマッピングするための情報
startup.cfg	CFG	IEDName および AccessPointName パラメータ等の設定ファイル
uca_obj.odf	ODF	foundry ユーティリティアプリケーションへの入力ファイル
sisco_sample.cid	CID	SCL サンプルファイル

## 7.2.4 設定ファイル

### 7.2.4.1 SCL ファイルのサンプル

下記の2つの SCL ファイルのサンプルを用意しています

- sisco\_sample.cid
- sisco\_sample\_ed2.scd
- sisco\_sample\_420.cid
- sisco\_sample\_wind.cid

但し、実施に使用するのは、デフォルトで設定されている sisco\_sample.cid と IEC61850 Edition 2 に対応した sisco\_sample\_ed2.scd になります。

### 7.2.4.2 起動時設定ファイル (startup.cfg)

サーバのサンプルアプリケーションでは、startup.cfg ファイルにアプリケーションの起動時に与えるパラメータを設定しています。特に重要な設定は下記のものになります。

- ① 使用する SCL ファイルのファイル名
- ② サーバが使用する IED 名
- ③ サーバが使用する IED のアクセスポイント名 (AP 名)

デフォルトの内容は以下の様になります。

```
SCLFileName sisco_sample.cid ①
#NOTE: Use this SCL file instead for IEC 61850 Edition 2.
#SCLFileName      sisco_sample_ed2.scd
IEDName      E1Q1SB1 ②
AccessPointName      S1 ③
ReportScanRate      2.0

#NOTE: BRCBBufferSize was not configurable before. Now it is.
BRCBBufferSize      10000

#NOTE: The old function "sc12_ld_create_all" ignores LogScanRateSeconds and LogMaxEntries.
# You must use the new function "sc12_ld_create_all_scd" to use these.
LogScanRateSeconds      2.0
LogMaxEntries      1000
```

#### 7.2.4.3 プロトコルスタックの設定 (osicfg.xml)

MMS Lite のサンプルアプリケーションでは、プロトコルスタック (MMS や TCP) のバッファサイズや最大接続数等の設定を osicfg.xml ファイルで行っています。

なお、osicfg.xml のファイル名にある "OSI" は「OSI 7階層」の OSI の事です。

#### 7.2.4.4 ログの設定 (logcfg.xml)

MMS Lite のサンプルアプリケーションでは、アプリケーションのログを設定するために logcfg.xml ファイルを使用します。

この設定ファイルでは、ログの有効・無効、ファイルサイズ、出力するログの選択 (ログマスク) の設定を行います。

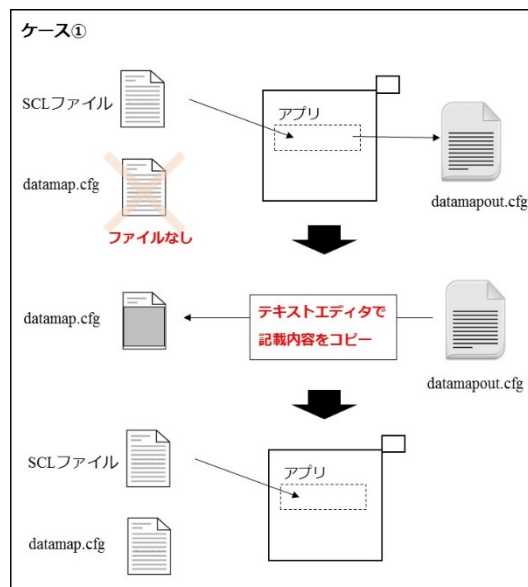
### 7.2.4.5 データマップファイル(datamap.cfg と datamapout.cfg)

各ファイルは以下の内容になります。

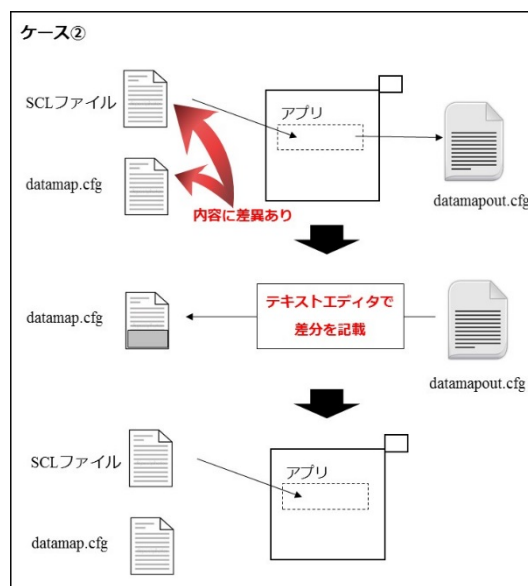
datamap.cfg	アプリケーション実行時に読み込まれるファイル
datamapout.cfg	SCL ファイルを元に生成されるファイル

datamap.cfg は、SCL ファイルに記載された機器構成に合わせて変更する必要があります。

これは、アプリケーションを実行するたびに SCL ファイルを元に"datamapout.cfg"が生成され、datamap.cfg の有効なマッピング情報と他のすべての Leaf 関数のエントリが含まれます。その為、datamapout.cfg の情報を"datamap.cfg"にコピーする事で使用できます。(ケース①)



また、アプリケーションを実行した時に SCL ファイルと datamap.cfg に違いがある場合も datamapout.cfg に出力されます。その場合、datamap.cfg と datamapout.cfg の差分を datamap.cfg に追記することで使用することが出来ます。(ケース②)



## 7.2.5 ビルド方法

### 7.2.5.1 ビルド手順

サンプルアプリの scl\_srvr をビルドするには、以下の手順にて実行します。

- ① <install directory>/cmd/gnu に移動

```
> cd <install directry>/cmd/gnu
```

- ② 付属のスクリプトを下記のコマンドにて実行

- ビルドする場合

```
> ./mmslite802.sh LINUX
```

- クリーンする場合は

```
> ./mmslite802.sh LINUX clean
```

※ Linux 環境の場合、引数に「LINUX」を記載する必要があります。

- ③ 以上でサンプルアプリケーションのビルドが完了です。

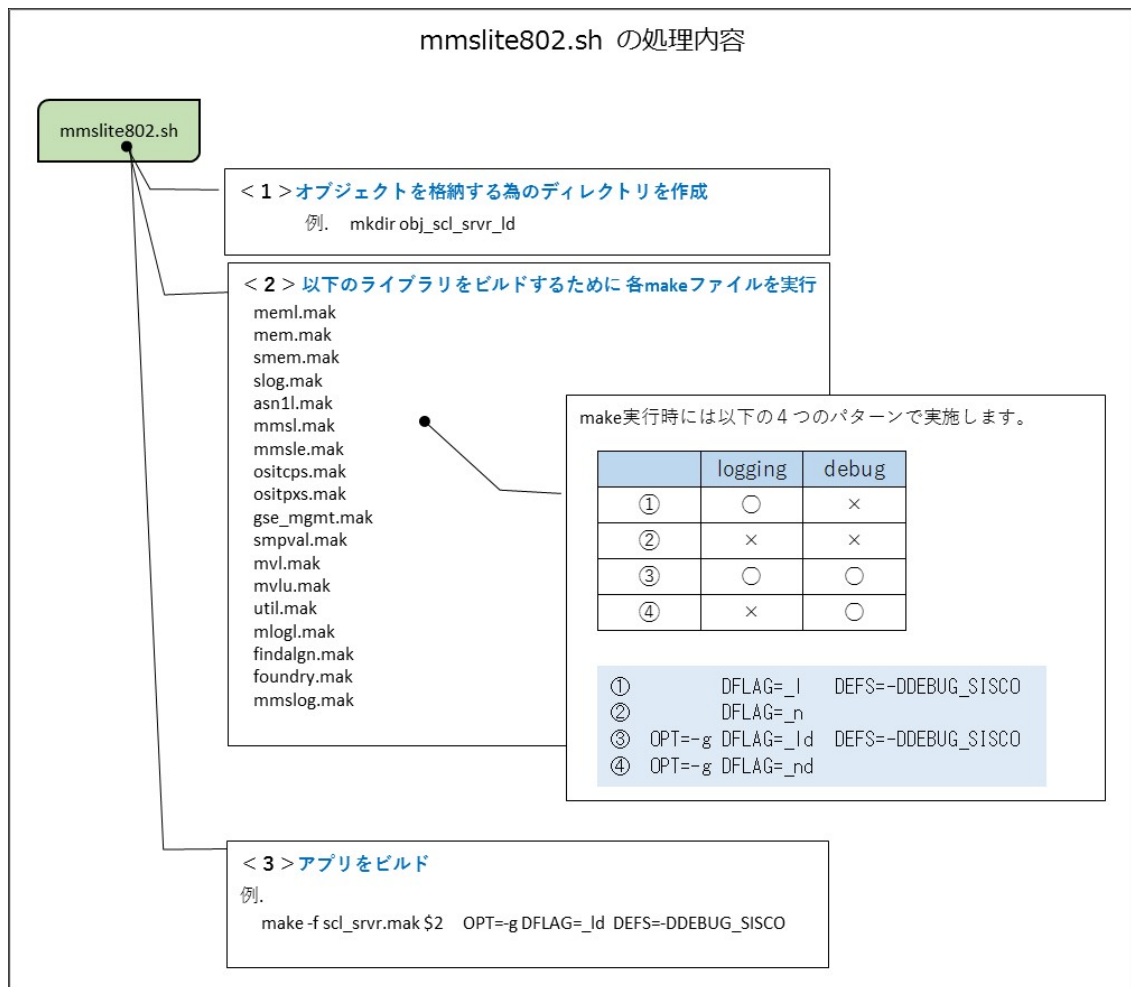
※ビルドに失敗した場合は、同じフォルダ内の cc.lst を確認して下さい。

このファイルにはビルド時のログが記載されているのでエラーの原因の手がかりになります。



### 7.2.5.2 ビルドシーケンス

ビルドを行う際に実行する付属のスク립ト mmslite802.sh は、以下の手順の処理を実施します



また、scl\_srvr.mak では、関連するソースコードのビルドだけでなく、foundry ユーティリティを用いて uca\_obj.c および、uca\_obj.h を生成します。

### 7.2.5.3 成果物

ビルドの実施において、以下の実行ファイルが生成されます。それぞれの実行ファイルには各設定が行われております。

- **scl\_tpxs0\_ld**

Sampled Values および、GOOSE の送受信を含んだ全ての機能が動作します。

定義	内容
MMS_LITE	MMS ソースコードをコンパイル
MOSI	最小 OSI プロファイルを選択
LEAN_T	トランスポート層コードを有効
MVL_UCA	UCA の MVL および IEC 61850 のサポート
MLOG_ENABLE	MLOG サブシステム用
USE_RECURSIVE_READ	再帰呼び出しを使用
USE_DIB_LIST	デバイスに依存しないビットマップを使用
SMPVAL_SUPPORT	SMPVAL 対応
GOOSE_RX_SUPP	GOOSE メッセージ受信
GOOSE_TX_SUPP	GOOSE メッセージ送信

- **scl\_srvr\_ld**

UCA の MVL および IEC 61850 にサポートした MMS の通信機能が動作します。

定義	内容
MMS_LITE	MMS ソースコードをコンパイル
MOSI	最小 OSI プロファイルを選択
LEAN_T	トランスポート層コードを有効
MVL_UCA	UCA の MVL および IEC 61850 のサポート
MLOG_ENABLE	MLOG サブシステム用
USE_RECURSIVE_READ	再帰呼び出しを使用
USE_DIB_LIST	デバイスに依存しないビットマップを使用

- **scl\_test\_ld**

MMS の通信のみの最小限の機能が動作します。(UCA の MVL および IEC 61850 は未サポート)

定義	内容
MMS_LITE	MMS ソースコードをコンパイル
MOSI	最小 OSI プロファイルを選択
LEAN_T	トランスポート層コードを有効

## 7.2.6 実行方法

実行方法に関しては、以下のコマンドで実行します。

尚、Linux 環境の場合は root 権限にて実行してください。

```
sudo ./scl_tpxs0_ld -m scl_parse_mode
```

```
usage: %s -m scl_parse_mode
```

「scl\_parse\_mode」は SCL 解析モードです。

「scd」または、「cid」(デフォルト)のどちらかの必要があります。

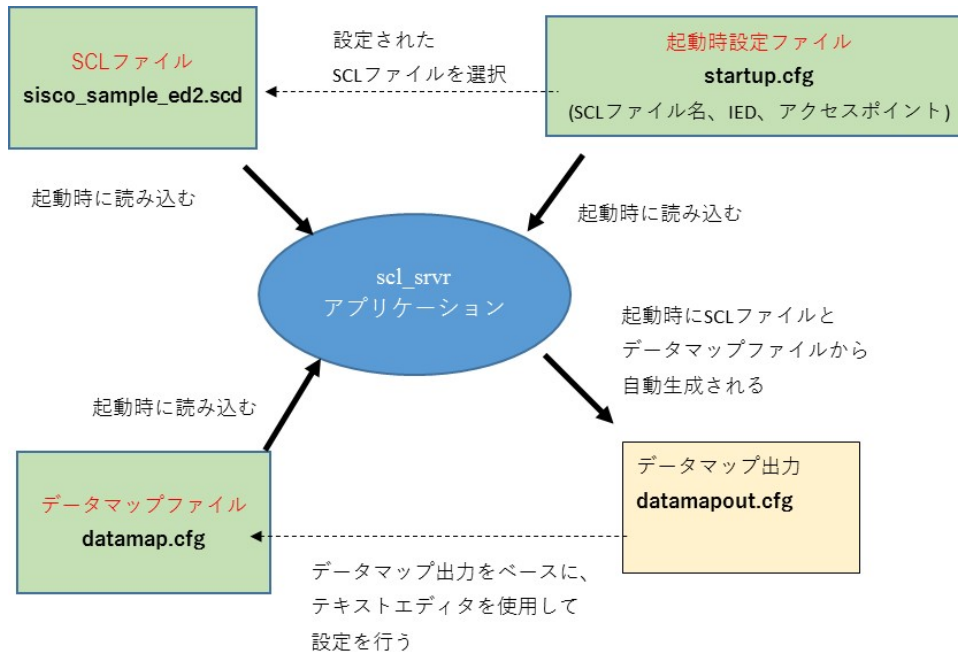
各モードの主な違いについては、以下の様になります。

- SCL\_PARSE\_MODE\_SCD 指定時 (-m scd)
  1. 指定された SCL ファイルにて定義された全ての論理ノードタイプから(MMS データ型)を生成
  2. 設定された(MMS データ型)の情報から VMD を生成
  3. データを datamapout.cfg にマッピング
  
- SCL\_PARSE\_MODE\_CID 指定時(-m cid) (-m の設定が無い場合のデフォルト値)
  1. 指定された SCL ファイルを読み込み、全てのオブジェクトを生成
  2. 1つの IED 内の1つの AccessPoint の情報を(SCL\_INFO 構造体)に設定
  3. SCL で定義された全ての論理ノードタイプ (LNodeType) から MMS データ型を生成
  4. (SCL\_INFO 構造体)から、論理デバイス(MMS ドメイン)、論理デバイス内のすべての論理ノード (MMS 変数)、すべてのデータセット(MMS NamedVariableLists)、およびすべての ReportControlBlocks を生成
  5. データを datamapout.cfg にマッピング

## 7.2.7 動作処理概要

### 7.2.7.1 サーバのサンプル scl\_srvr\_ld

#### ■ 起動時の動作



`startup.cfg` には、`scl_srvr` の起動時に必要な設定を行います。

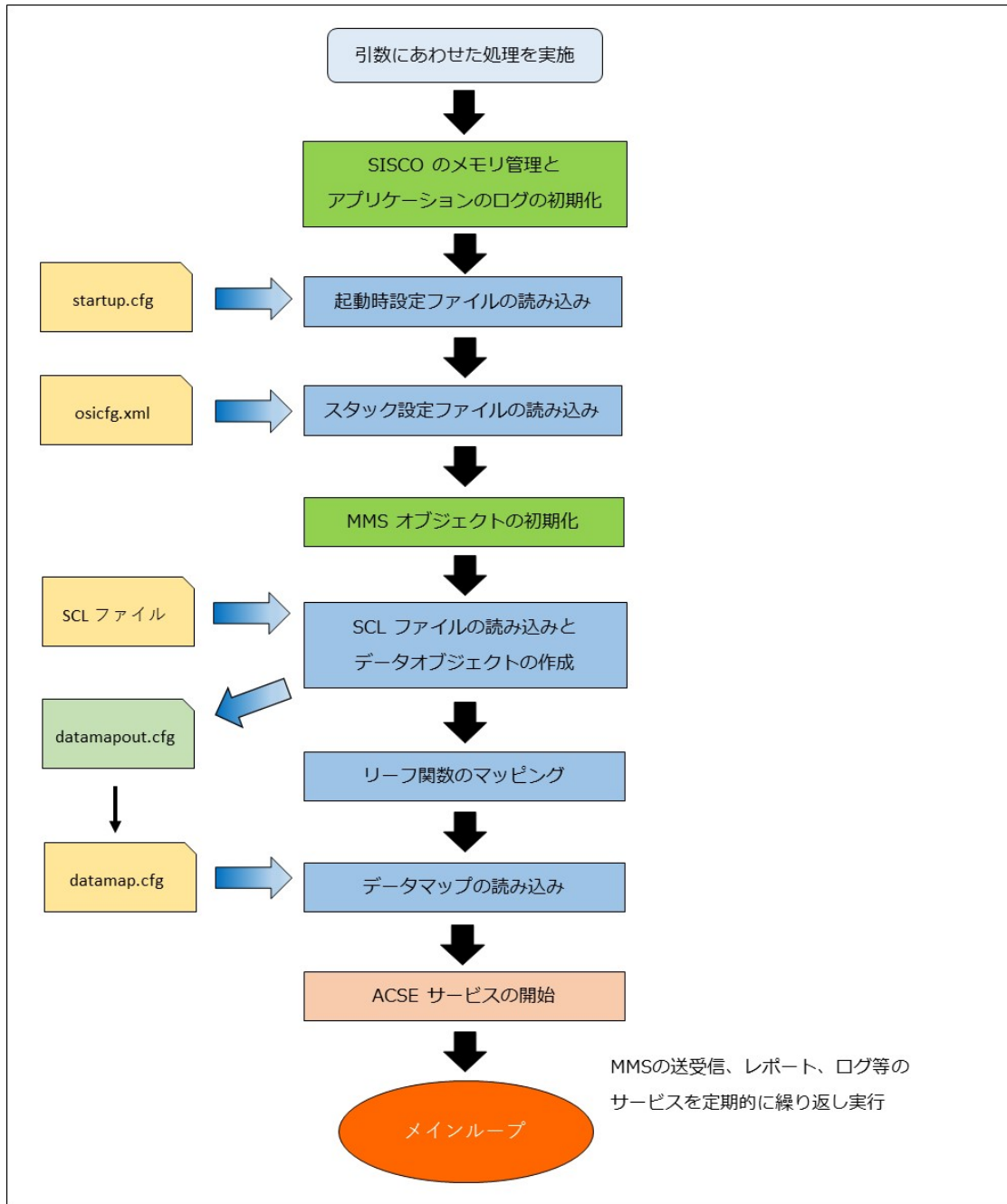
そこに記載されている SCL ファイル名から IEC 61850 の設定を読み込みます。

`datamap.cfg` は、MMS のオブジェクト名 (属性) と、実データ (センサ値等) とのマッピングを行う為のファイルになっています。この設定の雛形が、`datamapout.cfg` として出力されます。

SCL で定義したものに対するマッピング情報のため、SCL を変更した場合は `datamap.cfg` も変更の必要があります。

■ 起動時のフロー

scl\_srvr の起動時に、必要な引数の設定を行った後、scl\_srvr は以下のような処理を行い、最終的にレポート、ログ等の処理及び、クライアントとの MMS の送受信を行うメインループの処理を行います。



■ レポート機能

scl\_srvr では、以下の方法にて各レポート送信を行う為に設定が必要です。

◆ data-change(dchg)を有効にする方法

有効にするには以下の設定が必要です。

1. RCB の TrgOps で dchg ビットを設定するために SCL ファイルで設定

特定の属性（IEC 61850-7-3で定義）の値が変更された場合にのみ送信されるため  
これらの属性を次のように SCL で正しく設定してください。

例)  
<DA name="stVal" fc="ST" bType="BOOLEAN" dchg="true"/>

2. クライアントから、RptEna(report enable)に "1"を書き込んで RCB(Report Control Block)を有効にします。

◆ quality-change(qchg)を有効にする方法

有効にするには以下の設定が必要です。

1. RCB の TrgOps で qchg ビットを設定するために SCL ファイルで設定

"q"属性（IEC 61850-7-3で定義）の値が変更された場合にのみ、qchg レポートが  
送信されるため"q"属性も SCL で次のように正しく設定してください。

例)  
<DA name="q" fc="ST" bType="Quality" qchg="true"/>

2. クライアントから、RptEna(report enable)に "1"を書き込んで RCB(Report Control Block)を有効にします。

## ◆ data-update(dupd)を有効にする方法

dupd をトリガーとしたレポートはデータがいつ更新されたかを知る方法がないため、アプリケーションからのトリガーを行う必要があります。

また、次のように SCL で正しく設定されている必要があります。

```
例)
<DA name="stVal" fc="ST" bType="BOOLEAN" dupd="true"/>
```

MMS-Lite ライブラリではユーザーのデータがいつ update されたかを知る方法が無い為、処理を行う userleaf.c の set\_rpt\_reason 関数に次のようなコメントを記述しております。

```
/* NOTE: User must add customized code to support TrgOps = "dupd".
*/
```

その為、dupd をトリガーとしたレポートに対応するにはカスタマイズする必要があります。

## ◆ 周期的にレポートを送信する方法 (integrity)

有効にするには以下の設定が必要です。

1. RCB の TrgOps で period ビットを設定するために SCL ファイルで設定

SCL ファイルの ReportControl 要素内の TrgOps 要素に period=true に設定し、intgPd に周期時間を設定します。

```
<ReportControl name="PosReport" desc="description" rptID="E1Q1Switches"
  datSet="Positions" confRev="0" buffered="true" bufTime="60" intgPd="1000">
  <TrgOps dchg="true" qchg="true" dupd="true" period="true"/>
```

2. クライアントから、RptEna(report enable)に "1"を書き込んで RCB(Report Control Block)を有効にします。

◆ **GIによるレポートを送信する方法 (general interrogation)**

有効にするには以下の設定が必要です。

1. RCB の TrgOps で period ビットを設定するために SCL ファイルで設定

SCL ファイルの ReportControl 要素内の TrgOps 要素に gi="true" を設定します。

ただし、設定されていない場合は default 値として"true"になります。

2. クライアントから RCB の "GI"属性に "1"を書き込む

"GI"レポートを送信させるときはいつでも実施する必要があります。



## 7.3 クライアントアプリケーション(client)

### 7.3.1 対応機能

client 内にあるサンプルアプリケーションの主な対応機能は以下の様になっております。

内容	対応
MMS 通信	○
GOOSE	○
Sample Values	○
Reporting (送信要求)	○

尚、本フォルダ内には複数のアプリケーションが生成されるため、各アプリケーションの詳細機能に関しましては、こちら[成果物](#)を参考にして下さい。

### 7.3.2 特徴

本サンプルアプリケーションは、コマンドによりテストモードを選択することができます。テストモードの詳細は[実行方法](#)に記載しております。

また、起動時において、以下の2通りのパターンを選択することができます。

- サーバの scl ファイルを読み込み、その設定に合った動作を行う
- 起動時においてサーバへの問い合わせを行うことで情報を収集し、それに合った動作を行う

### 7.3.3 ファイル内容

以下のフォルダ内のファイルについて説明します。尚、ヘッダファイル(\*.h)に関しましては下記の説明には記載していません。

¥mmslite	root MMS-Lite directory
¥mvl	MMS Virtual Light
¥usr	MVL sample user root
¥client	MVL sample client

- Cソースファイル

ファイル名	拡張子	内容
client.c	C	main
cli_async.c	C	非同期リクエスト関数のサンプルコード
cli_goose.c	C	IEC 61850 GOOSE メッセージをサブスクライブするサンプルコード
cli_nvl.c	C	Named Variable List (NVL) を読み取るクライアント関数のソースコード
cli_rpt.c	C	クライアントのレポート処理に関するソースコード

- 設定関連ファイル

ファイル名	拡張子	内容
clntobj.odf	ODF	MMS オブジェクトコンフィグレーションファイル
logcfg.xml	XML	ロギングコンフィグレーションファイル
osicfg.xml	XML	Lean-T スタックコンフィグレーションファイル

## 7.3.4 ビルド方法

### 7.3.4.1 ビルド手順

サンプルアプリの client をビルドするには、以下の手順にて実行します。手順は scl\_srvr の手順と同じになります。

- ① <install directry>/cmd/gnu に移動

```
> cd <install directry>/cmd/gnu
```

- ② 付属のスクリプトを下記のコマンドにて実行

- ビルドする場合

```
> ./mmslite802.sh LINUX
```

- クリーンする場合は

```
> ./mmslite802.sh LINUX clean
```

※ Linux 環境の場合、引数に「LINUX」を記載する必要があります。

- ③ 以上でサンプルアプリケーションのビルドが完了です。

※ビルドに失敗した場合は、同じフォルダ内の cc.lst を確認して下さい。

このファイルにはビルド時のログが記載されているのでエラーの原因の手がかりになります。

## 7.3.4.2 成果物

ビルドの実施において、以下の実行ファイルが生成されます。それぞれの実行ファイルには各設定が行われております。

● **cositcps0\_ld**

ositcps スタックライブラリを使用した TCP/IP のサンプルアプリケーションです。

また、ositcps スタックライブラリは以下の機能があります。

定義	内容
MMS_LITE	MMS ソースコードをコンパイル
MOSI	最小 OSI プロファイルを選択
LEAN_T	トランスポート層コードを有効
MVL_UCA	UCA の MVL および IEC 61850 のサポート
MLOG_ENABLE	MLOG サブシステム用
USE_RECURSIVE_READ	再帰呼び出しを使用
USE_DIB_LIST	デバイスに依存しないビットマップを使用
SMPVAL_SUPPORT	SMPVAL 対応
GOOSE_RX_SUPP	GOOSE メッセージ受信
GOOSE_TX_SUPP	GOOSE メッセージ送信

● **cositpxs0\_ld**

ositpxs スタックライブラリを使用した TCP/IP over Ethernet のサンプルアプリケーションです。

定義	内容
MMS_LITE	MMS ソースコードをコンパイル
MOSI	最小 OSI プロファイルを選択
LEAN_T	トランスポート層コードを有効
MVL_UCA	UCA の MVL および IEC 61850 のサポート
MLOG_ENABLE	MLOG サブシステム用
USE_RECURSIVE_READ	再帰呼び出しを使用
USE_DIB_LIST	デバイスに依存しないビットマップを使用

### 7.3.5 cositcps0

#### 7.3.5.1 実行方法

実行方法に関しては、以下のコマンドで実行します。

尚、Linux 環境の場合は root 権限にて実行してください。

```
sudo ./ cositcps0_ld
```

```
usage: %s [-m test_mode]
```

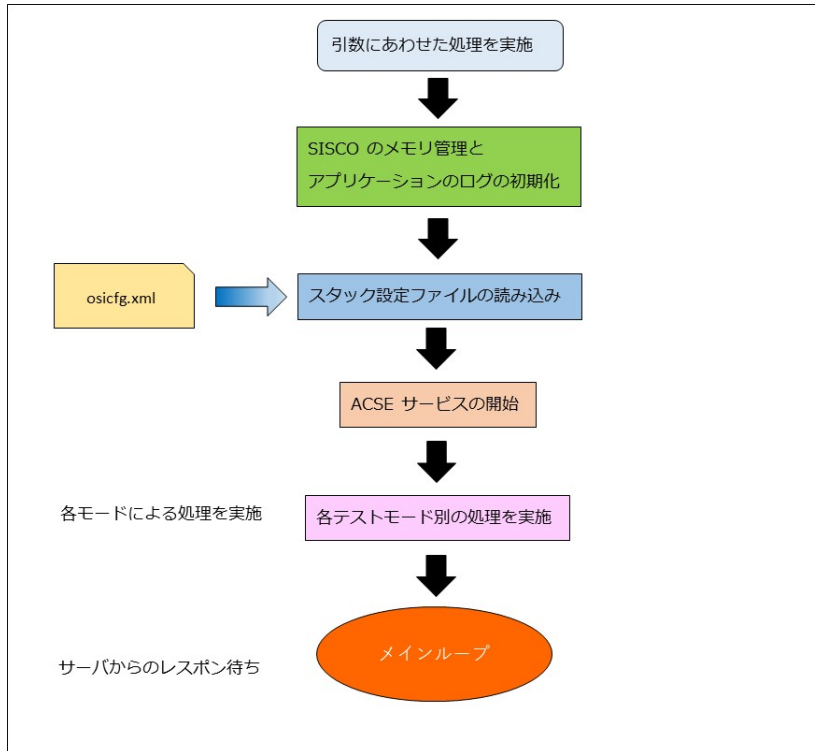
-m オプションでテストモードを指定します。

オプションのテストモードに関しては以下のようになります。

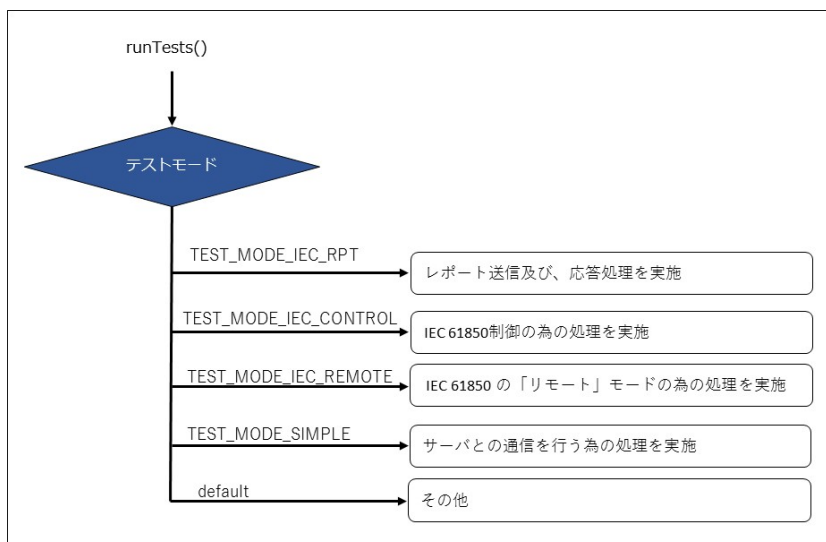
テストモード	定義	説明
未指定	TEST_MODE_SIMPLE	MVL サンプルサーバと MMS 通信のテストを行うモード MMS のリクエストを送信し、レスポンスを待ちます。 ※IEC61850/UCA に非対応
-m iecrpt	TEST_MODE_IEC_RPT	IEC 61850 レポートの受信のテストを行うモード
-m ieccontrol	TEST_MODE_IEC_CONTROL	IEC 61850 制御のテストを行うモード
-m iecremote <i>scIFileName</i> <i>iedName apName</i>	TEST_MODE_IEC_REMOTE	SCL ファイルを読み込む IEC 61850 の「リモート」モード (この実行は SCL ファイルを読み込むため、引数に SCL のファイル名、IED 名、アクセスポイント名を指定する必要がある)
-m identify <i>iterations</i>	TEST_MODE_IDENTIFY	MMS の IDENTIFY のリクエストの packets だけを <i>iterations</i> 回数繰り返し送るだけのモード(サーバの負荷試験用)

### 7.3.5.2 動作処理概要

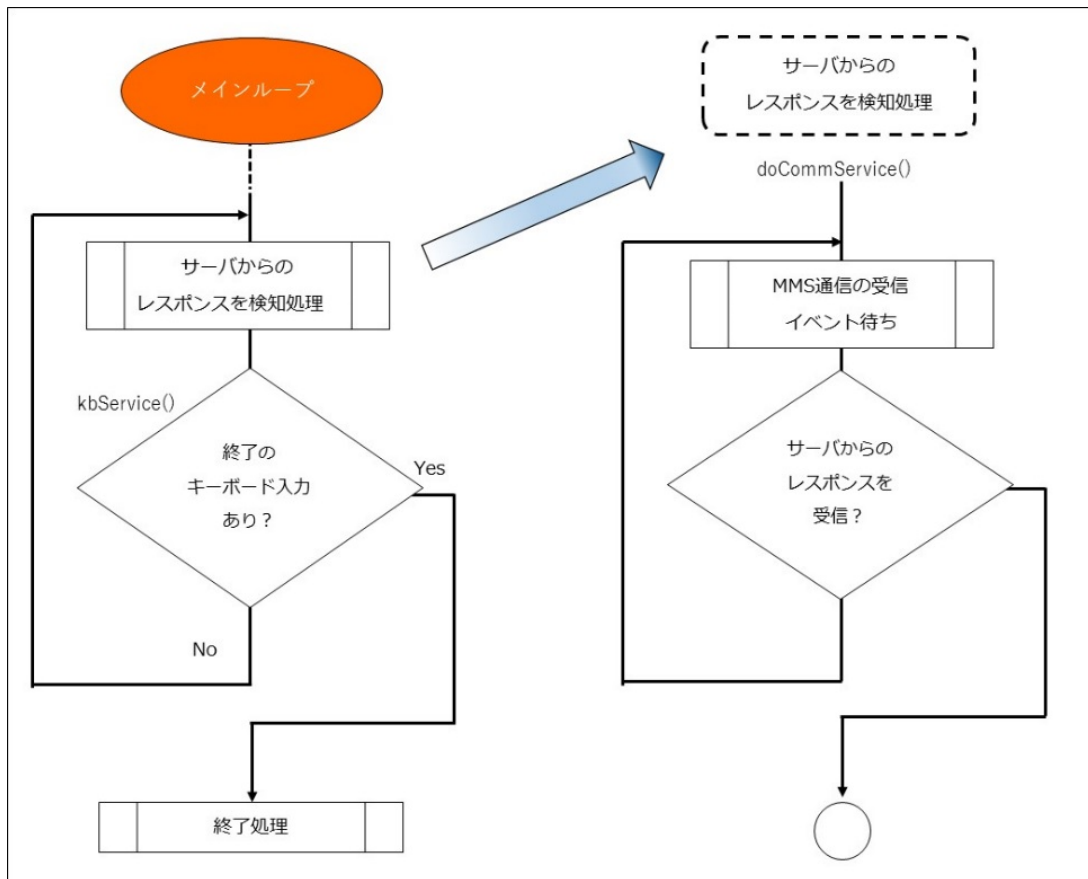
cositcps0の処理フローは、以下の手順にて各テストモードに合わせた処理を実施した後、サーバからのレスポンス待ちを行うメインループ処理を行います。



「各テストモード別の処理を実施」部分の処理については、以下の関数にて、各テストに合わせた処理を行います。



メインループの処理に関しては、以下の様なフロー図の様に、サーバからのレスポンスを待つ処理を行います。



■ テストモード

テストモードにおける動作について説明します。

① MVL サンプルサーバと MMS 通信のテストモード(モード未設定)

本モードでは、単純な MMS でのリクエスト・リプライの処理を行います。厳密には IEC61850の規格に沿った動作を行っておりません。

また、対向機器であるサーバに関しては、同じサンプルアプリケーションの scl\_srvr\_id を起動していることが前提条件になります。その際、scl\_srvr\_id は、デフォルト設定のまま起動してください。

次に、接続先(scl\_srvr\_id)の IP アドレスは SCL ファイルでの設定方法ではなく、osicfg.xml の方で設定する必要があります。

方法としては、mvl/usr/client の下にある osicfg.xml ファイルを開き、<RemoteAddress>の設定を scl\_srvr のアドレス情報に置き換えてください。

尚、cositcps0 はサンプルのため、接続先の AR 名がハードコートされている状態になっています。

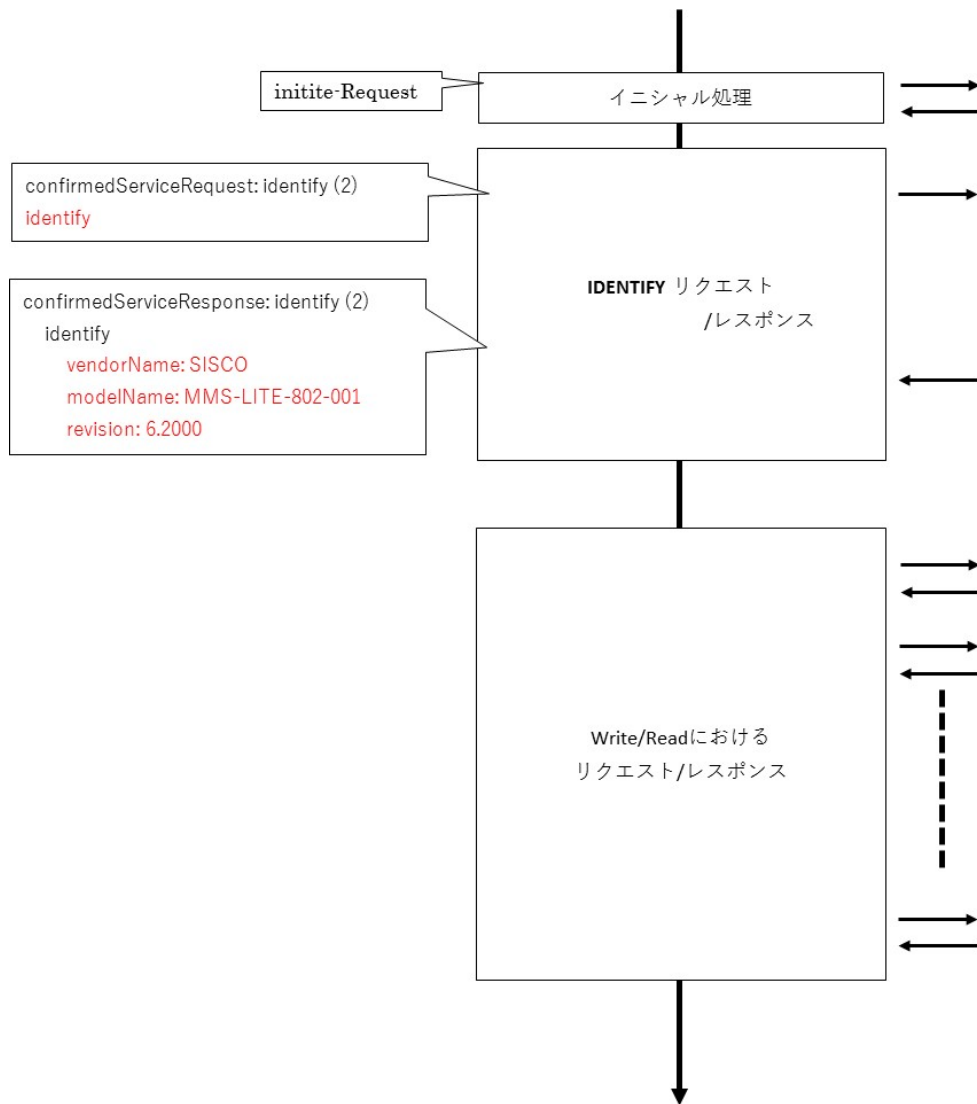
その為、接続するために、client.c の main の中で serverName に設定している AR 名を "E1Q1SB1/S1" に変更する必要があります。

例.

```
<RemoteAddressList>
  <RemoteAddress>
    <AR_Name> E1Q1SB1/S1 </AR_Name>
    <AP_Title>1 1 9999 1</AP_Title>
    <AE_Qualifier>101</AE_Qualifier>
    <Psel>00 00 00 01</Psel>
    <Ssel>00 01</Ssel>
    <Tsel>00 01</Tsel>
    <NetAddr Type="IPADDR">192.168.0.1</NetAddr>
  </RemoteAddress>
</RemoteAddressList>
```



動作手順としては、基本的には以下の様な動作を行います。



② IEC61850 レポートテストモード (iecrpt)

テストモード iecrpt を指定すると、test\_iec\_rpt 関数にて、IEC61850サーバの RCB (レポート・コントロール・ブロック) に対して有効にするためのパケットを送信し、その後、サーバに対してレポート送信を Request(要求)します。

デフォルトではサーバの以下のドメイン名及び、2つの RCB に対して実施します。

**ドメイン名**

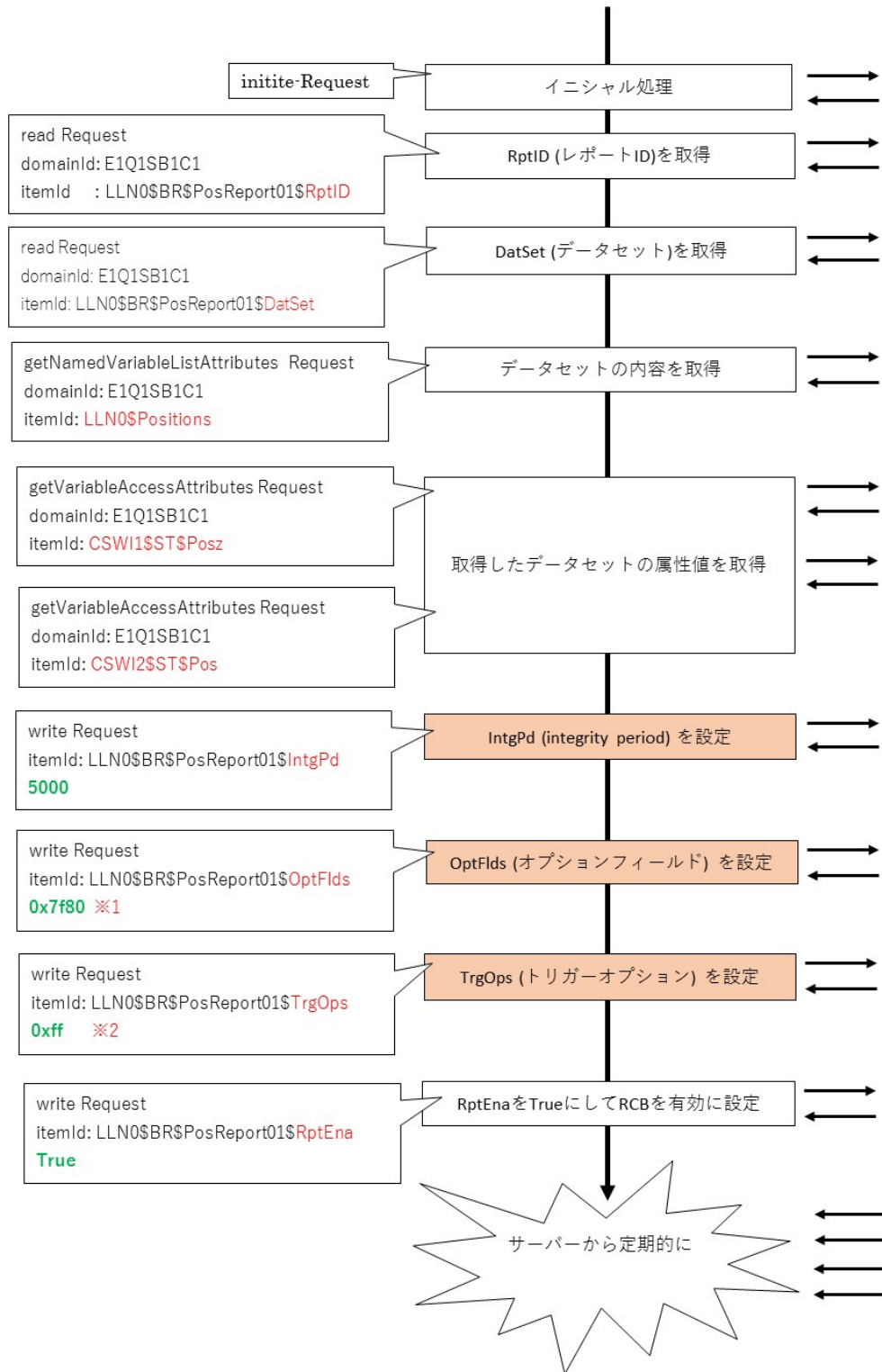
- E1Q1SB1C1

**RCB**

- LLN0\$BR\$PosReport01
- LLN0\$RP\$MeaReport01

※ 上記の設定自体は直接ソースコードに記載されております。

動作手順としては、基本的には以下の様な動作を行います。



※1 下記オプションをレポートに含める  
 sequence-number, report-time-stamp, reason-for-inclusion,  
 data-set-name, data-reference, buffer-overflow, entryID, conf-revision

※2 全トリガーオプションを有効とする  
 data-change, quality-change, data-update, integrity, general-interrogation

### ③ リモートテストモード (iecontrol)

引数 -m iecremote で実行した場合、引数にて指定された SCL ファイルを読み込み、その後、IEC 61850 のクライアントのモードで実行されます。

まず、動作させる準備として、SCL ファイルを読み込むために、サーバで使用している SCL ファイルを cositcps0と同じディレクトリにコピーする必要があります。

また、デフォルトからの変更箇所は、コピーした SCL ファイルの以下の箇所をサーバの情報に変更してください。

```
<ConnectedAP iedName="E1Q1SB1" apName="S1">
  <Address>
    <P type="IP">192.168.0.1</P>
    <P type="IP-SUBNET">255.255.255.0</P>
```

基本的な動作手順としては、前述の「MVL サンプルサーバと MMS 通信のテストモード(モード未設定)」と同じ動作になります。

ただし、動作結果としてはログファイル(mms.log)に出力されます。

### ④ リモートテストモード (iecontrol)

引数 -m iecontrol で実行した場合、サーバに対して制御処理を実施します。

処理としては、test\_iec\_control 関数にて、IEC61850サーバの以下のドメイン名、2つの SBO 及び、2つの Oper に対してタイミングを調整するために各4回実施します。

#### ドメイン名

- E1Q1SB1C1

#### SBO(Select Before Operate)

- CSWI1\$CO\$Mod\$SBO
- CSWI1\$CO\$Pos\$SBO

#### Oper

- CSWI1\$CO\$Mod\$Oper
- CSWI1\$CO\$Pos\$Oper

※ 上記の設定自体は直接ソースコードに記載されております。

