

# 装置用Control & Communication システムプロファイル パーサユーザーズマニュアル



CC-Link協会  
<http://www.cc-link.org/>

## 1. 概要

本書は、装置用 CSP+(The Control & Communication System Profile for machine)情報を解析するにあたり、装置用 CSP+パーサを使用するアプリケーション開発担当者向けのユーザズマニュアルです。

## 2. 用語定義

本マニュアルでは、特に明記する場合を除き、下記の総称・略称を使用して説明します。

表 2-1 用語一覧

総称・略称	内容
装置用 CSP+	The Control & Communication System Profile for machine の日本語名称
装置用 CSP+パーサ	装置用 CSP+の記述情報を解析するための機能を提供するモジュール
セクション	装置用 CSP+における最上位階層を示すキーワード 装置用 CSP+では、以下のセクションが存在する。 <ul style="list-style-type: none"><li>・ FILE セクション</li><li>・ DEVICE セクション</li><li>・ COMM_IF セクション</li><li>・ BLOCK セクション</li></ul>
パート	装置用 CSP+におけるセクションの下位層を示すキーワード 装置用 CSP+では以下のパートが存在する。 <ul style="list-style-type: none"><li>・ FILE_INFO パート</li><li>・ DEVICE_INFO パート</li><li>・ DEVICE_IF パート</li><li>・ COMM_IF_INFO パート</li><li>・ BLOCK_INFO パート</li><li>・ COMM_IF_VARIABLE パート</li><li>・ COMM_IF_CONFIGURATION パート</li><li>・ BLOCK_MEMORY パート</li><li>・ BLOCK_PARAM パート</li><li>・ ENUM パート</li></ul>
要素	装置用 CSP+におけるパートの下位層を示すキーワード 装置用 CSP+では、要素は複数の項目を持つ。
項目	装置用 CSP+における要素の下位層を示すキーワード 装置用 CSP+では、以下の項目が存在する。 <ul style="list-style-type: none"><li>・ LABEL</li><li>・ LABEL2</li><li>・ CATEGORY</li><li>・ NAME</li><li>・ DATATYPE</li><li>・ DATA</li><li>・ RANGE</li><li>・ MIN_INC</li><li>・ ENG_UNIT</li><li>・ ACCESS</li><li>・ ASSIGN</li><li>・ CODE</li><li>・ REF_MEMORY</li><li>・ REF_PARAM</li><li>・ COMMENT</li></ul>

### 3. 機能

#### 3.1. サポート対象

本機能は CC-Link 協会に公開している「装置用 Control & Communication システムプロファイル仕様書」の仕様をサポートしています。

#### 3.2. 提供機能

本モジュールは、以下の機能を提供します。

表 3.2-1 提供機能

No.	名称	内容
1	装置用 CSP+パーサ	装置用 CSP+の情報を利用したいアプリケーションに対して、装置用 CSP+の記述内容を取得し、解析するための機能を提供する。

### 4. アプリケーションの開発

#### 4.1. 開発環境

本書に属するモジュールは「VC++2010(Visual Studio 2010 Visual C++)」で開発しています。本機能を使用して開発する要件を以下に示します。

表 4.1-1 開発環境の要件

No.	開発環境の要件
1	Microsoft Visual Studio がインストールされている。(※1)
2	MSXML 4.0 Service Pack 2 以上がインストールされている。 ※インストールされていない場合は、以下の Microsoft サイトよりインストール可能。 < <a href="http://www.microsoft.com/downloads/ja-jp/details.aspx?FamilyID=3144b72b-b4f2-46da-b4b6-c5d7485f2b42">http://www.microsoft.com/downloads/ja-jp/details.aspx?FamilyID=3144b72b-b4f2-46da-b4b6-c5d7485f2b42</a> >

※1 Microsoft Visual Studio 2008 または 2010 を推奨します。

#### 4.2. 開発手順

##### 4.2.1. 提供モジュール

提供モジュールは「表 4.2.1-1」に示すプログラム部品で構成します。

表 4.2.1-1 提供するモジュール構成

No.	フォルダ名	ファイル名	説明
1	Root	—	—
2	Debug	CSPPlusParserForMachine.dll	装置用 CSP+の記述内容を取得し、解析する動的ライブラリ(Debug 版)
3		CSPPlusParserForMachine.lib	No.2 を使用するための静的ライブラリ(Debug 版)
4	Release	CSPPlusParserForMachine.dll	No.2 の Release 版
5		CSPPlusParserForMachine.lib	No.3 の Release 版
6	Header	CSPPlusParserForMachine.h	No.2 または No.4 の機能を使用するためのヘッダファイル。使用するプロジェクトに取り込み、No.1 の機能を使用する。

※No.2～5 のモジュールについては、使用するアプリケーションのビルド形式によって、使い分けて下さい。

#### 4.2.2. 開発手順

Microsoft Visual Studio2010 にて、「表 4.2.1-1」のモジュールを使用して、アプリケーションを開発する例を以下に示します。

表 4.2.2-1 開発手順

No.	開発手順
1	プロジェクトの「プロパティ」→「構成プロパティ」→「C/C++」→「全般」→「追加のインクルード ディレクトリ」に「表 4.2.1-1」の No.6 のディレクトリを指定する。
2	プロジェクトの「プロパティ」→「構成プロパティ」→「リンカー」→「全般」→「追加のライブラリ ディレクトリ」に「表 4.2.1-1」の No.5 のディレクトリを指定する。
3	プロジェクトの「プロパティ」→「構成プロパティ」→「リンカー」→「入力」→「追加の依存ファイル」に「表 4.2.1-1」の No.5 のファイル名を指定する。
4	プロジェクトの「プロパティ」→「構成プロパティ」→「ビルドイベント」→「ビルド後イベント」→「コマンドライン」に「表 4.2.1-1」の No.4 のモジュールをアプリケーションのモジュール配下にコピーするスクリプトを指定する。

※使用するアプリケーションのビルド形式によって、指定するモジュールが異なるので注意して下さい。

「表 4.2.2-1」では使用側のビルド形式が、Release 版の場合を例に記載しています。

#### 4.3. 動作環境

本機能を動作させる環境の要件を以下に示します。

表 4.3-1 動作環境の要件

No.	動作環境の要件
1	Microsoft Visual C++ Redistributable がインストールされている。(※1)
2	MSXML 4.0 Service Pack 2 以上がインストールされている。 インストールについては、「表 4.1-1の No.2」を参照。

※1 Microsoft Visual C++ Redistributable 2008 または 2010 を推奨します。

#### 4.4. 定義一覧

本節では装置用 CSP+パーサを使用する際に必要な定義一覧を示します。

##### 4.4.1. 列挙(ENUM)型

装置用 CSP+パーサで使用する ENUM 型について、以下に示します。

表 4.4.1-1 セクション種別の定義一覧(ECSPFMDEF\_SECTIONTYPE)

No.	意味	定義名	値	備考
1	未知セクション	ECSPFMDEF_SECTIONTYPE_UNKNOWN	0	
2	FILE セクション	ECSPFMDEF_SECTIONTYPE_FILE	1	
3	DEVICE セクション	ECSPFMDEF_SECTIONTYPE_DEVICE	2	
4	COMM_IF セクション	ECSPFMDEF_SECTIONTYPE_COMMIF	3	
5	BLOCK セクション	ECSPFMDEF_SECTIONTYPE_BLOCK	4	
6	セクション種別最大値	ECSPFMDEF_SECTIONTYPE_MAX	5	

表 4.4.1-2 パート種別の定義一覧(ECSPFMDEF\_PARTTYPE)

No.	意味	定義名	値	備考
1	未知パート	ECSPFMDEF_PARTTYPE_UNKNOWN	0	
2	FILE_INFO パート	ECSPFMDEF_PARTTYPE_FILE_INFO	1	
3	DEVICE_INFO パート	ECSPFMDEF_PARTTYPE_DEVICE_INFO	2	
4	DEVICE_IF パート	ECSPFMDEF_PARTTYPE_DEVICE_IF	3	
5	COMMIF_INFO パート	ECSPFMDEF_PARTTYPE_COMMIF_INFO	4	
6	COMMIF_VARIABLE パート	ECSPFMDEF_PARTTYPE_COMMIF_VARIABLE	5	
7	COMMIF_CONFIGURATION パート	ECSPFMDEF_PARTTYPE_COMMIF_CONFIGURATION	6	
8	BLOCK_INFO パート	ECSPFMDEF_PARTTYPE_BLOCK_INFO	7	
9	BLOCK_MEMORY パート	ECSPFMDEF_PARTTYPE_BLOCK_MEMORY	8	
10	BLOCK_PARAM パート	ECSPFMDEF_PARTTYPE_BLOCK_PARAM	9	
11	ENUM パート	ECSPFMDEF_PARTTYPE_ENUM	10	
12	パート種別最大値	ECSPFMDEF_PARTTYPE_MAX	11	

表 4.4.1-3 項目種別の定義一覧(ECSPFMDEF\_ITEMTYPE)

No.	意味	定義名	値	備考
1	未知項目	ECSPFMDEF_ITEMTYPE_UNKNOWN	0	
2	LABEL 項目	ECSPFMDEF_ITEMTYPE_LABEL	1	
3	LABEL2 項目	ECSPFMDEF_ITEMTYPE_LABEL2	2	
4	CATEGORY 項目	ECSPFMDEF_ITEMTYPE_CATEGORY	3	
5	NAME 項目	ECSPFMDEF_ITEMTYPE_NAME	4	
6	DATATYPE 項目	ECSPFMDEF_ITEMTYPE_DATATYPE	5	
7	DATA 項目	ECSPFMDEF_ITEMTYPE_DATA	6	
8	RANGE 項目	ECSPFMDEF_ITEMTYPE_RANGE	7	
9	MIN_INC 項目	ECSPFMDEF_ITEMTYPE_MIN_INC	8	
10	ENG_UNIT 項目	ECSPFMDEF_ITEMTYPE_ENG_UNIT	9	
11	ACCESS 項目	ECSPFMDEF_ITEMTYPE_ACCESS	10	
12	ASSIGN 項目	ECSPFMDEF_ITEMTYPE_ASSIGN	11	
13	CODE 項目	ECSPFMDEF_ITEMTYPE_CODE	12	
14	REF_MEMORY 項目	ECSPFMDEF_ITEMTYPE_REF_MEMORY	13	
15	REF_PARAM 項目	ECSPFMDEF_ITEMTYPE_REF_PARAM	14	
16	COMMENT 項目	ECSPFMDEF_ITEMTYPE_COMMENT	15	
17	項目種別最大値	ECSPFMDEF_ITEMTYPE_MAX	16	

表 4.4.1-4 項目データ取得設定種別の定義一覧(ECSPFMDEF\_GETITEMTYPE)

No.	意味	定義名	値	備考
1	何もしない	ECSPFMDEF_GETITEMTYPE_NONE	0	要素から指定項目を検索して取得する。
2	再帰的に取得する	ECSPFMDEF_GETITEMTYPE_RECURSIVE	2	現在の要素から指定項目を検索する。見つからない、かつ REF 項目より他の要素への参照が記述されている場合、その参照先から見つかるまで取得する。

表 4.4.1-5 範囲型データの制限タイプ種別の定義一覧(ELIMITTYPE)

No.	意味	定義名	値	備考
1	限界値の指定なし	CSPFMDEF_LIMITTYPE_NON	0	
2	小なり (<)	CSPFMDEF_LIMITTYPE_LESS	1	
3	大なり (>)	CSPFMDEF_LIMITTYPE_GREATER	2	
4	小なりイコール (<=)	CSPFMDEF_LIMITTYPE_LESSEQUAL	3	
5	大なりイコール (>=)	CSPFMDEF_LIMITTYPE_GREATEREQUAL	4	

#### 4.4.2. 構造体(STRUCT)型

装置用 CSP+パーサで使用する構造体型について、以下に示します。

表 4.4.2-1 範囲型データの構造体定義一覧(ST\_CSPFM\_RANGE\_DATA)

No.	意味	メンバ名	型	備考
1	最小値	strMinData	CString	
2	最小値の制限タイプ	dwMinType	ELIMITTYPE	
3	最大値	strMaxData	CString	
4	最大値の制限タイプ	dwMaxType	ELIMITTYPE	
5	最小値の項目	pMinItemInfo	CCSPFM_ItemInfo*	
6	最大値の項目	pMaxItemInfo	CCSPFM_ItemInfo*	

#### 4.4.3. データ種別

装置用 CSP+パーサで使用するデータ種別一覧について、以下に示します。

表 4.4.3-1 CheckDataType 関数で返却されるデータ種別の定義一覧(EDATATYPE)

No.	意味	定義名	値	備考
1	チェックなし	CSPFMDEF_DATATYPE_NON_CHECK	0x00000000	
2	列挙型	CSPFMDEF_DATATYPE_ENUM	0x00000001	
3	複数型(カンマ区切り)	CSPFMDEF_DATATYPE_MULTIPLE	0x00000002	
4	複数型(山括弧区切り)	CSPFMDEF_DATATYPE_BRACKETS	0x00000003	
5	範囲型	CSPFMDEF_DATATYPE_RANGE	0x00000004	
6	文字列型	CSPFMDEF_DATATYPE_STRING	0x00000005	
7	参照型(パート)	CSPFMDEF_DATATYPE_REF_PART	0x00000007	

#### 4.4.4. エラーコード一覧

装置用 CSP+パーサで返却されるエラーコード一覧について、以下に示します。

表 4.4.4-1 エラーコード一覧

No.	エラーコード	意味	備考
1	CSPFMERR_EXCEPTION_FILENOTFOUND	指定の装置用 CSP+ファイルが見つからない	
2	CSPFMERR_EXCEPTION_RANGE	引数の指定が無効	
3	CSPFMERR_EXCEPTION_INVALID_VERSION	装置用 CSP+ファイル記述仕様バージョンに未対応	
4	CSPFMERR_EXCEPTION_MEMORY	メモリ確保に失敗した	
5	CSPFMERR_EXCEPTION_CSPLOAD	指定の装置用 CSP+ファイルの解析に失敗した	
6	CSPFMERR_EXCEPTION_INVALID_DESCRIPTION	装置用 CSP+ファイルの記述異常	
7	CSPFMERR_EXCEPTION_INVALID_APICALL	呼び出し API が無効	



## 4.5. 関数一覧

提供する装置用 CSP+パーサで使用可能な関数一覧を以下に示します。

### 4.5.1. CCSPFM\_ProfileData クラス

表 4.5.1-1 CCSPFM\_ProfileData クラスの関数一覧

No.	関数名	関数概要	詳細仕様
1	CCSPFM_ProfileData ()	装置用 CSP+パーサのコンストラクタ 解析対象の装置用 CSP+ファイルのフルパスを指定し、解析準備を行う。	コンストラクタのため、省略
2	~ CCSPFM_ProfileData ()	装置用 CSP+パーサのデストラクタ機能	デストラクタのため、省略
3	Get()	全セクションを取得する機能	4.7.1.1を参照
4	Get()	内部変数のセクション一覧のポインタを取得する機能	4.7.1.2を参照
5	Get()	セクションのラベル名を指定して、セクションを一つ取得する機能	4.7.1.3を参照
6	Get()	セクション種別を指定して、0 個以上のセクションを取得する機能	4.7.1.4を参照

### 4.5.2. CCSPFM\_SectionInfo クラス

表 4.5.2-1 CCSPFM\_SectionInfo クラスの関数一覧

No.	関数名	関数概要	詳細仕様
1	CCSPFM_SectionInfo()	SectionInfo クラスのコンストラクタ	コンストラクタのため、省略
2	~CCSPFM_SectionInfo()	SectionInfo クラスのデストラクタ	デストラクタのため、省略
3	Get()	全パートを取得する機能	4.7.2.1を参照
4	Get()	内部変数のパート一覧のポインタを取得する機能	4.7.2.2を参照
5	Get()	パートのラベル名を指定して、パートを一つ取得する機能	4.7.2.3を参照
6	Get()	パート種別を指定して、0 個以上のパートを取得する機能	4.7.2.4を参照

### 4.5.3. CCSPFM\_PartInfo クラス

表 4.5.3-1 CCSPFM\_PartInfo クラスの関数一覧

No.	関数名	関数概要	詳細仕様
1	CCSPFM_PartInfo()	PartInfo クラスのコンストラクタ	コンストラクタのため、省略
2	~CCSPFM_PartInfo()	PartInfo クラスのデストラクタ	デストラクタのため、省略
3	Get()	全要素を取得する機能	4.7.3.1を参照
4	Get()	要素のラベル名を指定して、要素を一つ取得する機能	4.7.3.2を参照

5	Get()	内部変数の要素一覧のポインタを取得する機能	4.7.3.3を参照
6	GetItemList()	項目種別を指定して、項目を取得する機能	4.7.3.4を参照

#### 4.5.4. CCSPFM\_ElemInfo クラス

表 4.5.4-1 CCSPFM\_ElemInfo クラスの関数一覧

No.	関数名	関数概要	詳細仕様
1	CCSPFM_ElemInfo()	ElemInfo クラスのコンストラクタ	コンストラクタのため、省略
2	~CCSPFM_ElemInfo()	ElemInfo クラスのデストラクタ	デストラクタのため、省略
3	Get()	全項目を取得する機能	4.7.4.1を参照
4	Get()	内部変数の項目一覧のポインタを取得する機能	4.7.4.2を参照
5	Get()	項目種別を指定して項目を取得する機能	4.7.4.3を参照
6	GetCommIFRef()	REF 項目で参照している要素を取得する機能	4.7.4.4を参照
7	ReleaseItemArray()	項目のポインタ配列を解放する機能	4.7.4.5を参照

#### 4.5.5. CCSPFM\_ItemInfo クラス

表 4.5.5-1 CCSPFM\_ItemInfo クラスの関数一覧

No.	関数名	関数概要	詳細仕様
1	CCSPFM_ItemInfo()	ItemInfo クラスのコンストラクタ	コンストラクタのため、省略
2	~CCSPFM_ItemInfo()	ItemInfo クラスのデストラクタ	デストラクタのため、省略
3	CheckDataType()	データ種別を判定する機能	4.7.5.1を参照
4	GetStringData()	データを文字列として取得する機能	4.7.5.2を参照
5	GetEnumData()	データを列挙型として取得する機能	4.7.5.3を参照
6	GetRangeData()	データを範囲型として取得する機能	4.7.5.4を参照
7	GetMultipleData()	データを複数データ(カンマ区切り)として取得する機能	4.7.5.5を参照
8	GetMultipleDataEX()	データを複数データ(カンマ区切り)として取得する機能(参照取得)	4.7.5.6を参照
9	GetBracketsData()	データを複数データ(山括弧区切り)として取得する機能	4.7.5.7を参照
10	GetRefPart()	参照先のパートを取得する機能	4.7.5.8を参照
11	GetRawData()	データについて、記述を考慮せず文字列として取得する機能	4.7.5.9を参照
12	GetDataTypeSize()	データサイズを取得する機能	4.7.5.10を参照
13	GetSize()	指定されたデータサイズに対応するビットサイズを取得する機能	4.7.5.11を参照

#### 4.5.6. CCSPFM\_ExceptionBase クラス

表 4.5.6-1 CCSPFM\_ExceptionBase クラスの関数一覧

No.	関数名	関数概要	詳細仕様
1	CCSPFM_ExceptionBase ()	ExceptionBase クラスのコンストラクタ	コンストラクタのため、省略
2	~CCSPFM_ExceptionBase ()	ExceptionBase クラスのデストラクタ	デストラクタのため、省略
3	GetErrCode()	エラーコードを取得する機能	4.7.6.1を参照

#### 4.6. 装置用 CSP+パーサ呼び出し手順

装置用 CSP+パーサ呼び出し手順を以下に示します。

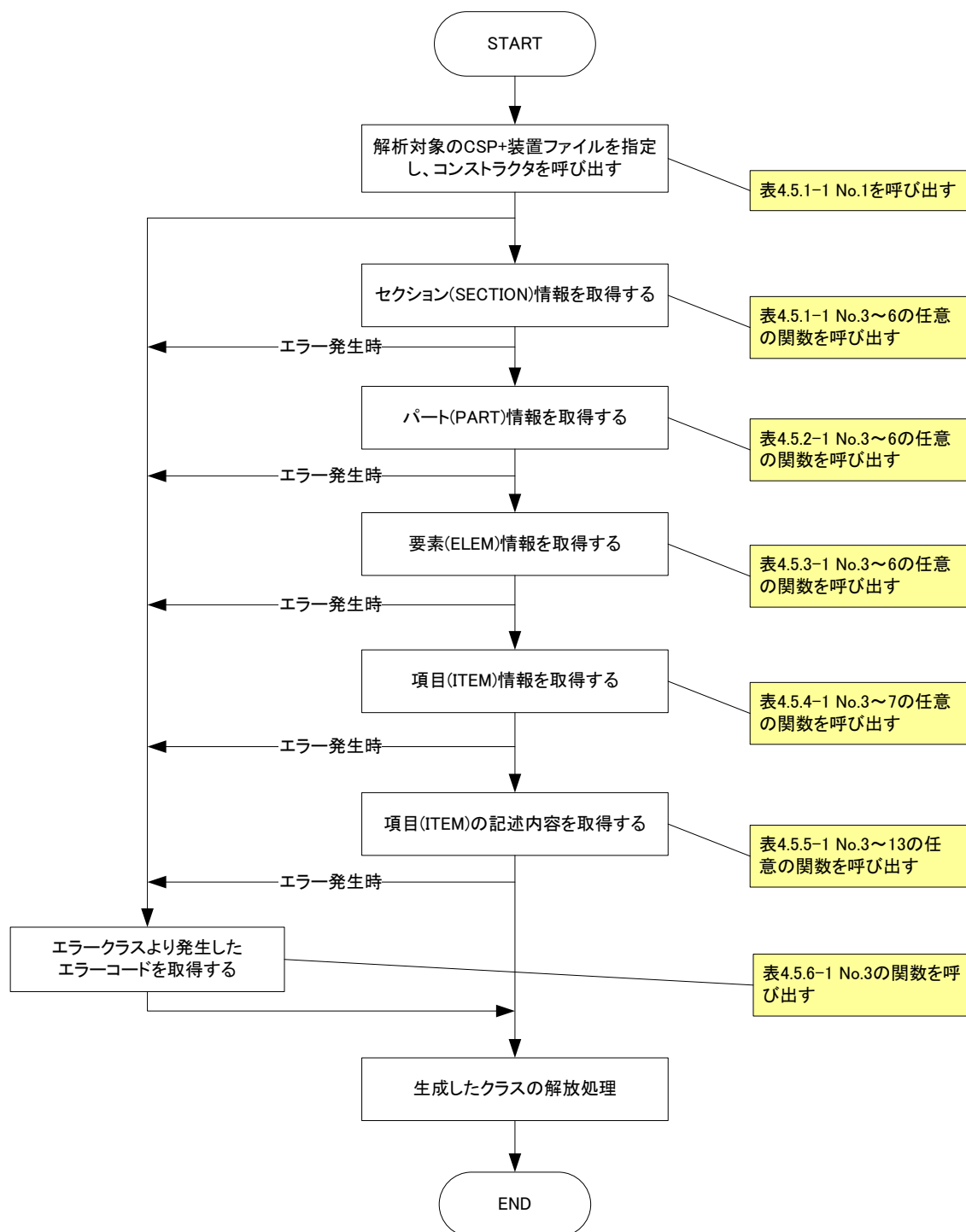


図 4.6-1 装置用 CSP+パーサ呼び出し手順

#### 4.7. 各関数の詳細仕様

各関数の詳細仕様について、以下に示します。

##### 4.7.1. CCSPFM\_ProfileData クラス

##### 4.7.1.1. Get([OUT] CCSPFM\_SectionInfoArray&)

###### (1) 関数概要

関数名	引数	戻り値
Get()	(OUT) CCSPFM_SectionInfoArray&	void

###### (2) パラメータ

CCSPFM\_SectionInfoArray& : (OUT)取得できたセクションの一覧

###### (3) 戻り値

なし

###### (4) 機能

全セクションを取得する。

OUT 引数に設定される各メンバは CCSPFM\_SectionInfo の情報となる。

###### (5) API 使用上の注意事項

特になし

#### 4.7.1.2. Get(void)

(1) 関数概要

関数名	引数	戻り値
Get()	void	CCSPFM_SectionInfoArray*

(2) パラメータ

なし

(3) 戻り値

CCSPFM\_SectionInfoArray\* : 内部変数のセクション一覧のポインタ

(4) 機能

内部変数のセクション一覧のポインタを取得する。

戻り値に設定される各メンバは CCSPFM\_SectionInfo の情報となる。

(5) API 使用上の注意事項

特になし

#### 4.7.1.3. Get([IN] CString)

(1) 関数概要

関数名	引数	戻り値
Get()	(IN) CString	CCSPFM_SectionInfo*

(2) パラメータ

CString : (IN)取得したいセクションのラベル名

(3) 戻り値

CCSPFM\_SectionInfo\* : 取得できたセクションへのポインタ

(4) 機能

セクションのラベル名を指定して、セクションを一つ取得する。

(5) API 使用上の注意事項

IN 引数に指定するラベル名はフルパスではなく、取得したいセクションのラベル名のみを指定する。

#### 4.7.1.4. Get([IN] DWORD,[OUT] CCSPFM\_SectionInfoArray&)

##### (1) 関数概要

関数名	引数	戻り値
Get()	(IN) DWORD (OUT) CCSPFM_SectionInfoArray&	void

##### (2) パラメータ

DWORD : (IN)取得したいセクション種別  
CCSPFM\_SectionInfoArray& : (OUT)取得できたセクションの一覧

##### (3) 戻り値

なし

##### (4) 機能

セクション種別を指定して、セクションの一覧を取得する。  
IN 引数に指定するセクション種別の値については、「表 4.4.1-1」を参照。

##### (5) API 使用上の注意事項

特になし



#### 4.7.2. CCSPFM\_PartInfo クラス

##### 4.7.2.1. Get([OUT] CCSPFM\_PartInfoArray&)

###### (1) 関数概要

関数名	引数	戻り値
Get()	(OUT) CCSPFM_PartInfoArray&	void

###### (2) パラメータ

CCSPFM\_PartInfoArray& : (OUT)取得できたパートの一覧

###### (3) 戻り値

なし

###### (4) 機能

全パートを取得する。

OUT 引数に設定される各メンバは CCSPFM\_PartInfo の情報となる。

###### (5) API 使用上の注意事項

特になし

#### 4.7.2.2. Get(void)

(1) 関数概要

関数名	引数	戻り値
Get()	void	CCSPFM_PartInfoArray*

(2) パラメータ

なし

(3) 戻り値

CCSPFM\_PartInfoArray\* : 内部変数のパート一覧のポインタ

(4) 機能

内部変数のパート一覧のポインタを取得する。

戻り値に設定される各メンバは CCSPFM\_PartInfo の情報となる。

(5) API 使用上の注意事項

特になし

#### 4.7.2.3. Get([IN] CString)

(1) 関数概要

関数名	引数	戻り値
Get()	(IN) CString	CCSPFM_PartInfo*

(2) パラメータ

CString : (IN)取得したいパートのラベル名

(3) 戻り値

CCSPFM\_PartInfo\* : 取得できたパートへのポインタ

(4) 機能

パートのラベル名を指定して、パートを一つ取得する。

(5) API 使用上の注意事項

IN 引数に指定するラベル名はフルパスではなく、取得したいパートのラベル名のみを指定する。

#### 4.7.2.4. Get([IN] DWORD,[OUT] CCSPFM\_PartInfoArray&)

##### (1) 関数概要

関数名	引数	戻り値
Get()	(IN) DWORD (OUT) CCSPFM_PartInfoArray&	void

##### (2) パラメータ

DWORD : (IN)取得したいパート種別  
CCSPFM\_PartInfoArray& : (OUT)取得できたパートの一覧

##### (3) 戻り値

なし

##### (4) 機能

パートの種別を指定して、パートを一つ取得する。  
IN 引数に指定するパート種別の値については、「表 4.4.1-2」を参照。  
OUT 引数に設定される各メンバは CCSPFM\_PartInfo の情報となる。

##### (5) API 使用上の注意事項

特になし

### 4.7.3. CCSPFM\_PartInfo クラス

#### 4.7.3.1. Get([OUT] CCSPFM\_ElemInfoArray&)

##### (1) 関数概要

関数名	引数	戻り値
Get()	(OUT) CCSPFM_ElemInfoArray&	void

##### (2) パラメータ

CCSPFM\_ElemInfoArray& : (OUT)取得できた全要素の一覧

##### (3) 戻り値

なし

##### (4) 機能

全要素を取得する。

OUT 引数に設定される各メンバは CCSPFM\_ElemInfo の情報となる。

##### (5) API 使用上の注意事項

特になし

#### 4.7.3.2. Get([IN] CString)

(1) 関数概要

関数名	引数	戻り値
Get()	(IN) CString	CCSPFM_ElemInfo*

(2) パラメータ

CString : (IN)取得したい要素のラベル名

(3) 戻り値

CCSPFM\_ElemInfo\* : 取得できた要素へのポインタ

(4) 機能

要素のラベル名を指定して、要素を一つ取得する。

(5) API 使用上の注意事項

IN 引数に指定するラベル名はフルパスではなく、取得したい要素のラベル名のみを指定する。

#### 4.7.3.3. Get(void)

(1) 関数概要

関数名	引数	戻り値
Get()	void	CCSPFM_ElemInfoArray*

(2) パラメータ

なし

(3) 戻り値

CCSPFM\_ElemInfoArray\* : 内部変数の要素一覧のポインタ

(4) 機能

内部変数の要素一覧のポインタを取得する。

戻り値に設定される各メンバは CCSPFM\_ElemInfo の情報となる。

(5) API 使用上の注意事項

特になし

#### 4.7.3.4. GetItemList([IN] DWORD,[OUT] CCSPFM\_ItemInfoArray&)

##### (1) 関数概要

関数名	引数	戻り値
GetItemList()	(IN) DWORD (OUT) CCSPFM_ItemInfoArray&	void

##### (2) パラメータ

DWORD : (IN)取得したい項目種別  
CCSPFM\_ItemInfoArray& : (OUT)取得できた項目の一覧

##### (3) 戻り値

なし

##### (4) 機能

指定する項目種別に一致する項目の一覧を取得する。  
IN 引数に指定する項目種別の値については、「表 4.4.1-3」を参照。

##### (5) API 使用上の注意事項

特になし



#### 4.7.4. CCSPFM\_ElemInfo クラス

##### 4.7.4.1. Get([OUT] CCSPFM\_ItemInfoArray&)

###### (1) 関数概要

関数名	引数	戻り値
Get()	(OUT) CCSPFM_ItemInfoArray&	void

###### (2) パラメータ

CCSPFM\_ItemInfoArray& : (OUT)取得できた全項目の一覧

###### (3) 戻り値

なし

###### (4) 機能

全項目を取得する。

OUT 引数に設定される各メンバは CCSPFM\_ItemInfo の情報となる。

###### (5) API 使用上の注意事項

特になし

#### 4.7.4.2. Get(void)

(1) 関数概要

関数名	引数	戻り値
Get()	void	CCSPFM_ItemInfoArray*

(2) パラメータ

なし

(3) 戻り値

CCSPFM\_ItemInfoArray\* : 内部変数の項目一覧のポインタ

(4) 機能

内部変数の項目一覧のポインタを取得する。

戻り値に設定される各メンバは CCSPFM\_ItemInfo の情報となる。

(5) API 使用上の注意事項

特になし

#### 4.7.4.3. Get([IN] DWORD,[IN] DWORD)

##### (1) 関数概要

関数名	引数	戻り値
Get()	(IN) DWORD (IN) DWORD	CCSPFM_ItemInfo*

##### (2) パラメータ

DWORD

: (IN)取得したい項目種別

DWORD

: (IN)取得設定方法 (使用しない)

##### (3) 戻り値

CCSPFM\_ItemInfo\*

: 取得できた項目へのポインタ

##### (4) 機能

全項目を取得する。

- ・ IN 引数に指定する「取得したい項目種別」の値については、「表 4.4.1-3」を参照。
- ・ IN 引数に指定する「取得設定方法」の値については、「表 4.4.1-4」を参照。  
指定しない場合、「ECSPFMDEF\_GETITEMTYPE\_NONE」がデフォルトで指定される。

##### (5) API 使用上の注意事項

特になし

#### 4.7.4.4. GetCommIFRef()

(1) 関数概要

関数名	引数	戻り値
GetCommIFRef()	—	CCSPFM_ElemInfo*

(2) パラメータ

なし

(3) 戻り値

CCSPFM\_ElemInfo\* : REF 項目に記述されている参照要素へのポインタ

(4) 機能

REF項目で参照している要素へのポインタを取得する。

(5) API 使用上の注意事項

装置用 CSP+では、常に NULL が返却される。

#### 4.7.4.5. ReleaseItemArray([OUT] CCSPFM\_ItemInfoArray&)

(1) 関数概要

関数名	引数	戻り値
ReleaseItemArray()	(OUT) CCSPFM_ItemInfoArray&	void

(2) パラメータ

CCSPFM\_ItemInfoArray& : (OUT)全項目クラスへのポインタ配列

(3) 戻り値

なし

(4) 機能

項目クラスのポインタ配列を解放する。

(5) API 使用上の注意事項

特になし

#### 4.7.5. CCSPFM\_ItemInfo クラス

##### 4.7.5.1. CheckDataType()

###### (1) 関数概要

関数名	引数	戻り値
CheckDataType()	—	DWORD

###### (2) パラメータ

なし

###### (3) 戻り値

DWORD : 指定項目のデータ種別

###### (4) 機能

指定した項目に記述されているデータ種別を取得する。  
戻り値に設定する値については、「表 4.4.3-1」を参照。

###### (5) API 使用上の注意事項

特になし

#### 4.7.5.2. GetStringData()

(1) 関数概要

関数名	引数	戻り値
GetStringData()	—	CString

(2) パラメータ

なし

(3) 戻り値

CString : 文字列型データ

(4) 機能

指定した項目に記述されているデータを文字列型として取得する。

(5) API 使用上の注意事項

CheckDataType()を実行した結果、データ種別が「表 4.4.3-1の No.6」の場合に本関数を使用する。

#### 4.7.5.3. GetEnumData()

(1) 関数概要

関数名	引数	戻り値
GetEnumData()	—	CCSPFM_PartInfo*

(2) パラメータ

なし

(3) 戻り値

CCSPFM\_PartInfo\* : 列挙型データのポインタ

(4) 機能

指定した項目に記述されているデータを列挙型として取得する。

(5) API 使用上の注意事項

CheckDataType()を実行した結果、データ種別が「表 4.4.3-1の No.2」の場合に本関数を使用する。



#### 4.7.5.4. GetRangeData()

(1) 関数概要

関数名	引数	戻り値
GetRangeData()	—	ST_CSPFM_RANGE_DATA

(2) パラメータ

なし

(3) 戻り値

ST\_CSPFM\_RANGE\_DATA : 範囲型データの構造体

(4) 機能

指定した項目に記述されているデータを範囲型として取得する。

(5) API 使用上の注意事項

CheckDataType()を実行した結果、データ種別が「表 4.4.3-1の No.5」の場合に本関数を使用する。  
この関数で取得した構造体において、以下のデータについては呼び出し元で解放すること。

「表 4.4.2-1の No.5」

「表 4.4.2-1の No.6」

#### 4.7.5.5. GetMultipleData([OUT] CCSPFM\_ItemInfoArray&)

(1) 関数概要

関数名	引数	戻り値
GetMultipleData()	(OUT) CCSPFM_ItemInfoArray&	void

(2) パラメータ

CCSPFM\_ItemInfoArray& : (OUT)複数データ(カンマ区切り)の配列

(3) 戻り値

なし

(4) 機能

指定した項目に記述されているデータを複数型(カンマ区切り)として取得する。

(5) API 使用上の注意事項

CheckDataType()を実行した結果、データ種別が「表 4.4.3-1の No.3」の場合に本関数を使用する。  
この関数で取得したポインタは呼び出し元で解放すること。  
解放する際は、「表 4.5.4-1の No.7」を呼び出すこと。

#### 4.7.5.6. GetMultipleDataEX ([OUT] CCSPFM\_ItemInfoArray&)

##### (1) 関数概要

関数名	引数	戻り値
GetMultipleDataEX ()	(OUT) CCSPFM_ItemInfoArray&	void

##### (2) パラメータ

CCSPFM\_ItemInfoArray& : (OUT)複数データ(カンマ区切り)の配列

##### (3) 戻り値

なし

##### (4) 機能

指定した項目に記述されているデータを複数型(カンマ区切り)として取得する。  
参照記述すれば、参照のすべて項目を取得する。

##### (5) API 使用上の注意事項

CheckDataType()を実行した結果、データ種別が「表 4.4.3-1の No.3」の場合に本関数を使用する。  
この関数で取得したポインタは呼び出し元で解放すること。  
解放する際は、「表 4.5.4-1の No.7」を呼び出すこと。

#### 4.7.5.7. GetBracketsData([OUT] CCSPFM\_ItemInfoArray&)

(1) 関数概要

関数名	引数	戻り値
GetBracketsData()	(OUT) CCSPFM_ItemInfoArray&	void

(2) パラメータ

CCSPFM\_ItemInfoArray& : (OUT)複数データ(山括弧区切り)の配列

(3) 戻り値

なし

(4) 機能

指定した項目に記述されているデータを複数型(山括弧区切り)として取得する。

(5) API 使用上の注意事項

CheckDataType()を実行した結果、データ種別が「表 4.4.3-1の No.4」の場合に本関数を使用する。  
この関数で取得したポインタは呼び出し元で解放すること。  
解放する際は、「表 4.5.4-1の No.7」を呼び出すこと。

#### 4.7.5.8. GetRefPart()

(1) 関数概要

関数名	引数	戻り値
GetRefPart()	—	CCSPFM_PartInfo*

(2) パラメータ

なし

(3) 戻り値

CCSPFM\_PartInfo\* : 参照先のパートへのポインタ

(4) 機能

参照先のパートを取得する。

(5) API 使用上の注意事項

CheckDataType()を実行した結果、データ種別が「表 4.4.3-1の No.7」の場合に本関数を使用する。

#### 4.7.5.9. GetRawData()

(1) 関数概要

関数名	引数	戻り値
GetRawData()	—	CString

(2) パラメータ

なし

(3) 戻り値

CString : 取得した文字列型データ

(4) 機能

記述されているデータを記述記法に考慮せず、文字列として取得する。

(5) API 使用上の注意事項

特になし

#### 4.7.5.10. GetDataTypeInfo([IN]BOOL bBitSize = FALSE)

##### (1) 関数概要

関数名	引数	戻り値
GetDataTypeInfo()	(IN) BOOL bBitSize = FALSE	DWORD

##### (2) パラメータ

BOOL : (IN)ビットサイズかどうかのフラグ(デフォルト : FALSE)  
TRUE : ビットサイズ指定で取得する  
FALSE : ワードサイズ指定で取得する

##### (3) 戻り値

DWORD : DATATYPE 項目のデータサイズ

##### (4) 機能

DATATYPE 項目のデータサイズを取得する。

IN 引数が TRUE の場合 : ビットサイズでデータサイズを取得する。

IN 引数が FALSE の場合 : ワードサイズでデータサイズを取得する。

##### (5) API 使用上の注意事項

特になし

#### 4.7.5.11. GetSize([IN]CString&)

(1) 関数概要

関数名	引数	戻り値
GetSize()	(IN) CString&	DWORD

(2) パラメータ

CString : (IN)指定されたデータ型

(3) 戻り値

DWORD : 指定されたデータ型のビットサイズ

(4) 機能

指定されたデータ型のビットサイズを取得する。

(5) API 使用上の注意事項

特になし



#### 4.7.6. CCSPFM\_ExceptionBase クラス

##### 4.7.6.1. GetErrCode()

###### (1) 関数概要

関数名	引数	戻り値
GetErrCode()	—	DWORD

###### (2) パラメータ

なし

###### (3) 戻り値

DWORD : エラーコード

###### (4) 機能

発生したエラーコードを取得する。

取得できるエラーコードは、「表 4.4.4-1」を参照。

###### (5) API 使用上の注意事項

特になし

## 5. 装置用 CSP+パーサの使用サンプル

Microsoft Visual Studio2010 にて、装置用 CSP+パーサを使用するサンプルを以下に示します。

使用側のソースコード例①)

指定の項目の記述内容を取得する場合のサンプルを以下に示します。

```
void CTagSampleDlg::OnBnClickedButton1()
{
    CString strFilePath = GetFilePath();
    ClearOutputMessage();
    //[初期化]
    CCSPFM_ProfileData* pcCSPFM_ProfileData = NULL;
    try{

        //[対象の装置用CSP+ファイルを読み込む]
        pcCSPFM_ProfileData = new CCSPFM_ProfileData(strFilePath);

        CCSPFM_SectionInfoArray cSectionInfoArray;
        CCSPFM_SectionInfo* pcSectionInfo;
        //[セクション種別を指定して、COMM_IFセクションを取得する]
        pcCSPFM_ProfileData->Get(ECSPFMDEF_SECTIONTYPE_COMMIF, cSectionInfoArray);
        CCSPFM_PartInfo* pcPartInfo = NULL;
        CCSPFM_ElemInfo* pcElemInfo = NULL;
        CCSPFM_ItemInfo* pcItemInfo = NULL;

        for(INT iSectionInfo = 0; iSectionInfo < cSectionInfoArray.GetCount(); iSectionInfo++)
        {

            //[COMM_IFセクションを取得する]
            pcSectionInfo = cSectionInfoArray.GetAt(iSectionInfo);
            //[パート種別を指定して、COMM_IF_INFOパートを取得する]
            CCSPFM_PartInfoArray pcPartInfoArray;
            pcSectionInfo->Get(ECSPFMDEF_PARTTYPE_COMMIF_INFO, pcPartInfoArray);
            OUTPUT_MESSAGE(_T("Section:%s¥r¥n"), pcSectionInfo->m_strLabel);

            if(0 < pcPartInfoArray.GetCount())
            {

                //[COMMIF_INFOパートを取得する]
                pcPartInfo = pcPartInfoArray.GetAt(0);
                if(NULL == pcPartInfo)
                {
                    continue;
                }
                //[要素「VendorName」を取得する]
                pcElemInfo = pcPartInfo->Get(_T("VendorName"));
                if(NULL == pcElemInfo)
                {
                    continue;
                }
                //[項目「DATA」を取得する]
                pcItemInfo = pcElemInfo->Get(ECSPFMDEF_ITEMTYPE_DATA);
                if(NULL == pcItemInfo)
                {
                    continue;
                }
                //[項目の記述内容を取得する]
                CString strData = pcItemInfo->GetRawData();
                OUTPUT_MESSAGE(_T("¥tVendorName:%s¥r¥n"), strData);
            }
        }
    }
}
```

```

        }
    }
    //[生成したクラスを削除]
    if (NULL != pcCSPFM_ProfileData)
    {
        delete pcCSPFM_ProfileData;
        pcCSPFM_ProfileData = NULL;
    }
}
catch(CCSPFM_ExceptionBase& e)
{
    //[生成したクラスを削除]
    if (NULL != pcCSPFM_ProfileData)
    {
        delete pcCSPFM_ProfileData;
        pcCSPFM_ProfileData = NULL;
    }
    CString strMsg;
    strMsg.Format(_T("Error : 0x%08x"),
        e.GetErrCode());
    MessageBox(strMsg);
}
}

```

## 使用側のソースコード例②)

特定のセクション、パートを検索し、参照先パート情報を取得する場合のサンプルを以下に示します。

```
void CTagSampleDlg::OnBnClickedButton1()
{
    CString strFilePath = GetFilePath();
    ClearOutputMessage();

    //[初期化]
    CCSPFM_ProfileData* pcCSPFM_ProfileData = NULL;
    try
    {
        //[対象の装置用CSP+ファイルを読み込む]
        pcCSPFM_ProfileData = new CCSPFM_ProfileData(strFilePath);
        //[LABEL名「CommIfSec_MC」のセクションを取得する。]
        CCSPFM_SectionInfo* pcCommIfSection = pcCSPFM_ProfileData->Get(_T("CommIfSec_MC"));
        if(NULL == pcCommIfSection)
        {
            //[生成したクラスを削除]
            RELEASE_MEM(pcCSPFM_ProfileData)
            return;
        }

        //[LABEL名「CommIfVar」のパートを取得する]
        CCSPFM_PartInfo* pcCommandPart = pcCommIfSection->Get(_T("CommIfVar"));
        if(NULL == pcCommandPart)
        {
            //[生成したクラスを削除]
            RELEASE_MEM(pcCSPFM_ProfileData)
            return;
        }

        //[LABEL名「P_GetStatusVar」の要素を取得する]
        CCSPFM_ElemInfo* pcElement = pcCommandPart->Get(_T("P_GetStatusVar"));
        if(NULL == pcElement)
        {
            //[生成したクラスを削除]
            RELEASE_MEM(pcCSPFM_ProfileData)
            return;
        }

        //[項目「DATA」を取得する]
        CCSPFM_ItemInfo* pcItemInfo = pcElement->Get(ECSPFMDEF_ITEMTYPE_DATA);
        if(NULL == pcItemInfo)
        {
            //[生成したクラスを削除]
            RELEASE_MEM(pcCSPFM_ProfileData)
            return;
        }

        //[「DATA」項目には参照できるデータであるかチェックする]
        if(CSPFMDEF_DATATYPE_REF_PART != pcItemInfo->CheckDataType())
        {
            //[生成したクラスを削除]
            RELEASE_MEM(pcCSPFM_ProfileData)
            return;
        }
    }
}
```

```

        // [参照パートを取得する]
        CCSPFM_PartInfo* pcCommandRefPart = pcItemInfo->GetRefPart();
        if(NULL == pcCommandRefPart)
        {
            // [生成したクラスを削除]
            RELEASE_MEM(pcCSPFM_ProfileData)
            return;
        }
        // [生成したクラスを削除]
        RELEASE_MEM(pcCSPFM_ProfileData)
    }
    catch(CCSPFM_ExceptionBase& e)
    {
        // [生成したクラスを削除]
        RELEASE_MEM(pcCSPFM_ProfileData)

        CString strMsg;
        strMsg.Format(_T("Error : 0x%08x"),
            e.GetErrCode());
        MessageBox(strMsg);
        return;
    }
}

```

#### 6. 商標、登録商標などについて

Microsoft, Windows, Windows Vista, Windows XP, Windows7, Windows10, Excel, Visual Studio は、米国 Microsoft Corporation の米国およびその他の国における商標または登録商標です。

本文中における会社名、システム名、製品名などは、一般に各社の登録商標または商標です。

本文中で、商標記号(™ , ®)は明記していない場合があります。

装置用 *CSP+*

2018年7月作成