RADIOCOMMUNICATION BUREAU	EPFDMASKSAPI Library	**Ref** : EPFDMASKSAPI-IE-UM
		Version : 4.2
		Author : V. PHILAVONG
	Dept./Div. : IAP/SAS	**Date** : 2023-08-24

EPFDMASKSAPI Library

User Manual

Abstract

- Contains the description of the library's functions you can use to programmatically read the EPFD_Masks.

Participants

- SAS division

Point of contact

- Virasouk.philavong@itu.int

CHAPTER 1 INTRODUCTION 5

CHAPTER 2 DATA TYPES AND STRUCTURES 7

CHAPTER 3 FUNCTIONS DIRECTORY 10

MASKS DATA FUNCTIONS 11

EPFD_Masks_Extract 11

EPFD_Masks_Count 11

EPFD_Masks_List 12

EPFD_Masks_List2 13

CHAPTER 4 USING EPFDMASKSAPI 15

4.1 ADVANCED TOPICS 15

4.2 SAMPLES 15

DOCUMENT CHANGE RECORD

Issue	Date	Change Description
V1	2007-03-08	Template release – V. Philavong
V2	2011-08-09	Added ***mask id*** as input parameter of function EPFD_Masks_Extract
V2.2	2011-11-24	Added new functions: **EPFD_Masks_Count** and **EPFD_Masks_List**
V2.3	2011-12-12	Added new function: **EPFD_Masks_List2**
V4.0	2016-11-30	Modified *epfd_st.h* + Recompiled with VS2010
V4.1	2016-12-19	Modified *epfd_st.h:* increase the size of the sat_name field (WRC15 SRSV8)
V4.2	2023-08-24	64-bit version release

Chapter 1
INTRODUCTION

Chapter 1 INTRODUCTION

The **EPFDMASKSAPI** library will give your applications the possibility to access and extract data stored in the EPFD_Masks. It does that through an interface defined by a set of C-style functions that are not supposed to change regardless of the format of the database accessed.

This manual gives application developers general as well as detailed information about **EPFDMASKSAPI** functions and data types.

Therefore, it is divided into three main parts:

Chapter 2. **DATA TYPES AND STRUCTURES**

Chapter 3. **FUNCTIONS DIRECTORY**

Chapter 4. **USING EPFDMASKSAPI**

Chapter 2
DATA TYPES AND STRUCTURES

Chapter 2 DATA TYPES AND STRUCTURES

The prototypes of functions are defined in the file **EpfdMasksApi.h**, distributed with the library. Their contents are listed below.

```
#ifdef __cplusplus
extern "C"
{
#endif

#include "Epfd_st.h"


// The following ifdef block is the standard way of creating macros which make exporting
// from a DLL simpler. All files within this DLL are compiled with the EPFDMASKSAPI_EXPORTS
// symbol defined on the command line. this symbol should not be defined on any project
// that uses this DLL. This way any other project whose source files include this file see
// EPFDMASKSAPI_API functions as being imported from a DLL, wheras this DLL sees symbols
// defined with this macro as being exported.
#ifdef EPFDMASKSAPI_EXPORTS
#define EPFDMASKSAPI_API __declspec(dllexport)
#else
#define EPFDMASKSAPI_API __declspec(dllimport)
#endif

////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////MASK////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////
/*!
 *      \brief  Extract XML masks file from Masks database.
 */
EPFDMASKSAPI_API stEpfdStatus WINAPI EPFD_Masks_Extract(unsigned long inNoticeID,

                                                        unsigned long inMaskID,

                                                        const char* inMasksDBPath,

                                                        const char* inMasksFilePath);


/*!
 *      \brief  Count the number of masks of a given notice
 */
EPFDMASKSAPI_API stEpfdStatus WINAPI EPFD_Masks_Count(unsigned long inNoticeID,

                                                       const char* inMasksDBPath,

                                                       int* outNumMasks);



/*!
 *      \brief  get the list of masks and mask type flags of a given notice
 */
EPFDMASKSAPI_API stEpfdStatus WINAPI EPFD_Masks_List(unsigned long inNoticeID,

                                                      const char* inMasksDBPath,

                                                      unsigned long *outMaskIDList,

                                                      char *outF_mask);
```

```
/*!
 *      \brief  get the list of masks and mask type flags of a given notice
 */
EPFDMASKSAPI_API stEpfdStatus WINAPI EPFD_Masks_List2(unsigned long inNoticeID,
                                                      const char* inMasksDBPath,
                                                      unsigned long *outMaskIDList,
                                                      char *outF_mask,
                                                      char *outF_mask_type);


#ifdef __cplusplus
}
#endif
```

The file **Epfd_st.h** is also necessary when one uses EpfdMasksApi library in user program implementation because it contains the definitions of constants and types used.

Chapter 3
FUNCTIONS DIRECTORY

Chapter 3 FUNCTIONS DIRECTORY

This chapter contains the list of functions from the **EPFDMASKSAPI** that are available to the application developer.

It includes functions allowing to extract data stored in EPFD_Masks.mdb database. The names of functions are begining with **EPFD_Masks_** and following by a description of what they use for. For example: **EPFD_Masks_Extract**

The general scheme for the presentation is as follows:

Function: **[Function Name]**

Syntax: **[Function Syntax]**

Purpose: **[Function's Purpose]**

Parameters: **[Description Of The Function's Input Parameters]**

Return value: **[Description Of The Value Returned By The Function]**

Remarks: **[Additional Information]**

See Also: **[Cross Reference To Other Related Items]**

MASKS DATA FUNCTIONS

Function: **EPFD_Masks_Extract**

Syntax: **stEpfdStatus EPFD_Masks_Extract (**unsigned long *inNoticeID*,
unsigned long *inMaskID*,
const char* *inMasksDBPath*,
const char* *inMasksFilePath***)**;

Purpose: This function allows to extract an XML masks file from EPFD_Masks Database.

Parameters:

inNoticeID	unsigned long	Identifier of the satellite system notice.
inMaskID	unsigned long	Identifier of the mask
inMasksDBPath	const char*	The path of the Masks Database where the XML file masks of is stored.
inMasksFilePath	const char*	The path where the XML file must be extracted to.

Return value: On success, this function returns a value 0 that means the masks file is extracted and written on the XML file specified in parameter *inMasksFilePath*. It returns a warning value (=1), if the given notice is not exist in the database. Otherwise, it returns an error value (<0) if it is a Database access error.

Function: **EPFD_Masks_Count**

Syntax: **stEpfdStatus EPFD_Masks_Count (**unsigned long *inNoticeID*,
const char* *inMasksDBPath,,*
int* *outNumMasks***)**;

Purpose: This function allows to extract an XML masks file from EPFD_Masks Database.

Parameters:

inNoticeID	unsigned long	Identifier of the satellite system notice.
inMasksDBPath	const char*	The path of the Masks Database where the XML file masks of is stored.
outNumMasks	int*	The number of masks PFD and EIRP of the given notice.

Return value: On success, this function returns a value 0. Otherwise, it returns an error value (<0) if it is a Database access error.

Function: **EPFD_Masks_List**

Syntax: **stEpfdStatus EPFD_Masks_List (**unsigned long *inNoticeID*,
const char* *inMasksDBPath*,
unsigned long **outMaskIDList*,
char **outF_mask***)**;

Purpose: This function allows to extract an XML masks file from EPFD_Masks Database.

Parameters:

inNoticeID	unsigned long	Identifier of the satellite system notice.
inMasksDBPath	const char*	The path of the Masks Database where the XML file masks of is stored.
**outMaskIDList*	unsigned long	The list of Mask ID.
**outF_mask*	char	The list of flag indicating if the mask type is eirp at the space station [S], eirp at the associated earth station [E] or pfd at the space station [P].

Return value: On success, this function returns a value 0. Otherwise, it returns an error value (<0) if it is a Database access error.

Remarks: The User is responsible for defining the size of **outMaskIDList* and **outF_mask*. The size must be equal, at least, to the number of Mask ID of the notice plus one (see example below).

Example

```
#include <iostream.h>
#include "EpfdMasksApi.h"

int Main()
{
	unsigned long ntc_id = 96520515;
	const char* PathToMasksDb = "C:\\projects\\EPFD\\EPFD_masks.mdb";

	// get the number of masks of the given notice
	int numMasks ;
	int status = EPFD_Masks_Count(ntc_id,  PathToMasksDb, &numMasks);

	if (numMasks > 0)
	{
		 // get the list of mask ID and
		unsigned long *MaskID_List = new unsigned long[numMasks+1] ;
		char *f_mask_List = new char[numMasks+1] ;
		status = EPFD_Masks_List(ntc_id,  PathToMasksDb, MaskID_List, f_mask_List) ;

		for (int i=0; i<numMasks; i++)
		{
			cout << MaskID_List[i] << "\n";
			cout << f_mask_List[i] << "\n";
		}

		delete[] MaskID_List;
		delete[] f_mask_List;
	}
	return status ;
}
```

Function: **EPFD_Masks_List2**

Syntax: **stEpfdStatus EPFD_Masks_List2 (**unsigned long *inNoticeID*,
const char* *inMasksDBPath*,
unsigned long **outMaskIDList*,
char **outF_mask,*
char **outF_mask_type***);

Purpose: This function allows to extract an XML masks file from EPFD_Masks Database.

Parameters:

	inNoticeID	unsigned long Identifier of the satellite system notice.
	inMasksDBPath	const char* The path of the Masks Database where the XML file masks of is stored.
	**outMaskIDList*	unsigned long The list of Mask ID.
	**outF_mask*	char The list of flag indicating if the mask type is eirp at the space station [S], eirp at the associated earth station [E] or pfd at the space station [P].
	**outF_mask_type*	char The list of flag indicating the PFD mask option: [A] for 'alpha_deltaLongitude' ; [X] for 'X_deltaLongitude' ; [Z] for 'azimuth_elevation'. For the mask type eirp F_mask_type is the same as F_mask i.e. [E] for eirp at the associated earth station and [S] for eirp at the space station.

Return value: On success, this function returns a value 0. Otherwise, it returns an error value (<0) if it is a Database access error.

Remarks: The User is responsible for defining the size of **outMaskIDList*, **outF_mask and* **outF_mask_type*. The size must be equal, at least, to the number of Mask ID of the notice plus one (see example below).

Chapter 4
Using EPFDMASKSAPI

Chapter 4 Using EPFDMASKSAPI

4.1 ADVANCED TOPICS

This section deals with a number of issues (calling conventions, passing parameters, ...) that the programmer needs to know in order to properly use **EPFDMASKSAPI**, especially when he uses another programming language than **C** to interface the main program with **EPFDMASKSAPI**.

- In its current version, **EPFDMASKSAPI** was developed using **Microsoft Visual C++ 6.0™** and its target operating environment is Windows NT 4.0, 2000, XP as well as Windows 9x.
- When you are using the **EPFDMASKSAPI** library from a C/C++ program you should *#include* the header file **Epfd_st.h** and **EpfdMasksApi.h**. It contains the declaration of all functions, constants and return codes.
- Regarding **Calling Conventions**, it should to be noted that all **EPFDMASKSAPI** functions are declared to be **WINAPI** (which is actually a *#define* for **__stdcall**).
- To run an application using the **EPFDMASKSAPI** library you will also need some other DLLs. These DLLs will be provided in a EPFD Package.

4.2 SAMPLES

The **masks.cpp** file contains a small C++ console application showing the use of **EPFD_Masks_Extract** function of the **EPFDMASKSAPI** library.