


# Open Mobile API Specification

Errata for Version 3.1

Published by  simalliance now Trusted Connectivity Alliance

July 2016

**Copyright © 2016 Trusted Connectivity Alliance Ltd.**

The information contained in this document may be used, disclosed and reproduced without the prior written authorization of Trusted Connectivity Alliance. Readers are advised that Trusted Connectivity Alliance reserves the right to amend and update this document without prior notice. Ownership of the OMAPI Specification has been transferred to GlobalPlatform. All future releases will be available on the GlobalPlatform website.

# Table of Contents

- 1. Introduction ..... 3
- 2. List of modifications ..... 3
  - 2.1 Chapter 6.2.2 e) Method int getVersion() ..... 3
  - 2.2 Chapter 6.3.2 a) OMAPI RESULT omapi\_get\_version ..... 3
  - 2.3 Annex A: Ansi-C Reference Header for Transport Procedural Interface ..... 4

## 1. Introduction

This document contains errata notes for Open Mobile API Specification v3.1. This document uses revision marks to show the new applicable content. The document contains only those parts of the subchapters where the errata are made. All the other parts of the Open Mobile API Specification v3.1 remain unchanged and applicable.

## 2. List of modifications

### 2.1 Chapter 6.2.2 e) Method `int getVersion()`

*Rationale of the errata: In order to maintain compatibility with 3.0 and 2.05 versions of OMAPI specification the Method `int getVersion()` is removed and replaced by `String getVersion()`.*

#### (e) Method: ~~`int String getVersion()`~~

Returns the version of the Open Mobile API Specification this implementation is based on.

#### Return value:

~~The version number is returned as an integer value, with the precise type used being one which is natural to the OS platform (with the proviso of a minimum of 32 bits precision).~~

~~The overall version number is calculated based on a major version number and minor version number. The major version number is a value between 2 and 2147483 and the minor number is a value between 0 and 999. The calculation is:~~

~~$version\_number = (major\_version\_number * 1000) + minor\_version\_number$  (e.g. "3001" for the implementation which is based on the OMAPI specification v3.1).~~

~~String containing the Open Mobile API version (e.g. "3.1" for Open Mobile API Specification version 3.1).~~

### 2.2 Chapter 6.3.2 a) OMAPI RESULT `omapi_get_version`

*Rationale of the errata: same as for 2.1 in procedural interface.*

#### (a) OMAPI RESULT `omapi_get_version(String pVersion, Int *pLengthInt *pVersion)`

~~`pVersion` has to be a properly allocated integer.~~

~~`pVersion` has to be a properly allocated string or null if the function should return the proper string length in `pLength`.~~

~~Depending on the software platform, `pVersion` might include a `\0` after the version name if the platform representation of strings is zero-terminated.~~

#### Parameters

~~[out] `Int *pVersion` – allocated integer where the version number will be stored.~~

[out] String pVersion - allocated string buffer that will contain the version string or null to determine the proper length in pLength.

[in/out] Int \*pLength - size of version string length.

### Return

Success - all went ok.

IllegalParameterError - pLength is too short for the version string to be returned.

GeneralError - general error not further specified.

## 2.3 Annex A: Ansi-C Reference Header for Transport Procedural Interface

*Rationale of the errata : align the reference header to the modification made in this document.*

```

/* omapi.h
 * Copyright (c) 20146 SIMalliance.org*/
#ifdef __omapi_h__
#define __omapi_h__

#ifdef __cplusplus
extern "C" {
#endif

/* platform specific mapping of SIMalliance data types */
#ifdef OMAPI_API
#define OMAPI_API
#endif
typedef int OMAPI_RESULT;
typedef int OMAPI_HANDLE;
typedef int Int;
typedef char * String;
typedef unsigned char Byte;
typedef enum { false, true } Boolean;

/* SIMalliance return codes */
#define OMAPI_SUCCESS ((Int)0x00000000) /* No
error was encountered */
#define OMAPI_GENERAL_ERROR ((Int)0x10000000) /* A
general error occurred */
#define OMAPI_IO_ERROR ((Int)0x10000001) /*
Communication error */
#define OMAPI_NO_SUCH_ELEMENT_ERROR ((Int)0x10000002) /* No
such element error */
#define OMAPI_ILLEGAL_STATE_ERROR ((Int)0x10000003) /*
Illegal state of execution error */

```

```

#define OMAPI_ILLEGAL_PARAMETER_ERROR ((Int)0x10000004) /*
Illegal or invalid parameter */
#define OMAPI_ILLEGAL_REFERENCE_ERROR ((Int)0x10000005) /*
Illegal reference */
#define OMAPI_OPERATION_NOT_SUPPORTED_ERROR ((Int)0x10000006) /*
Operation not supported from SE */
#define OMAPI_SECURITY_ERROR ((Int)0x10000007) /*
Security Error blocks execution */
#define OMAPI_CHANNEL_NOT_AVAILABLE_ERROR ((Int)0x10000008) /* No
channel available */
#define OMAPI_NULL_POINTER_ERROR ((Int)0x10000009) /* Null
pointer not allowed */

/* SIMalliance Open Mobile API */
OMAPI_API OMAPI_RESULT omapi_get_readers(OMAPI_HANDLE *phReaders, Int
*pLength);
OMAPI_API OMAPI_RESULT omapi_get_version(String pVersion, Int *pLength, Int
*pVersion);

OMAPI_API OMAPI_RESULT omapi_reader_get_name(OMAPI_HANDLE hReader, String
pReader, Int *pLength);
OMAPI_API OMAPI_RESULT omapi_reader_is_secure_element_present(OMAPI_HANDLE
hReader, Boolean *pIsPresent);
OMAPI_API OMAPI_RESULT omapi_reader_open_session(OMAPI_HANDLE hReader,
OMAPI_HANDLE *phSession);
OMAPI_API OMAPI_RESULT omapi_reader_close_sessions(OMAPI_HANDLE hReader);

OMAPI_API OMAPI_RESULT omapi_session_get_reader(OMAPI_HANDLE hSession,
OMAPI_HANDLE *phReader);
OMAPI_API OMAPI_RESULT omapi_session_get_atr(OMAPI_HANDLE hSession, Byte
*pAtr, Int *pLength);
OMAPI_API OMAPI_RESULT omapi_session_close(OMAPI_HANDLE hSession);
OMAPI_API OMAPI_RESULT omapi_session_is_closed(OMAPI_HANDLE hSession,
Boolean *pIsClosed);
OMAPI_API OMAPI_RESULT omapi_session_close_channels(OMAPI_HANDLE hSession);
OMAPI_API OMAPI_RESULT omapi_session_open_basic_channel(OMAPI_HANDLE
hSession, Byte *AID, Int length, Byte P2, OMAPI_HANDLE *phChannel);
OMAPI_API OMAPI_RESULT omapi_session_open_logical_channel(OMAPI_HANDLE
hSession, Byte *AID, Int length, Byte P2, OMAPI_HANDLE *phChannel);

OMAPI_API OMAPI_RESULT omapi_channel_close(OMAPI_HANDLE hChannel);
OMAPI_API OMAPI_RESULT omapi_channel_is_basic_channel(OMAPI_HANDLE
hChannel, Boolean *pIsBasicChannel);
OMAPI_API OMAPI_RESULT omapi_channel_is_closed(OMAPI_HANDLE hChannel,
Boolean *pIsClosed);
OMAPI_API OMAPI_RESULT omapi_channel_get_select_response(OMAPI_HANDLE
hChannel, Byte *pSelectResponse, Int *pLength);
OMAPI_API OMAPI_RESULT omapi_channel_get_session(OMAPI_HANDLE hChannel,
OMAPI_HANDLE *phSession);
OMAPI_API OMAPI_RESULT omapi_channel_transmit(OMAPI_HANDLE hChannel, Byte
*pCommand, Int cmdLength, Byte *pResponse, Int *pRspLength);

```

```
OMAPI_API OMAPI_RESULT  
omapi_channel_transmit_retrieve_response(OMAPI_HANDLE hChannel, Byte  
*pResponse, Int *pRspLength);  
OMAPI_API OMAPI_RESULT omapi_channel_select_next(OMAPI_HANDLE hChannel,  
Boolean *pSuccess);  
  
#ifdef __cplusplus  
}  
#endif  
  
#endif
```