Winwap Technologies Oy

WAP Stack utility library

Application Programming Interface

WAP stack utility version 1.x
WAP specification version: 1.2.1
Document dated: 5 Jan 2005

Notice of Confidentiality

Revision history

Date	Author	Description
16-Jan-2003	M Shkirja	Initial versions of document
19-Mar-2003	S Markelov	Updated 1.1: Description of wbxml2wml 2.2
25-Nov-2004	Maria Sandell	English spell checked.

Preamble

The reader of this document should be familiar with all or some of the following in order to fully understand and evaluate the information in this document:

- Basic knowledge in programming techniques.
- The interaction between a WAP user-agent, a WAP Gateway and a WEB Server.

Contents

1 Normative references

RFC-3629	"UTF-8, a transformation format of ISO 10646", ftp://ftp.isi.edu/in-notes/rfc2616.txt.
WML	"Wireless Markup Language Specification", http://www.openmobilealliance.org/.
WBXML	"Binary XML Content Format Specification", http://www.openmobilealliance.org/.

2 API specification

2.1 Declarations

All functional and type declarations are available in the following C-header file:

wps_utils.h — API functions not belonging to data transfer operations.

2.2 Decoding WBXML to WML

NAME

wbxml2wml — decode binary encoded WML.

SYNOPSIS

```
#include "wps_utils.h"

RET_CODE wbxml2wml(const void *in, size_t in_size,
        unsigned char *out, size_t *out_size,
        unsigned int *charset);
```

DESCRIPTION

The wbxml2wml function is used to decode a buffer with binary encoded WML content.

The function parameters are:

in	Pointer to the first byte of binary WML content.
in_size	Size in bytes of the binary WML content.
out	Output buffer for decoded (plain text) WML content.
out_size	Size in bytes of the output buffer. Returns the size of decoded (plain text) WML content.
charset	Output parameter for the character set code of decoded (plain text) WML content.

RETURN VALUE

The function returns the size in bytes of the decoded (plain text) WML content in the parameter out_size. If the output buffer contains enough space, a trailing 0 follows the last character of the output data. The return error code values are listed below.

ERRORS

RC_OK — Successfully decoded.

RC_EOF — Attempt to read past the end of the input stream.

RC_NOMEM — Not enough memory for internal data buffers.

RC_WRONG_WBXML_VER — Unexpected WBXML document version.

RC_WRONG_PUBLICID — Unexpected XML document public identifier.

RC_NOSTRTERM — Missing terminator char in string table.

RC_OUTOFTABLE — Invalid string table index

RC_WRONG_TAG — Unexpected tag token encountered

RC_WRONG_ATTRSTART — Unexpected ATTRSTART code for current CP.

RC_WRONG_ATTRVALUE — Unexpected ATTRVALUE code for current CP.

RC_WRONG_ INPUT — Invalid input character.

RC_UNKNOWN_OR_MISSING_PUBLICID — Unknown or missing public identifier.

2.3 Character conversion

NAME

convert — convert data of any character set to data of USC-2 character set.

SYNOPSIS

```
#include "wps_utils.h"

size_t convert(const unsigned char *in, size_t in_size,
        t_charset cs, wchar_t *out, size_t out_size);
```

DESCRIPTION

The convert function is used to convert data of any character set to data of USC-2 (two-byte unicode) character set.

The function parameters are:

in	Pointer to the input buffer with characters to convert.
in_size	Size in bytes of the input buffer.
cs	Character set code of characters to convert.
out	Pointer to output buffer for UCS-2 characters.
charset	Size in wchar_t of the output buffer (count of UCS-2 characters in the output buffer).

RETURN VALUE

The function returns a count of wchar_t characters in the out buffer. In order to get the needed size in wchar_t for the out buffer, a 0 must be passed in out_size. The character set codes are defined as constants as shown below:

CS_ISO_8859_1	— ISO-8859-1 (latin1)
CS_ISO_8859_2	— ISO-8859-2 (latin2)
CS_ISO_8859_3	— ISO-8859-3 (latin3)
CS_ISO_8859_4	— ISO-8859-4 (latin4)
CS_ISO_8859_5	— ISO-8859-5 (Cyrillic)
CS_ISO_8859_6	— ISO-8859-6 (Arabic)
CS_ISO_8859_7	— ISO-8859-7 (Greek)
CS_ISO_8859_8	— ISO-8859-8 (Hebrew)
CS_ISO_8859_9	— ISO-8859-9 (latin5)
CS_ISO_8859_10	— ISO-8859-10 (latin6)
CS_ISO_8859_13	— ISO-8859-13
CS_ISO_8859_14	— ISO-8859-14 (latin8)
CS_ISO_8859_15	— ISO-8859-15 (Latin-9)
CS_SHIFT_JIS	— SHIFT_JIS
CS_UTF_8	— UTF-8

Index