

**TPJSONTUUBF(3)**

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

# Contents

<a href="#">1</a>	<a href="#">SYNOPSIS</a>	<a href="#">1</a>
<a href="#">2</a>	<a href="#">DESCRIPTION</a>	<a href="#">2</a>
<a href="#">3</a>	<a href="#">RETURN VALUE</a>	<a href="#">3</a>
<a href="#">4</a>	<a href="#">ERRORS</a>	<a href="#">4</a>
<a href="#">5</a>	<a href="#">EXAMPLE</a>	<a href="#">5</a>
<a href="#">6</a>	<a href="#">BUGS</a>	<a href="#">6</a>
<a href="#">7</a>	<a href="#">SEE ALSO</a>	<a href="#">7</a>
<a href="#">8</a>	<a href="#">COPYING</a>	<a href="#">8</a>

## Chapter 1

# SYNOPSIS

```
#include <atmi.h>
```

```
int tpjsontoubf(UBFH *p_ub, char *buffer);
```

For XATMI client link with *-latmiclt -latmi -lubf -lnstd -lpthread -lrt -lm*

For XATMI server link with *-latmisrvl -latmisrvnomainl -latmisrvinteg -latmi -lubf -lnstd -lpthread -lrt -lm*

---

## Chapter 2

# DESCRIPTION

Function converts JSON formatted text in *buffer* to UBF buffer. JSON should be formatted in single level. Where JSON field name matches UBF field name. If multiple occurrences/array of values is present in JSON, then it is converted as multiple field occurrences in UBF buffer.

For example following JSON buffer:

```
{
  "T_SHORT_FLD":1765,
  "T_LONG_FLD":[
    3333111,
    2
  ],
  "T_CHAR_FLD":"A",
  "T_FLOAT_FLD":1.330000,
  "T_DOUBLE_FLD":[
    1111.220000,
    333,
    444
  ],
  "T_STRING_FLD":"HELLO WORLD",
  "T_CARRAY_FLD":"AAECA0hFTE×PIEJJTkFSWQQFAA=="
}
```

Will be converted to following UBF buffer:

T_SHORT_FLD	1765
T_LONG_FLD	3333111
T_LONG_FLD	2
T_CHAR_FLD	A
T_FLOAT_FLD	1.33000
T_DOUBLE_FLD	1111.220000
T_DOUBLE_FLD	333.000000
T_DOUBLE_FLD	444.000000
T_STRING_FLD	HELLO WORLD
T_CARRAY_FLD	\00\01\02\03HELLO BINARY\04\05\00

Note that **BFLD\_CARRAY** (binary data) fields should be present in base64 encoding in JSON buffer. It is up to user to allocate UBF buffer with enough size to fit the converted message.

## Chapter 3

# RETURN VALUE

On success, **tpcall()** return zero; on error, -1 is returned, with **tperrno** set to indicate the error.

## Chapter 4

# ERRORS

Note that **tpstrerror()** returns generic error message plus custom message with debug info from last function call.

**TPEINVAL** Invalid JSON or invalid base64 encoding,

**TPESYSTEM** UBF sub-system error, JSON sub-system error.

**TPEOS** System failure occurred during serving. See logs i.e. user log, or debugs for more info. In case of insufficient memory this error will be generated too.

---

## Chapter 5

# EXAMPLE

See `atmitest/test024_json/atmict24.c` for sample code.



## Chapter 6

# BUGS

Report bugs to [support@mavimax.com](mailto:support@mavimax.com)

## Chapter 7

## SEE ALSO

**ndrxdebug.conf(5), tpubftojson(3) tpjsontoview(3) tpviewtojson(3)\***

## Chapter 8

# COPYING

© Mavimax, Ltd