

**TPVIEWTOJSON(3)**

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>SYNOPSIS</b>	<b>1</b>
<b>2</b>	<b>DESCRIPTION</b>	<b>2</b>
<b>3</b>	<b>RETURN VALUE</b>	<b>4</b>
<b>4</b>	<b>ERRORS</b>	<b>5</b>
<b>5</b>	<b>EXAMPLE</b>	<b>6</b>
<b>6</b>	<b>BUGS</b>	<b>7</b>
<b>7</b>	<b>SEE ALSO</b>	<b>8</b>
<b>8</b>	<b>COPYING</b>	<b>9</b>

## Chapter 1

# SYNOPSIS

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

```
int tpviewtojson(char *cstruct, char *view, char *buffer, int bufsize, long flags);
```

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

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

## Chapter 2

# DESCRIPTION

Function converts VIEW buffer in *cstruct*, described by *view* to JSON string stored in *buffer* string. The field *bufsize* must indicate the space in **buffer**. If the size is not enough, function will fail. *flags* current is used to define the function logic in case for *NULL* fields in view file.

For example, following buffer in *cstruct/view*:

```
VIEW MYVIEW2
#type      cname      fbname      count    flag      size      null
short      tshort1    -          1        -         -         2000
long       tlong1     -          1        -         -         5
char       tchar1     -          1        -         -         '\0'
float      tfloat1    -          1        -         -         1.1
double     tdouble1   -          1        -         -         0.0
string     tstring1   -          3        -         15        'HELLO WORLD'
carray     tcarray1   -          1        -         10        '- '
END
```

and structure initialized to (in C, no error handling for simplicity):

```
struct MYVIEW2 v2;
char buf_with_nulls[65000];
char buf_no_nulls[65000];

Bvsinit((char *)&v2, "MYVIEW2"); /* initialize to NULL values */

v2.tshort1=100;
strcpy(v2.tstring1[1], "Enduro/X");

/* by default null fields are exported */
tpviewtojson((char *)&v2, "MYVIEW2", buf_with_nulls, sizeof(buf_with_nulls), 0);

/* BVACCESS_NOTNULL makes to export only non NULL fields*/
tpviewtojson((char *)&v2, "MYVIEW2", buf_no_nulls, sizeof(buf_with_nulls), BVACCESS_NOTNULL ←
);
```

Buffer *buf\_with\_nulls* in this example will be filled with following message:

```
{
  "MYVIEW2":
  {
    "tshort1":100,
    "tlong1":5,
```

```
        "tchar1": "",
        "tfloat1": 1.10000,
        "tdouble1": 0.000000,
        "tstring1": ["HELLO WORLD", "Enduro/X", "HELLO WORLD"],
        "tcarray1": "LQ=="
    }
}
```

Buffer *buf\_no\_nulls* in this example will be filled with following message:

```
{
    "MYVIEW2":
    {
        "tshort1": 100,
        "tstring1": ["HELLO WORLD", "Enduro/X"]
    }
}
```

NOTE that "carray" typed view fields are exported as base64 strings.

## Chapter 3

# RETURN VALUE

On success, **tpacall()** return call descriptor (>0); 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.

**TPESYSTEM** System failure occurred during serving. See logs i.e. user log, or debugs for more info. Might be insufficient memory, JSON or UBF library failure. Enduro/X environment not configured.

**TPEOS** Not enough space in output *buffer*.

**TPEINVAL** *cstruct* is NULL, *view* is NULL or empty string, *buffer* is NULL.



## Chapter 5

# EXAMPLE

See `atmitest/test040_typedview/atmict40.c` for sample code.

## Chapter 6

# BUGS

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

## Chapter 7

## SEE ALSO

`ndrxdebug.conf(5)` `tpjsontoview(3)` `viewc(8)` `ex_env(5)`

## Chapter 8

# COPYING

© Mavimax, Ltd