

**foptest-fop-2044**

**preview\_20140117\_182057**

## **fop-test-fop-2044**

### **Copyright**

This is just a test of FOP hyphenation 2014.

## Table of Contents

Overview .....	iv
I - Xxxxx .....	5
<b>1 - fop_test.h</b> .....	6
1.1 - EpmStatus .....	7
1.2 - EthDestConn .....	7
1.3 - HvGuestResult .....	8
1.4 - HvGuestXx .....	9
1.5 - ProfilingPerfCounterStatus .....	9
1.6 - SCTP_PARTIAL_DELIVERY_ABORTED .....	10
1.7 - SCTP_PARTIAL_DELIVERY_EVENT .....	10
1.8 - Testaddress .....	10
1.9 - XSE_BSP_EKRN_EVT_MISALIGNED_MSG_QUEUE .....	10
1.10 - ip6_hdr .....	10
1.11 - sctp_pdapi_event .....	11
<b>Index</b> .....	12

# Overview

Below is an overview of the documented files.

**Table 1 Header Files**

<a href="#">fop_test.h</a>	Special test files for testing FOP hyphenation
----------------------------	--

## **Part I. Xxxxx**

# 1. fop\_test.h

**Description** Special test files for testing FOP hyphenation

There are now only very few words that FOP can not hyphenate nicely but those cause overflow in this case where all XML files are autogenerated.

The PDF created from this is FOP trunk dated Jan 2, 2014. The only fop.conf is <source-resolution>120</source-resolution> and <target-resolution>120</target-resolution> which should not affect the result.

Command `fop --noconfig -Djava.awt.headless=true -server -pdf xxxxx.fo`

XSE\_BSP\_EKRN\_EVT\_MISALIGNED\_MSG\_QUEUE gives WARNING but is very close to the column right edge, very slightly overflowing. See Table 1.3 below.

`ip6_ctlun.ip6_un1.ip6_un1_flow` overflows "ow". See [ip6\\_hdr](#).

`ip6_ctlun.ip6_un1.ip6_un1_plen` overflows "en". See [ip6\\_hdr](#).

`ip6_ctlun.ip6_un1.ip6_un1_nxt` overflows "t". See [ip6\\_hdr](#).

EPM\_REQUEST\_TO\_CLEAR\_NONEXISTING\_BP overflows tailing "P" even though it is hyphenated "EPM\_RE-" much too early. See [EpmStatus](#).

PROFILING\_PC\_STATUS\_NO\_COUNTER\_AVAILABLE\_FOR\_TYPE overflows even though it breaks in one place "PROFILING\_PC\_S-". See [ProfilingPerfCounterStatus](#).

LINUX\_CM\_ETH\_EDESTROY\_IN\_PROGRESS overflows and "also" has a hyphenation break far out inside overflow. See [EthDestConn](#).

HV\_GUEST\_RESULT\_EGUEST\_NOT\_INITIALIZED overflows "NITIALIZED". See [HvGuestResult](#).

HV\_GUEST\_TRANSITION\_GUEST\_USER\_MODE overflows "ER\_MODE". See [HvGuestXx](#).

HV\_GUEST\_TRANSITION\_GUEST\_SU\_MODE overflows "MODE". See [HvGuestXx](#).

`ff01:0000:0000:0000:0000:0000:0101:0000:0010:0001` overflows. See [Testaddress](#).

**Table 1.1 Macros**

<a href="#">SCTP_PARTIAL_DELIVERY_ABORTED</a>	Partial delivery was aborted.
<a href="#">SCTP_PARTIAL_DELIVERY_EVENT</a>	This notification is used to tell a receiver that the partial delivery has been aborted.

**Table 1.2 Types**

<a href="#">EpmStatus</a>	enum EpmStatus defines the various return values from EPM functions.
<a href="#">EthDestConn</a>	XXXXXXXXXXXXXXXXXX
<a href="#">HvGuestResult</a>	Reactions from Guest on external stimuli.

<a href="#">HvGuestXx</a>	Specifies the mode in which the guest should be running.
<a href="#">ProfilingPerfCounterStatus</a>	Enumeration of status values that may be returned.
<a href="#">Testaddress</a>	Testing ipv6 address hyphenation but because the column width in this case is larger than when the address is used in another book, we make it longer than normal to force hyphenation or rather overflow because FOP does not break it.
<a href="#">ip6_hdr</a>	IPv6 header structure.
<a href="#">sctp_pdapi_event</a>	Xxxxxx

**Table 1.3 Numbers**

<a href="#">XSE_BSP_EKRN_EVT_MISALIGNED_MSG_QUEUE</a>	Disadvantaged size of the message queue.
---	--

## 1.1 EpmStatus

### Declaration

```
typedef enum EpmStatus
{
    EPM_OK = 0,
    EPM_UNKNOWN_FLAGS = 1,
    EPM_INVALID_ARGUMENTS = 2,
    EPM_INVALID_INSTRUCTION = 3,
    EPM_FUNCTION_NOT_IMPLEMENTED = 4,
    EPM_INTERNAL_ERROR = 5,
    EPM_NO_UNUSED_HW_BREAKPOINTS = 6,
    EPM_REQUEST_TO_CLEAR_NONEXISTING_BP = 7,
    EPM_INIT_FAILED = 8,
    EPM_TOO_MANY_BP = 9
} EpmStatus;
```

### Description

enum EpmStatus defines the various return values from EPM functions.

### Fields

EPM_OK	Operation was successful.
EPM_UNKNOWN_FLAGS	Use of unknown flag in the flag field.
EPM_INVALID_ARGUMENTS	Bad argument passed to an EPM function.
EPM_INVALID_INSTRUCTION	It is not possible to set an execution point at specified address.
EPM_FUNCTION_NOT_IMPLEMENTED	The requested function is not implemented.
EPM_INTERNAL_ERROR	Internal error. See RAMLOG for further information.
EPM_NO_UNUSED_HW_BREAKPOINTS	Failure to set a breakpoint because no unused breakpoint registers was found.
EPM_REQUEST_TO_CLEAR_NONEXISTING_BP	Attempt to remove a breakpoint that has not been set.
EPM_INIT_FAILED	Initiation of breakpoint failed. Proper breakpoint functionality is not guaranteed.
EPM_TOO_MANY_BP	Too many breakpoints have set.

## 1.2 EthDestConn

### Declaration

```
struct EthDestConn
```

```
{
    SIGSELECT signo;
    U32 status;
    struct LinxConnObj *co;
};
```

<b>Description</b>	XXXXXXXXXXXXXXXXXX	
	XXXXXX XXXXX XXXXXXXXXXXX XXXXXXXXXXXX XXXXXXXXXXXX XXXXXXX.	
<b>Input</b>	signo	LINX_ETH_DESTROY_CONN_REQUEST LINX_ETH_DESTROY_CONN_REPLY
	co	Handle to the connection. Use the handle that is returned in LINX_ETH_CREATE_CONN_REPLY signal from XXXX.
<b>Output</b>	status	Returns the status.
	LINX_CM_ETH_SUCCESS	Connection destroyed.
	LINX_CM_ETH_INVALID_CONN_OBJ	Invalid connection object.
	LINX_CM_ETH_DESTROY_IN_PROGRESS	Destroying in progress.

### 1.3 HvGuestResult

```
enum HvGuestResult
{
    HV_GUEST_RESULT_ESUCCESS = 0,
    HV_GUEST_RESULT_EGUEST_NOT_INITIALIZED = 1,
    HV_GUEST_RESULT_EGUEST_ALREADY_INITIALIZED = 2,
    HV_GUEST_RESULT_EUNKNOWN_GUEST = 3,
    HV_GUEST_RESULT_EUNKNOWN_STATE = 4,
    HV_GUEST_RESULT_EUNKNOWN_EVENT = 5,
    HV_GUEST_RESULT_EILLEGAL_TRANSITION = 6,
    HV_GUEST_RESULT_ENOT_BOOTED = 7,
    HV_GUEST_RESULT_EALREADY_HAS_HANDLER = 8
};
```

<b>Description</b>	Reactions from Guest on external stimuli.	
<b>Input</b>	HV_GUEST_RESULT_ESUCCESS	Success.
	HV_GUEST_RESULT_EGUEST_NOT_INITIALIZED	Attempt to re-initialize guest.
	HV_GUEST_RESULT_EGUEST_ALREADY_INITIALIZED	Attempt to re-initialize guest.
	HV_GUEST_RESULT_EUNKNOWN_GUEST	The guest type is unknown.
	HV_GUEST_RESULT_EUNKNOWN_STATE	The state is unknown.
	HV_GUEST_RESULT_EUNKNOWN_EVENT	Attempt to trigger unknown event.
	HV_GUEST_RESULT_EILLEGAL_TRANSITION	Attempt to perform illegal state-transition on guest.
	HV_GUEST_RESULT_ENOT_BOOTED	N/A



HV\_GUEST\_RESULT\_EAL-                    Attempt to re-install handler.  
 READY\_HAS\_HANDLER

## 1.4 HvGuestXx

### Declaration

```
enum HvGuestXx
{
    HV_GUEST_TRANSITION_HYPERVISOR_USER_MODE = 1,
    HV_GUEST_TRANSITION_HYPERVISOR_SU_MODE = 2,
    HV_GUEST_TRANSITION_GUEST_USER_MODE = 3,
    HV_GUEST_TRANSITION_GUEST_SU_MODE = 4
};
```

### Description

Specifies the mode in which the guest should be running.

### Fields

HV\_GUEST\_TRANSITION\_HYPERVISOR\_USER\_MODE    The guest should start in hypervisor user mode.

HV\_GUEST\_TRANSITION\_HYPERVISOR\_SU\_MODE    The guest should start in hypervisor supervisor mode.

HV\_GUEST\_TRANSITION\_GUEST\_USER\_MODE    The guest should start in guest user mode.

HV\_GUEST\_TRANSITION\_GUEST\_SU\_MODE    The guest should start in guest supervisor mode.

## 1.5 ProfilingPerfCounterStatus

### Declaration

```
typedef enum ProfilingPerfCounterStatus {
    PROFILING_PC_STATUS_OK = 0,
    PROFILING_PC_STATUS_TYPE_NOT_COUNTED,
    PROFILING_PC_STATUS_TYPE_ALREADY_COUNTED,
    PROFILING_PC_STATUS_NO_COUNTER_AVAILABLE_FOR_TYPE,
    PROFILING_PC_STATUS_NO_BUFFERS,
    PROFILING_PC_STATUS_INVALID_ARGUMENT
} ProfilingPerfCounterStatus;
```

### Description

Enumeration of status values that may be returned.

All status values, except for PROFILING\_PC\_STATUS\_OK, should be considered to be error codes.

### Fields

PROFILING\_PC\_STATUS\_OK                    Status OK.

PROFILING\_PC\_STATUS\_TYPE\_NOT\_COUNTED    The specified counter event type is currently not being counted.

PROFILING\_PC\_STATUS\_TYPE\_ALREADY\_COUNTED    The specified counter event type is already being counted.

PROFILING\_PC\_STATUS\_NO\_COUNTER\_AVAILABLE\_FOR\_TYPE    The specified counter event type cannot be counted as there is currently no counter available that can accommodate it.

PROFILING\_PC\_STATUS\_NO\_BUFFERS            the specified counter event type cannot be started due to the lack of sample buffers (see profiling\_add\_sample\_buffer)

PROFILING\_PC\_STATUS\_INVALID\_ARGUMENT    the user called the driver API function with invalid arguments

## 1.6 SCTP\_PARTIAL\_DELIVERY\_ABORTED

**Declaration** `#define SCTP_PARTIAL_DELIVERY_ABORTED`

**Description** Partial delivery was aborted.

## 1.7 SCTP\_PARTIAL\_DELIVERY\_EVENT

**Declaration** `#define SCTP_PARTIAL_DELIVERY_EVENT`

**Description** This notification is used to tell a receiver that the partial delivery has been aborted.

**See Also** `sctp_notification`

## 1.8 Testaddress

**Declaration**

```
enum Testaddress
{
    XYZ = 1,
    ABC = 2
};
```

**Description** Testing ipv6 address hyphenation but because the column width in this case is larger than when the address is used in another book, we make it longer than normal to force hyphenation or rather overflow because FOP does not break it.

<b>Fields</b>	XYZ	This IPv6 address ff01:0000:0000:0000:0000:0000:0101:0000:0010:0001 is probably too long
	ABC	xxxx

## 1.9 XSE\_BSP\_EKRN\_EVT\_MISALIGNED\_MSG\_QUEUE

**Declaration** `#define XSE_BSP_EKRN_EVT_MISALIGNED_MSG_QUEUE (0x00880006)`

**Description** Disadvantaged size of the message queue.

## 1.10 ip6\_hdr

**Declaration**

```
struct ip6_hdr
{
    union
    {
        struct ip6_hdrctl
        {
            uint32_t ip6_unl_flow;
            uint16_t ip6_unl_plen;
            uint8_t ip6_unl_nxt;
            uint8_t ip6_unl_hlim;
        } ip6_unl;
        uint8_t ip6_un2_vfc;
    } ip6_ctlun;
    struct in6_addr ip6_src;
    struct in6_addr ip6_dst;
};
```

```
};
```

<b>Description</b>	IPv6 header structure.	
<b>Fields</b>	<code>ip6_ctlun</code>	Union with type specific data.
	<code>ip6_ctlun.ip6_un1</code>	Type specific data with a struct.
	<code>ip6_ctlun.ip6_un1.ip6_un1_flow</code>	4 bits version, 8 bits TC, 20 bits flow ID.
	<code>ip6_ctlun.ip6_un1.ip6_un1_plen</code>	Payload length.
	<code>ip6_ctlun.ip6_un1.ip6_un1_nxt</code>	Next header.
	<code>ip6_ctlun.ip6_un1.ip6_un1_hop</code>	Hop limit.
	<code>lim</code>	
	<code>ip6_ctlun.ip6_un2_vfc</code>	Type specific data. 4 bits version, top 4 bits tclass.
	<code>ip6_src</code>	Source address.
	<code>ip6_dst</code>	Destination address.

## 1.11 sctp\_pdapi\_event

**Declaration**

```
struct sctp_pdapi_event
{
    uint16_t pdapi_type;
    uint32_t pdapi_indication;
    uint16_t pdapi_stream;
};
```

**Description** Xxxxxx

**Fields**

<code>pdapi_type</code>	Should be <a href="#">SCTP_PARTIAL_DELIVERY_EVENT</a> .
<code>pdapi_indication</code>	Holds the indication being sent to the application. Currently only <a href="#">SCTP_PARTIAL_DELIVERY_ABORTED</a> indication is defined.
<code>pdapi_stream</code>	The stream on which the partial delivery event happened.

**See Also** [sctp\\_notification](#)

## Index

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [K](#) [L](#) [M](#)  
[N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [X](#) [Y](#) [Z](#)

### E

EpmStatus, [7](#)  
EthDestConn, [7](#)

### F

fop\_test.h, [6](#), [6](#)

### H

HvGuestResult, [8](#)  
HvGuestXx, [9](#)

### I

ip6\_hdr, [10](#)

### P

ProfilingPerfCounterStatus, [9](#)

### S

SCTP\_PARTIAL\_DELIVERY\_ABORTED, [10](#)  
SCTP\_PARTIAL\_DELIVERY\_EVENT, [10](#)  
sctp\_pdapi\_event, [11](#)

### T

Testaddress, [10](#)

### X

XSE\_BSP\_EKRN\_EVT\_MISALIGNED\_MSG\_QUEUE,  
[10](#)