Interface Management API Implementation Agreement (Packet Over SONET Interfaces)

Revision 3.0

Editor: John Renwick, Agere Systems, jrenwick@agere.com

The words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", “MAY", and "OPTIONAL" in the remainder of this document are to be interpreted as described in the NPF Software API Conventions Implementation Agreement revision 1.0.

For additional information contact:
The Network Processing Forum, 39355 California Street,
Suite 307, Fremont, CA 94538
+1 510 608-5990 phone info@npforum.org

Table of Contents

Table of Figures

1 Revision History

Revision	Date	Reason for Changes
3.0	11/22/2004	Extracted POS interface features from Interface Management API Implementation Agreement; made editorial changes, and technical changes as needed for independence of Interface Management header files.

2 Introduction

This document defines the Packet Over SONET (POS) interface type under the NPF Interface Management API.

2.1 Assumptions and External Requirements

1. This API assumes the existence of the Interface Management API Core Function Set, and shares all the same assumptions and external requirements of that API.

2.2 Scope

This document is concerned only with definitions and functions supporting POS interfaces under NPF Interface Management.

2.3 Dependencies

This API shares the same dependences as the Interface Management API Core Function Set.

2.3.1 Packet Over Sonet (POS) Interface

- This interface type represents a point-to-point SONET link carrying datagrams encapsulated in PPP protocol. It currently has no attributes; instead it inherits those of a SONET interface, of which it is a child.

3 Data Types

3.1 POS Interface Data Types

3.1.1 Interface Type Code (NPF_IF_TYPE_POS)

The type code for POS interfaces is 4 (this is a restatement of the existing definition in the Interface Management API Implementation Agreement, Core Function Set).

```
#define NPF_IF_TYPE_POS       4           /* POS Interface */
```

3.1.2 Packet Over SONET (POS) Interface Attributes: NPF_IfPOS_t

```
/*
 *    Packet over SONET (POS) Interface Attributes
 */
typedef struct NPF_IfPOS {
      /* Structure is currently empty */
};
```

3.2 Data Structures for Completion Callbacks

3.2.1 Completion Callback Type Codes: NPF_IfCallbackType_t

These codes are used as values of the NPF_IfCallbackType_t variable in callbacks, to indicate what function is returning information. No codes for POS are defined at present.

```
/*
 *    Completion Callback Types for POS interfaces
 */
```

3.2.2 Asynchronous Response Details

Asynchronous responses are returned by the mechanism defined in Interface Management API Core Function Set. The NPF_IfAsyncResponse_t structure defined there contains a (void *) pointer; in the case of POS API function calls, this pointer can point to a structure containing POS-specific or generic information. The following table indicates what is returned by each of the functions defined in this Implementation Agreement. (No POS-specific functions are currently defined.)

Function Name	Type Code	Structure Returned

3.3 Error Codes

The following codes are used as values of **`NPF_IfErrorType_t`**.

```
/*
 *    Asynchronous error codes (returned in function callbacks)
```

```
 *    used by POS interface functions
 */

#define NPF_IF_E_POS_CODE(code) (0x10000+(NPF_IF_TYPE_POS<<8)+(code))
```

3.4 Data Structures for Event Notifications

3.4.1 Event Types: NPF_IfEvent_t

```
/*
 *    Event types
 */
```

4 Functions

4.1 Completion Callback

No completion callback function is defined in this document. All interface type-specific functions use the callback definitions of the Core Interface Management API.

4.2 Event Notification

No event-related functions are defined in this document. All interface type-specific functions use the event function definitions of the Core Interface Management API.

4.3 Interface Management API

This section will define functions for querying and modifying the interface properties and attributes.

No functions for Packet Over Sonet interfaces are currently defined.

5 References

1 NP Forum – Software API Conventions Implementation Agreement Revision 2.0.

6 API Capabilities

This section defines the capabilities of the Interface Management API.

It summarizes the defined APIs and Events and defines the mandatory and optional features.

6.1 Optional support of specific types

The support of any specific type of interface is optional in an implementation. An implementation MAY support exclusively one type of interface, and still claim compliance to the NP Forum Interface Management API.

6.2 API Functions

Function Name	Required?

6.3 API Events

Event Name	Required?

APPENDIX A HEADER FILE: NPF_IF_POS.H

```
/*
 * This header file defines typedefs, constants, and functions
 * that apply to the NPF Interface Management API, support for
 * Packet Over Sonet (POS) interface type.
 */
#ifndef __NPF_IF_POS_H__
#define __NPF_IF_POS_H__

#ifdef __cplusplus
extern "C"  {
#endif

#define NPF_IF_TYPE_POS 4     /* Packet over SONET interface */


/*
 *    Packet over SONET (POS) Interface Attributes
 */
struct NPF_IfPOS {
      /* Structure is currently empty */
};

#ifdef __cplusplus
}
#endif

#endif
```

APPENDIX B LIST OF COMPANIES BELONGING TO NPF DURING APPROVAL PROCESS

Agere Systems
Alcatel
Altera
AMCC
Analog Devices
Avici Systems
Azanda Network Devices
Cypress Semiconductor
Ericsson
Erlang Technologies
EZ Chip
Flextronics
Fujitsu Ltd.
FutureSoft
HCL Technologies
Hi/fn
IBM
IDT
Intel
IP Infusion
Kawasaki LSI
LSI Logic
Modelware
Mosaid
Motorola
NEC
NetLogic
Nokia
Paion Co., Ltd.
PMC Sierra
RadiSys
Samsung Electronics
Sandburst Corporation
Silicon & Software Systems
Silicon Access
Sony Electronics
STMicroelectronics
Sun Microsystems
Teja Technologies
TranSwitch
U4EA Group
Xelerated
Xilinx
Zettacom
ZTE