# Wide Area Network Interface Specification

# Wide Area Network Interface Specification

Version 1.1 Edition 7.20141001
Updated October 25, 2014
Distributed with Package openss7-1.1.7.20141001



## Abstract:

This document is a Specification containing technical details concerning the implementation of the Wide Area Network Interface for OpenSS7. It contains recommendations on software architecture as well as platform and system applicability of the Wide Area Network Interface. It provides abstraction of the HDLC Frame Protocol (ISO 3309) to these components as well as providing a basis for link layer control for other network protocols.

**Brian Bidulock** <bidulock@openss7.org> **for**
**The OpenSS7 Project** <http://www.openss7.org/>

## Published by:

OpenSS7 Corporation
1469 Jefferys Crescent
Edmonton, Alberta T6L 6T1
Canada



## Notice:

## Short Contents

# Table of Contents

## List of Figures

## List of Tables

# Preface

## Notice

Software in this document and related software is released under the AGPL (see [GNU Affero General Public License], page 79). Please note, however, that there are different licensing terms for some of the manual package and some of the documentation. Consult permission notices contained in the documentation of those components for more information.

This document is released under the FDL (see [GNU Free Documentation License], page 89) with no invariant sections, no front-cover texts and no back-cover texts.

## Abstract

This document is a Specification containing technical details concerning the implementation of the Wide Area Network Interface for OpenSS7. It contains recommendations on software architecture as well as platform and system applicability of the Wide Area Network Interface.

This document specifies a Wide Area Network Interface Specification in support of the OpenSS7 Media Access Control Protocol (MAC) protocol stacks. It provides abstraction of the Media Access Control interface to these components as well as providing a basis for Media Access Control control for other Media Access Control protocols.

### Purpose

The purpose of this document is to provide technical documentation of the Wide Area Network Interface. This document is intended to be included with the OpenSS7 STREAMS software package released by *OpenSS7 Corporation*. It is intended to assist software developers, maintainers and users of the Wide Area Network Interface with understanding the software architecture and technical interfaces that are made available in the software package.

### Intent

It is the intent of this document that it act as the primary source of information concerning the Wide Area Network Interface. This document is intended to provide information for writers of OpenSS7 Wide Area Network Interface applications as well as writers of OpenSS7 Wide Area Network Interface Users.

### Audience

The audience for this document is software developers, maintainers and users and integrators of the Wide Area Network Interface. The target audience is developers and users of the OpenSS7 SS7 stack.

## Revision History

Take care that you are working with a current version of this documentation: you will not be notified of updates. To ensure that you are working with a current version, check the OpenSS7 Project website for a current version.

A current version of this specification is normally distributed with the *OpenSS7* package, `openss7-1.1.7.20141001`.[1]

[1] `http://www.openss7.org/repos/tarballs/openss7-1.1.7.20141001.tar.bz2`

### Version Control

Although the author has attempted to ensure that the information in this document is complete and correct, neither the Author nor OpenSS7 Corporation will take any responsibility in it. *OpenSS7 Corporation* is making this documentation available as a reference point for the industry. While *OpenSS7 Corporation* believes that these interfaces are well defined in this release of the document, minor changes may be made prior to products conforming to the interfaces being made available. *OpenSS7 Corporation* reserves the right to revise this software and documentation for any reason, including but not limited to, conformity with standards promulgated by various agencies, utilization of advances in the state of the technical arts, or the reflection of changes in the design of any techniques, or procedures embodied, described, or referred to herein. *OpenSS7 Corporation* is under no obligation to provide any feature listed herein.

```
$Log: wan.texi,v $
Revision 1.1.2.2  2011-02-07 02:21:48  brian
- updated manuals

Revision 1.1.2.1  2009-06-21 10:57:42  brian
- added files to new distro
```

## ISO 9000 Compliance

Only the TEX, texinfo, or roff source for this maual is controlled. An opaque (printed, postscript or portable document format) version of this manual is a **UNCONTROLLED VERSION**.

### Disclaimer

*OpenSS7 Corporation* **disclaims all warranties with regard to this documentation including all implied warranties of merchantability, fitness for a particular purpose, non-infrincement, or title; that the contents of the manual are suitable for any purpose, or that the implementation of such contents will not infringe on any third party patents, copyrights, trademarks or other rights. In no event shall** *OpenSS7 Corporation* **be liable for any direct, indirect, special or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action or contract, negligence or other tortious action, arising out of or in connection with any use of this documentation or the performance or implementation of the contents thereof.**

### U.S. Government Restricted Rights

If you are licensing this Software on behalf of the U.S. Government ("Government"), the following provisions apply to you. If the Software is supplied by the Department of Defense ("DoD"), it is classified as "Commercial Computer Software" under paragraph 252.227-7014 of the DoD Supplement to the Federal Aquisition Regulations ("DFARS") (or any successor regulations) and the Government is acquiring only the license rights granded herein (the license rights customarily provided to non-Government users). If the Software is supplied to any unit or agency of the Government other than DoD, it is classified as "Restricted Computer Software" and the Government's rights in the Software are defined in paragraph 52.227-19 of the Federal Acquisition Regulations ("FAR") (or any successor regulations) or, in the cases of NASA, in paragraph 18.52.227-86 of the NASA Supplerment to the FAR (or any successor regulations).

## Acknowledgements

The OpenSS7 Project was funded in part by:

- Monavacon Limited
- OpenSS7 Corporation

Thanks to the subscribers to and sponsors of The OpenSS7 Project. Without their support, open software like this would not be possible.

As with most open source projects, this project would not have been possible without the valiant efforts and productive software of the Free Software Foundation, the Linux Kernel Community, and the open source software movement at large.

# 1 Introduction

The Wide Area Network (WAN) Interface was developed by Spider Systems, Ltd., (now a division of Emerson Power) and is widely available on many platforms. For example, *AIX AIXlink/X.25*, *HP-UX HP X.25/9000*, *Solaris Solstice X.25* and *SunLink X.25*, *IRIX IRIS SX.25*, *PT X.25*, *RadiSys WAN* and *SBE X.25* implement the Wide Area Newtork (WAN) Interface.

The Wide Area Network (WAN) Inteface was designed to be used directly with standard STREAMS system calls and does not require the use of a cooperating user space shared library. Applications program directly use the `getmsg(2s)`, `getpmsg(2s)`, `putmsg(2s)` and `putpmsg(2s)` system calls.[1] Nevertheless, user shared object libraries can easily be constructed using this STREAMS service primitive interface.

The system header files that must be included when compiling user applications, or STREAMS drivers and modules that use the interface, are detailed in Appendix A [WAN Header Files], page 49.

A user library, `libcdiapi`, is provided, not for interfacing to the message primitive service interface, but for providing various helper functions when using the STREAMS service interface. This library is detailed in ⟨undefined⟩ [⟨undefined⟩], page ⟨undefined⟩.

[1] See `getmsg(2s)`, `getpmsg(2s)`, `putmsg(2s)` and `putpmsg(2s)` manual pages.

# 2 Model of the WAN Layer

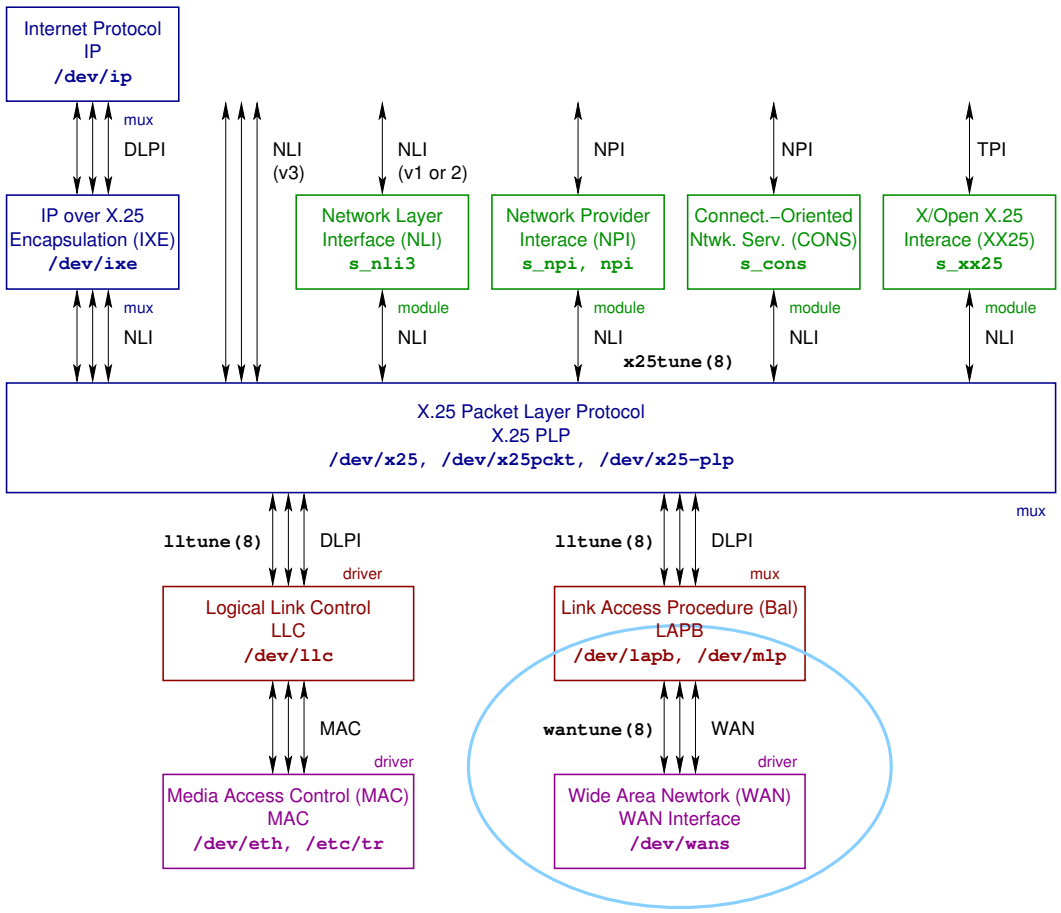


Figure 2.1: *X.25 Overview*

# 3 WAN Services

## 3.1 WAN Commands

## 3.2 WAN Data Structures

# 4 WAN Message Primitives

The `/usr/include/openss7/sys/snet/wan_proto.h` header file (`<sys/snet/wan_proto.h>` with proper compile flags) contains definitions and declarations of primitive structures and field values.

The `WAN_primitives` union is formatted as follows:

```
union WAN_primitives {
    uint8_t wan_type;
    struct wan_sid wsid;
    struct wan_reg wreg;
    struct wan_ctl wctl;
    struct wan_msg wmsg;
    struct wan_nty wnty;
};
```

The `WAN_primitives` union contains the following members:

*wan_type* Specifies the type of the structure contained in the `M_PROTO` message block. Always one of the following values:

| | |
|---|---|
| `WAN_SID` | The contained structure is a `wan_sid` structure. |
| `WAN_REG` | The contained structure is a `wan_reg` structure. |
| `WAN_CTL` | The contained structure is a `wan_ctl` structure. |
| `WAN_DAT` | The contained structure is a `wan_msg` structure. |
| `WAN_NTY` | The contained structure is a `wan_nty` structure. |

*wsid* The structure of the primitive when *wan_type* is `WAN_SID`.
See Section 4.1 [WAN˙SID - Set Subnetwork Identifier], page 14.

*wreg* The structure of the primitive when *wan_type* is `WAN_REG`.
See Section 4.2 [WAN˙REG - Register Subnetwork Identifier], page 15.

*wctl* The structure of the primitive when *wan_type* is `WAN_CTL`.
See Section 4.3 [WAN˙CTL - Control], page 16.

*wmsg* The structure of the primitive when *wan_type* is `WAN_DAT`.
See Section 4.4 [WAN˙DAT - Data], page 19.

*wnty* The structure of the primitive when *wan_type* is `WAN_NTY`.
See Section 4.5 [WAN˙NTY - Notify], page 21.

These primitive types are described in detail in the sections that follow.

## 4.1 WAN_SID - Set Subnetwork Identifier

Requests that the WAN driver assign the specified subnetwork identifier to the Stream and associate the Stream with the underlying device identified by the subnetwork identifier. This primitive is equivalent to the CD_ATTACH_REQ(7) primitive of the cdi(7).

### Format

The primitive consists of one M_PROTO message block containing a wan_sid structure. The wan_sid structure is formatted as follows:

```
struct wan_sid {
    uint8_t wan_type;
    uint8_t wan_spare[3];
    uint32_t wan_snid;
};
```

### Parameters

The wan_sid structure contains the following members:

*wan_type* Specifies the type of the structure contained in the M_PROTO message block. Always WAN_SID.

*wan_spare* Spare bytes for alignment: ignored by the responder and set to zero by the initiator.

*wan_snid* Conveys the subnetwork identifier. Equivalent to the Physical Point of Attachment (PPA) of the WAN interface.

### Response

Should an error occur, an M_ERROR message is sent upstream with an appropriate error code, resulting in the failure of all system calls on the Stream until closed. The WAN_SID primitive may fail when:

[ENODEV] The specified *wan_snid* is unknown or invalid.

[EINVAL] The M_PROTO message block is of an incorrect length for the primitive.

[EEXIST] The *wan_snid* specified is in use by another Stream.

[ERANGE] The *wan_snid* member contains invalid information.

[EBUSY] The physical channel referenced by the *wan_snid* is in use by another Stream.

[ENOSR] The s_wan module, or underlying CDI driver lacks the STREAMS resources necessary to satisfy the request.

[EIO] The WAN_SID primitive was issued from an incorrect state for the subnetwork entity.

[ENXIO] The underlying device has encountered a fatal error.

## 4.2 WAN_REG - Register Subnetwork Identifier

Registers the subnetwork identifier specified in the *wan_snid* field of the primitive. Once a subnetwork entity has been registered, it cannot be tuned or otherwise altered unless disabled with `W_DISABLE`.[1] This primitive is used by management applications to place subnetwork entities into service.

### Format

The primitive consists of one `M_PROTO` message block containing a `wan_reg` structure. The `wan_reg` structure is formatted as follows:

```
struct wan_reg {
    uint8_t wan_type;
    uint8_t wan_spare[3];
    uint32_t wan_snid;
};
```

### Parameters

The `wan_reg` structure contains the following members:

*wan_type* Specifies the type of the structure contained in the `M_PROTO` message block. Always `WAN_REG`.

*wan_spare* Spare bytes for alignment: ignored by the responder and set to zero by the initiator.

*wan_snid* Conveys the subnetwork identifier. Equivalent to the Physical Point of Attachment (PPA) of the WAN interface.

### Response

Should an error occur, an `M_ERROR` message is sent upstream with an appropriate error code, resulting in the failure of all system calls on the Stream until closed. The `WAN_REG` primitive may fail when:

`[ENODEV]` The specified *wan_snid* is unknown or invalid.

`[EINVAL]` The `M_PROTO` message block is of an incorrect length for the primitive.

`[EXDEV]` There exists a hardware configuration error for the specified *wan_snid*.

`[EBUSY]` The physical channel referenced by the *wan_snid* is in use by another Stream.

`[EIO]` The `WAN_REG` primitive was issued from an incorrect state for the subnetwork entity.

`[ENXIO]` The underlying device has encountered a fatal error.

`[ENOMEM]` There was insufficient memory immediately available to register the subnetwork identity.

`[E2BIG]` The maximum receive buffer size is too small to hold the largest frame required by the device.

[1] See Section 5.2.10 [W˙DISABLE - Disable Interface Data Transfer], page 46.

## 4.3 WAN_CTL - Control

This primitive class consists of four primitives used to enable or disable the WAN interface. These primitives are equivalent to the CD_ENABLE_REQ(7), CD_ENABLE_CON(7), CD_DISABLE_REQ(7), CD_DISABLE_CON(7), and the CD_ERROR_IND(7) primitives of the cdi(7).

### Format

The WAN_CTL primitives consist of one M_PROTO message block containing a wan_ctl structure. The wan_ctl structure is formatted as follows:

```
struct wan_ctl {
    uint8_t wan_type;
    uint8_t wan_command;
    uint8_t wan_remtype;
    uint8_t wan_remsize;
    uint8_t wan_remaddr[20];
    uint8_t wan_status;
    uint8_t wan_diag;
};
```

### Usage

The WC_CONNECT command is valid from the WAN user to the WAN driver; or from the WAN driver to the WAN user. Fields *wan_remtype*, *wan_remsize* and *wan_remaddr* are significant, *wan_status* and *wan_diag* are ignored.

This WC_CONCNF command is valid when sent from the WAN driver to the WAN user; or from the WAN user to the WAN driver. Fields *wan_status* and *wan_diag* are significant, *wan_remtype*, *wan_remsize* and *wan_remaddr* are ignored.

This WC_DISC command is valid when sent from the WAN user to the WAN driver; or from the WAN driver to the WAN user. All fields are ignored.

This WC_DISCCNF command is valid when sent from the WAN driver to the WAN user; or from the WAN user to the WAN driver. Fields *wan_status* and *wan_diag* are significant, *wan_remtype*, *wan_remsize* and *wan_remaddr* are ignored.

### Parameters

The wan_ctl structure contains the following members:

*wan_type* Specifies the type of the structure contained in the M_PROTO message block. Always WAN_CTL.

*wan_command*
Conveys the WAN command. This field may assume one of the following values:

| | |
|---|---|
| WC_CONNECT | The primitive is a connect request or indication. |
| WC_CONCNF | The primitive is a connect confirmation. |
| WC_DISC | The primitive is a disconnect request of indication. |
| WC_DISCCNF | The primitive is a disconnect confirmation. |

*wan_remtype*
When significant, conveys the type of the remote address. This field may have one of the following values:

`WAN_TYPE_ASC`
: The *wan_remaddr* field contains ASCII coded digits. The *wan_remsize* field contains the number of digits (in octets).

`WAN_TYPE_BCD`
: The *wan_remaddr* field contains BCD encoded digits. The *wan_remsize* field contains the number of digits (in semi-octets).

This field is only significant in the `WC_CONNECT` primitive, and for devices that have call procedural definitions. Otherwise, the field is set to zero (0).

*wan_remsize*
: When significant, conveys the length of the remote address in digits (either octets or semi-octets depending on the *wan_remtype* member).

This field is only significant in the `WC_CONNECT` primitive, and for devices that have call procedural definitions. Otherwise, the field is set to zero (0).

*wan_remaddr*
: When significant, conveys the remote address. The address contianed in this field is either represented as ASCII digits or BCD encoded digits, depending on the value of the *wan_remtype* field. This field is only significant in the `WC_CONNECT` primitive.

This field is only significant in the `WC_CONNECT` primitive, and for devices that have call procedural definitions. Otherwise, the field is set to null.

*wan_status* When significant, provides the status for the `WC_CONCNF` or `WC_DISCCNF` command. This field can assume one of the following values:

`WAN_FAIL` The preceding `WC_CONNECT` or `WC_DISC` command was unsuccessful. The link remains in the disconnected or connected state as the case may be.

`WAN_SUCCESS`
: The preceding `WC_CONNECT` or `WC_DISC` command was successful. The link moves to the connected or disconnected state as the case may be.

This field is only significant in the `WC_CONCNF` and `WC_DISC` primitives.

*wan_diag* When significant and the status field is `WAN_FAIL`, provides diagnostic information concerning the failure. This field is only significant in the `WC_CONCNF` and `WC_DISCCNF` primitives when failure is indicated.

### State

The `WC_DISCCNF` command is only valid in response to a preceding and corresponding `WC_DISC` command from the opposite direction. The `WC_DISC` and `WC_DISCCNF` commands are valid during the connecting, data transfer, or disconnecting phases.

The `WC_CONCNF` command is only valid in response to a preceding and corresponding `WC_CONNECT` command from the opposite direction. The `WC_CONNECT` and `WC_CONCNF` commands are valid during the idle or connecting phases.

### Response

Should an error occur, an `M_PROTO` message is sent upstream with an appropriate error code, resulting in the failure of all system acalls on the Stream until closed. The `WAN_CTL` primitive may fail when:

`[EINVAL]` The `M_PROTO` message block is of an incorrect length for the primitive, or the *wan_command* is invalid.

| | |
|---|---|
| [ENXIO] | The underlying device driver has encountered a fatal error. |
| [EIO] | The WAN_CTL primitive was issued form an incorrect state for the subnetwork entity. |
| [E2BIG] | The maximum receive buffer size is too small to hold the largest frame required by the device. |

### Equivalence

When sent from the WAN user, the WC_CONNECT command corresponds to the CD_ENABLE_REQ(7) primitive of the cdi(7). When sent from the WAN driver, the primitive has no corresponding primitive. When from the WAN driver, WC_CONCNF corresponds to CD_ENABLE_CON(7). When from the WAN user, WC_CONCNF has no corresponding primitive.

When to the WAN driver, WC_DISC corresponds to CD_DISABLE_REQ(7). When from the WAN driver to the WAN user, WC_DISC corresponds to CD_ERROR_IND(7). When from the WAN driver, WC_DISCCNF corresponds to CD_DISABLE_CON(7). When to the WAN driver, WC_DISCCNF has no corresponding primitive.

### Compatibility

Some implementations ignore all of the parameter fields of the wan_ctl structure other than *wan_type* and *wan_command*, as is normally the case for WAN_NONE: no call procedural definitions.

## 4.4 WAN_DAT - Data

This primitive class provides two primitives for the transfer of data across the service interface. Attached M_DATA message blocks contain user data.

### Format

The WAN_DAT primitive contains a wan_msg structure. The primitive consists of one M_PROTO message block followed by one or more M_DATA message blocks containing user data. The M_PROTO message block is structured as follows:

```
struct wan_msg {
    uint8_t wan_type;
    uint8_t wan_command;
};
```

### Usage

The WC_TX command specifies that the user data in the associated M_DATA message blocks consist of data for transmission.

The WC_RX command indicates that the user data in the associated M_DATA message blocks consist of received data.

### Parameters

The wan_msg structure contains the following members:

*wan_type* Specifies the type of the structure contained in the M_PROTO message block. Always WAN_DAT.

*wan_command*
Conveys the WAN command. This field may assume one of the following values:

- WC_TX — Specifies that the user data in the associated M_DATA message blocks consist of data for transmission.
- WC_RX — Indicates that the user data in the associated M_DATA message blocks consist of received data.

### State

The WAN_DAT primitive may be issued by WAN user or WAN driver in the data transfer phase.

### Response

Should an error condition occur, an M_ERROR message is sent upstream with an appropriate error code, resulting in the failure of all system calls on the Stream until closed. The WAN_DAT primitive may fail when:

- [EINVAL] The M_PROTO message block is of an incorrect length for the primitive.
- [ENXIO] The underlying device driver has encountered a fatal error.
- [EIO] The WAN_DAT primitive was issued from an incorrect state for the subnetwork entity.
- [E2BIG] The maximum receive buffer size is too small to hold the largest frame required by the device.

### Equivalence

These primitives are equivalent to the CD_UNITDATA_REQ(7) and CD_UNITDATA_IND(7) primitives of the cdi(7). The WC_TX primitive, issued by the WAN user, is equivalent to the CD_UNITDATA_REQ(7) primitive; the WC_RX, issued by the WAN driver, the CD_UNITDATA_IND(7).

### Compatibility

Some implementations provide additional *wan_command* values with hardware- or implementation-specific fields. Some implementations also define a structure for the initial portion of the M_DATA block that contains media- or hardware-specific fields.

## 4.5 WAN_NTY - Notify

Registers for or provide notification of events for the *wan_snid* field of the primitive. When passed to the WAN driver, the primitive requests that the WAN driver record the events for which notification is to be given. When passed to the WAN user, the primitive notifies of a triggered event. This primitive corresponds to the CD_MODEM_SIG_IND(7) and CD_ERROR_IND(7) primitives of the cdi(7).

### Format

The primitive consists of a signle M_PROTO message block containing a wan_nty structure. The wan_nty structure is formatted as follows:

```
struct wan_nty {
    uint8_t wan_type;
    uint8_t wan_spare[3];
    uint32_t wan_snid;
    uint32_t wan_eventstat;
    uint32_t wan_reserved[2];
};
```

### Parameters

The wan_nty structure has the following members:

*wan_type* Specifies the type of the structure contained in the M_PROTO message block. Always WAN_NTY.

*wan_spare* Spare bytes for alignment: ignored by the responder and set to zero by the initiator.

*wan_snid* Conveys the subnetwork identifier. Equivalent to the Physical Point of Attachment (PPA) of the WAN interface.

*wan_eventstat*
This member contains one of the following bit masks:

| | |
|---|---|
| W_RECEIVE_BUFFER_OVFL | A received buffer overflow has occurred. |
| W_FRAMING_ERROR | A received frame framing error has occured. |
| W_TIMEOUT | A timeout has occurred. |
| W_HD_OVERRUN | A hardware device overrun has occurred. |
| W_ATTACHED_DEV_INACT | The attached device has gone inactive. |
| W_ATTACHED_DEV_ACTIVE | The attached device has become active. |
| W_FCS_ERR | A Frame Check Sequence (CRC) error has occurred. |
| W_CTS_ON | The CTS (Clear to Send) lead has gone high. |
| W_CTS_OFF | The CTS (Clear to Send) lead has gone low. |
| W_DCD_ON | The DCD (Data Carrier Detect) lead has gone high. |
| W_DCD_OFF | The DCD (Data Carrier Detect) lead has gone low. |
| W_DSR_ON | The DSR (Data Set Ready) lead has gone high. |
| W_DSR_OFF | The DSR (Data Set Ready) lead has gone low. |
| W_RI_ON | The RI (Ring Indicator) lead has gone high. |
| W_RI_OFF | The RI (Ring Indicator) lead has gone low. |
| W_PARITY_ERROR | A parity error has occurred on an asynchronous line. |
| W_BREAK_DETECTED | A break has been detected on an asyncrhonous line. |
| W_SHORT_FRAME | A short frame has been received. |
| W_TX_UNDERRUN | The transmitter FIFO has underrun. |
| W_ABORT | An aborted frame has been received. |

| | |
|---|---|
| W_RCL_NONZERO | |
| W_BSC_PAD_ERR | A Bisynchronous Character padding error has occured. |
| W_CTS_UNDERRUN | A Clear to Send underrun condition has occurred. |

*wan_reserved*
Reserved for future use: set to zero by the issuer and ignored by the receiver.

### State

The WAN_NTY primitive may be issued by WAN user or WAN driver in the connecting, data transfer, disconnecting, and disconnected phases.

### Response

When WAN_NTY is issued by the WAN driver, the WAN driver does not expect any response.
When issued by the WAN interface user, the WAN interface user expects the WAN driver to register the specified events and generate a WAN_NTY primitive should any of the registered events be detected, and to not generate a WAN_NTY primitive for any events that have not been registered.
Should an error occur as a result of a primitive issued by the WAN interface user, an M_PROTO message is sent upstream with an appropriate error code, resulting in the failure of all system acalls on the Stream until closed. The WAN_NTY primitive may fail when:

| | |
|---|---|
| [EINVAL] | The M_PROTO message block is of an incorrect length for the primitive, or the *wan_command* is invalid. |
| [ENXIO] | The underlying device driver has encountered a fatal error. |
| [EIO] | The WAN_CTL primitive was issued form an incorrect state for the subnetwork entity. |
| [E2BIG] | The maximum receive buffer size is too small to hold the largest frame required by the device. |

### Equivalence

When issued by the WAN user, the WAN_NTY primitive is equivalent to the CD_MODEM_SIG_POLL(7) primitive of the cdi(7). When issued by the WAN driver, the WAN_NTY primitive is equivalent to the CD_MODEM_SIG_IND(7) and CD_ERROR_IND(7) primitive.

### Compatibility

The WAN_NTY primitive is *OpenSS7*-specific and was modelled after the *IBM* WAN_NOTIFY primitive included in the *ARTIC* implementation.[1] While WAN_NTY is similar in structure and form to WAN_NOTIFY when issued by the WAN driver, WAN_NTY also permits registration of events when issued by the WAN interface user. This is accomplished in *ARTIC* using the W_SETLINE and W_GETLINE and other device-dependent input-output controls.
The default behaviour for a freshly created Stream is to not generate any notifications at all. This provides maximum compatibility with implementations for which applications programs, drivers and modules are not expecting to receive WAN_NTY or WAN_NOTIFY primitives.

[1] See [ARTIC WAN], page 77.

# 5 WAN Input-Output Controls

## 5.1 Input-Output Control Data Structures

The `/usr/include/openss7/sys/snet/wan_control.h` header file (`<sys/snet/wan_control.h>` with proper compile flags) defines a number of structures, pointers to which are used as arguments to input-output controls. These structures fall into four classes, identified by the value of the first byte of the structure, as follows:

`WAN_STATS` A `wan_stioc` structure that identifies the subnetwork and contains the state and statistics associated with the subnetwork. Used with the `W_ZEROSTATS` and `W_GETSTATS` input-output controls.

`WAN_TUNE` A `wan_tnioc` structure that idnetifies the subnetwork and contains the tunable parameters associated with the subnetwork. Used with the `W_SETTUNE` and `W_GETTUNE` input-output controls.

`WAN_MAP` A `wanmapgf`, `wanmappf` or `wanmapdf` structure that identifies the mapping entries, mapping entry or subnetwork, respectively. Used with the `W_GETWANMAP`, `W_PUTWANMAP` and `W_DELWANMAP` input-output controls, respectively.

`WAN_PLAIN` A `wan_hdioc` structure that identifies the subnetwork. Used with the `W_STATUS`, `W_ENABLE` and `W_DISABLE` input-output controls.

`WAN_SETSIG` A `wan_setsigf` structure that identifies the subnetwork and contains the setting for leads associated with the subnetwork. Used with the `W_SETSIG` and `W_GETSIG` input-output controls.

These structures are described in detail in the subsections that follow.

### 5.1.1 WAN_STATS - Statistics Data Structures

The value of `WAN_STATS` in the *w_type* field of the structure pointed to by the input-output control argument specifies that the pointed-to structure is a `wan_stioc` structure. This structure is used by the `W_ZEROSTATS` and `W_GETSTATS` input-output controls.

#### 5.1.1.1 wan_stioc Structure

The `wan_stioc` structure is formatted as follows:

```
struct wan_stioc {
    uint8_t w_type;
    uint8_t w_state;
    uint8_t w_spare[2];
    uint32_t w_snid;
    hdlcstats_t hdlc_stats;
};
```

The `wan_stioc` structure contains the following members:

*w_type* Specifies the type of the structure associated with the input-output control. Always `WAN_STATS`.

*w_state* Returns the state of the subnetwork entity. This member may have one of the following values:

| | |
|---|---|
| `HDLC_IDLE` | The raw HDLC connection is idle. |
| `HDLC_ESTB` | The raw HDLC connection is established. |
| `HDLC_DISABLED` | The raw HDLC connection is disabled. |
| `HDLC_CONN` | The raw HDLC connection is connecting. |
| `HDLC_DISC` | The raw HDLC connection is disconnecting. |

*w_spare* Spare bytes for alignment: set to zero (0) by the issuer and ignored by the responder.

*w_snid* Specifies the subnetwork identifier. Equivalent to the CDI Physical Point of Attachment (PPA).

*hdlc_stats* Contains the `hdlcstats_t` structure described below.

#### 5.1.1.2 hdlcstats_t Structure

The `hdlcstats_t` structure is formatted as follows:

```
typedef struct hstats {
    uint32_t hc_txgood;
    uint32_t hc_txurun;
    uint32_t hc_rxgood;
    uint32_t hc_rxorun;
    uint32_t hc_rxcrc;
    uint32_t hc_rxnobuf;
    uint32_t hc_rxnflow;
    uint32_t hc_rxoflow;
    uint32_t hc_rxabort;
    uint32_t hc_intframes;
} hdlcstats_t;
```

The `hdlcstats_t` structure has the following members, each reflecting a count since the last reset:

*hc_txgood* A count of the number of good frames transmitted since the last reset. This is a non-wrapping counter: should the counter reach its maximum value, it will no longer be incremented.

*hc_txurun* A count of the number of transmitter underruns since the last reset. This is a non-wrapping counter: should the counter reach its maximum value, it will no longer be incremented.

*hc_rxgood* A count of the number of good frames received since the last reset. This is a non-wrapping counter: should the counter reach its maximum value, it will no longer be incremented.

*hc_rxorun* A count of the number of receiver overruns since the last reset. This is a non-wrapping counter: should the counter reach its maximum value, it will no longer be incremented.

*hc_rxcrc* A count of the number of received CRC or framing errors since the last reset. This is a non-wrapping counter: should the counter reach its maximum value, it will no longer be incremented.

*hc_rxnobuf* A count of the number of receive buffer overflows since the last reset. This is a non-wrapping counter: should the counter reach its maximum value, it will no longer be incremented.

*hc_rxnflow* A count of the number of received frames with no flow control since the last reset. This is a non-wrapping counter: should the counter reach its maximum value, it will no longer be incremented.

*hc_rxoflow* A count of the number of received buffer overflows since the last reset. This is a non-wrapping counter: should the counter reach its maximum value, it will no longer be incremented.

*hc_rxabort* A count of the number of receiver aborts since the last reset. This is a non-wrapping counter: should the counter reach its maximum value, it will no longer be incremented.

*hc_intframes*
A count of the number of frames failed to be transmitted by the loss of modem signals or other physical medium error since the last reset. This is a non-wrapping counter: should the counter reach its maximum value, it will no longer be incremented.

### 5.1.2 WAN_TUNE - Tunable Data Structures

The value of `WAN_TUNE` in the *w_type* field of the structure pointed to by the input-output control argument specifies that the pointed-to structure is a `wan_tnioc` structure. This structure is used by the `W_SETTUNE` and `W_GETTUNE` input-output controls.

#### 5.1.2.1 wan_tnioc Structure

The `wan_tnioc` structure is formatted as follows:

```
struct wan_tnioc {
    uint8_t w_type;
    uint8_t w_spare[3];
    uint32_t w_snid;
    wantune_t wan_tune;
};
```

*w_type* Specifies the type of the structure associated with the input-output control. Always `WAN_TUNE`.

*w_spare* Spare bytes for alignment: set to zero (0) by the issuer and ignored by the responder.

*w_snid* Specifies the subnetwork identifier. Equivalent to the CDI Physical Point of Attachment (PPA).

*wan_tune* Contains the `wantune_t` structure described below.

#### 5.1.2.2 wantune_t Structure

The `wantune_t` structure is formatted as follows:

```
typedef struct {
    uint16_t WAN_options;
    struct WAN_hddef WAN_hd;
} wantune_t;
```

The `wantune_t` structure contains the following members:

*WAN_options*
: Specifies a number of WAN options associated with the device. This member is a bitwise OR of zero or more of the following values:

  `TRANSLATE` When set, indicates that a translation using the WAN remote address to interface address mapping function is to be performed.

*WAN_pad* Padding for alignment, set to zero by the issuer and ignored by the responder.

*WAN_hd* Contains the `WAN_hddef` structure described below.

#### 5.1.2.3 WAN_hddef Structure

The `WAN_hddef` structure is formatted as follows:

```
struct WAN_hddef {
    uint16_t WAN_maxframe;
    uint32_t WAN_baud;
    uint16_t WAN_interface;
    union {
        uint16_t WAN_cptype;
        struct WAN_x21 WAN_x21def;
        struct WAN_v25 WAN_v25def;
    } WAN_cpdef;
};
```

The `WAN_hddef` structure contains the following members:

*WAN_maxframe*
: Conveys the maximum frame size in octets.

*WAN_baud*
: Conveys the transmission rate in bits per second.

*WAN_interface*
: This member specifies the WAN interface. It can assume one of the following values:

| | |
|---|---|
| `WAN_X21` | The interface is a X.21 interface. |
| `WAN_V28` | The interface is a V.28 interface. |
| `WAN_V35` | The interface is a V.35 interface. |
| `WAN_V36` | The interface is a V.36 interface. |
| `WAN_RS232` | The interface is a RS-232 interface. |
| `WAN_RS422` | The interface is a RS-422 interface. |
| `WAN_T1E1` | The interface is a G.703/G.704 interface. |
| `WAN_ATM` | The interface is a ATM interface. |

*WAN_cpdef*
: This member defines a number of alternate call procedural defintions described by a union. The union contains the following members:

  *WAN_cptype*
  : Specifies the tye of the call procedural definitions. Always `WAN_NONE`, `WAN_X21P` or `WAN_V25bis`. When *WAN_cptype* is `WAN_NONE`, only this member of the union is significant.

  *WAN_x21def*
  : When *WAN_cptype* is `WAN_X21P`, this member of the union is significant. This member contains the `WAN_x21` structure described below.

  *WAN_v25def*
  : When *WAN_cptype* is `WAN_V25bis`, this member of the union is significant. This member contains the `WAN_v25` structure described below.

#### 5.1.2.4 WAN_x21 Structure

The `WAN_x21` structure is formatted as follows:

```
struct WAN_x21 {
    uint16_t WAN_cptype;
    uint16_t T1;
    uint16_t T2;
    uint16_t T3A;
    uint16_t T4B;
    uint16_t T5;
    uint16_t T6;
};
```

The `WAN_x21` structure has the following members:

*WAN_cptype*
: Specifies the type of the call procedural defintions. Always `WAN_X21P`.

*T1*
: Specifies the timeout for the call request state in deciseconds.

*T2*
: Specifies the timeout for the EOS to data transfer in deciseconds.

*T3A*
: Specifies the timeout for call progress signals in deciseconds.

*T4B*
: Specifies the timeout for DCE provided information in deciseconds.

*T5*
: Specifies the timeout for DTE clear request in deciseconds.

*T6*
: Specifies the timeout for DTE clear confirm state in deciseconds.

#### 5.1.2.5 WAN_v25 Structure

The `WAN_v25` structure is formatted as follows:

```
struct WAN_v25 {
    uint16_t WAN_cptype;
    uint16_t callreq;
};
```

The `WAN_v25` structure has the following members:

*WAN_cptype*
: Specifies the type of the call procedural defintions. Always `WAN_V25bis`.

*callreq*
: Contains the abort time (in deciseconds) for the call request command if the network does not support CFI.

### 5.1.3 WAN_MAP - Mapping Data Structures

The value of `WAN_MAP` in the *w_type* field of the structure pointed to by the input-output control argument specifies that the pointed-to structure is a `wanmapgf`, `wanmappf` or `wanmapdf` structure. These structures are used by the `W_GETWANMAP`, `W_PUTWANMAP` and `W_DELWANMAP` input-output controls, respectively.

#### 5.1.3.1 wanmapgf Structure

The `wanmapgf` structure is formatted as follows:

```
struct wanmapgf {
    uint8_t w_type;
    uint8_t w_spare[3];
    uint32_t w_snid;
    wanget_t wan_ents;
};
```

The `wanmapgf` structure contains the following members:

*w_type* Specifies the type of the structure associated with the input-output control. Always `WAN_MAP`.

*w_spare* Spare bytes for alignment: set to zero (0) by the issuer and ignored by the responder.

*w_snid* Specifies the subnetwork identifier. Equivalent to the CDI Physical Point of Attachment (PPA).

*wan_ents* Contains the `wanget_t` structure described below.

#### 5.1.3.2 wanget_t Structure

The `wanget_t` structure is formatted as follows:

```
typedef struct {
    uint16_t first_ent;
    uint16_t num_ent;
    wanmap_t entries[0];
} wanget_t;
```

The `wanget_t` structure contains the following members:

*first_ent* Specifies the index of the first entry in the *entries* member.

*num_ent* Specifies the number of entries in the *entries* member.

*entries* Contains *num_ent* entries of `wanmap_t` structures. The `wanmap_t` structure is described below.

#### 5.1.3.3 wanmappf Structure

The `wanmappf` structure is formatted as follows:

```
struct wanmappf {
    uint8_t w_type;
    uint8_t w_spare[3];
    uint32_t w_snid;
    wanmap_t wan_ent;
};
```

The `wanmappf` structure contains the following members:

*w_type* Specifies the type of the structure associated with the input-output control. Always `WAN_MAP`.

*w_spare* Spare bytes for alignment: set to zero (0) by the issuer and ignored by the responder.

*w_snid* Specifies the subnetwork identifier. Equivalent to the CDI Physical Point of Attachment (PPA).

*wan_ent* Contains the `wanmap_t` structure described below.

#### 5.1.3.4 wanmap_t Structure

The `wanmap_t` structure is formatted as follows:

```
typedef struct {
    uint8_t remsize;
    uint8_t remaddr[20];
    uint8_t infsize;
    uint8_t infaddr[30];
} wanmap_t;
```

The `wanmap_t` structure contains the following members:

*remsize* Conveys the size of the remote address contained in the *remaddr* field in octets.

*remaddr* Contains the remote address, significant to *remsize* octets.

*infsize* Conveys the size of the interface address contained in the *infaddr* field in octets.

*infaddr* Contains the remote address, significant to *infsize* octets.

#### 5.1.3.5 wanmapdf Structure

The `wanmapdf` structure is formatted as follows:

```
struct wanmapdf {
    uint8_t w_type;
    uint8_t w_spare[3];
    uint32_t w_snid;
};
```

The `wanmapdf` structure contains the following members:

*w_type* Specifies the type of the structure associated with the input-output control. Always `WAN_MAP`.

*w_spare* Spare bytes for alignment: set to zero (0) by the issuer and ignored by the responder.

*w_snid* Specifies the subnetwork identifier. Equivalent to the CDI Physical Point of Attachment (PPA).

### 5.1.4 WAN_PLAIN - Plain Data Structures

The value of `WAN_PLAIN` in the *w_type* field of the structure pointed to by the input-output control argument specifies that the pointed-to structure is a `wan_hdioc` structure. The structure is used by the `W_STATUS`, `W_ENABLE` and `W_DISABLE` input-output controls.

#### 5.1.4.1 wan_hdioc Structure

The `wan_hdioc` structure is formatted as follows:

```
struct wan_hdioc {
    uint8_t w_type;
    uint8_t w_spare[3];
    uint32_t w_snid;
};
```

The `wan_hdioc` structure contains the following members:

*w_type* Specifies the type of the structure associated with the input-output control. Always `WAN_PLAIN`.

*w_spare* Spare bytes for alignment: set to zero (0) by the issuer and ignored by the responder.

*w_snid* Specifies the subnetwork identifier. Equivalent to the CDI Physical Point of Attachment (PPA).

### 5.1.5 WAN_SETSIG - Signal and Lead Data Structures

The value `WAN_SETSIG` in the *w_type* field of the structure pointed to by the input-output control argument specifies that the pointed-to structure is a `wan_setsigf` structure. This structure is used by the `W_SETSIG` and `W_GETSIG` input-output controls.

#### 5.1.5.1 wan_setsigf Structure

The `wan_setsigf` structure is formatted as follows:

```
struct wan_setsigf {
    uint8_t w_type;
    uint8_t w_spare[3];
    uint32_t w_snid;
    wan_setsig_t wan_setsig;
};
```

The `wan_setsigf` structure contains the following members:

*w_type* Specifies the type of the structure associated with the input-output control. Always `WAN_SETSIG`.

*w_spare* Spare bytes for alignment: set to zero (0) by the issuer and ignored by the responder.

*w_snid* Specifies the subnetwork identifier. Equivalent to the CDI Physical Point of Attachment (PPA).

*wan_setsig* Contains the `wan_setsig_t` structure described below.

#### 5.1.5.2 wan_setsig_t Structure

The `wan_setsig_t` structure is formatted as follows:

```
typedef struct {
    uint8_t w_ctrlsignal;
    uint8_t w_reserved[3];
} wan_setsig_t;
```

The `wan_setsig_t` structure contains the following members:

*w_ctrlsignal*

Contains the control signals. This can be a bitwise OR of zero or more of the following:

| | |
|---|---|
| `W_RTS_HIGH` | Set or indicate RTS high, (or X.21 C signal). |
| `W_DTR_HIGH` | Set or indicate DTR high. |
| `W_DCD_HIGH` | Indicate DCD high. |
| `W_DSR_HIGH` | Indicate DSR high. |
| `W_CTS_HIGH` | Indicate CTS high, (or X.21 I signal). |
| `W_RI_HIGH` | Indicate RI high. |
| `W_RTS_LOW` | Set RTS low. |
| `W_DTR_LOW` | Set DTR low. |

*w_reserved* Reserved field: set to zero (0) by issuer and ignored by responder.

## 5.2 Input-Output Control Commands

The `/usr/include/openss7/sys/snet/wan_control.h` header file (`<sys/snet/wan_control.h>` with proper compile flags) defines a number of input-output controls, as follows:

| | |
|---|---|
| `W_ZEROSTATS` | Zeroes statistics associated with a subnetwork entity and collects the statistics and state of the subnetwork prior to reset. |
| `W_GETSTATS` | Retrieves the statistics and state associated with a subnetwork entity. |
| `W_SETTUNE` | Sets the tunable parameters associated with a subnetwork entity. |
| `W_GETTUNE` | Gets the tunable parameters associated with a subnetwork entity. |
| `W_PUTWANMAP` | Puts a remote address to interface address mapping entry. |
| `W_GETWANMAP` | Gets a block of remote address to interface address mapping entries. |
| `W_DELWANMAP` | Deletes all remote address to interface address mapping entries associated with a subnetwork entity. |
| `W_STATUS` | Retrieves the state of a subnetwork entity. |
| `W_ENABLE` | Enables a subnetwork entity for data transfer. |
| `W_DISABLE` | Disables a subnetwork entity from data transfer. |

These input-output controls are described in detail in the subsections that follow.

### 5.2.1 W_ZEROSTATS - Zero Statistics

#### Argument

This input-output control takes an argument that is a pointer to a `wan_stioc` structure, see Section 5.1.1.1 [wan˙stioc Structure], page 24.

#### Description

The `W_ZEROSTATS` input-output control requests that the WAN driver reset the statistics associated with the *w_snid* contained in the passed-in structure. The WAN driver is to reset the statistics, returning the statistics and state immediately before reset in the *hdlc_stats* and *w_state* members of the provided structure. See Section 5.1.1.2 [hdlcstats˙t Structure], page 25.

#### Return Value

When successful, the input-output control operation returns zero (0).
In addition to the errors that may be returned by `ioctl(2s)` and `streamio(7)`, errors that may be returned by this input-output control are as follows:

`[ENODEV]` The specified *w_snid* is unknown or invalid.

`[EINVAL]` The size of the structure corresponding to the passed-in argument is incorrect for the input-output control and *w_type* specified.

`[ENXIO]` The underlying device has encountered a fatal error.

#### Compatibility

### 5.2.2 W_GETSTATS - Get Statistics

#### Argument

This input-output control takes an argument that is a pointer to a `wan_stioc` structure, see Section 5.1.1.1 [wan˙stioc Structure], page 24.

#### Description

The `W_GETSTATS` input-output control requests that the WAN driver retrieve the statistics and state associated with the *w_snid* contained in the passed-in structure. The WAN driver is to retrieve the current statistics and state, returning them in the *hdlc_stats* and *w_state* members of the provided structure.

#### Return Value

When successful, the input-output control operation returns zero (0).
In addition to the errors that may be returned by `ioctl(2s)` and `streamio(7)`, errors that may be returned by this input-output control are as follows:

`[ENODEV]` The specified *w_snid* is unknown or invalid.

`[EINVAL]` The size of the structure corresponding to the passed-in argument is incorrect for the input-output control and *w_type* specified.

`[ENXIO]` The underlying device has encountered a fatal error.

#### Compatibility

### 5.2.3 W_SETTUNE - Set Tunables

#### Argument

This input-output control takes an argument that is a pointer to a `wan_tnioc` structure, see Section 5.1.2.1 [wan˙tnioc Structure], page 26.

#### Description

The `W_SETTUNE` input-output control requests that the WAN driver set the tunable parameters form the passed-in structure for the *w_snid* contained in that structure. The WAN driver is to set th tunable, returning any negotiated value in the provided structure.

#### Return Value

When successful, the input-output control operation returns zero (0).
In addition to the errors that may be returned by `ioctl(2s)` and `streamio(7)`, errors that may be returned by this input-output control are as follows:

`[ENODEV]` The specified *w_snid* is unknown or invalid.

`[EINVAL]` The size of the structure corresponding to the passed-in argument is incorrect for the input-output control and *w_type* specified.

`[ENXIO]` The underlying device has encountered a fatal error.

`[E2BIG]` The specified *WAN_maxframe* is of insufficient size to hold the maximum size frame necessary for proper operation of the protocol.

`[ENOMEM]` The WAN driver cannot allocate single message buffers of size *WAN_maxframe*.

`[EIO]` The interface is in a wrong state. For example, the tuning input-output control was issued after the interface was already registered with the `WAN_REG` primitive.

`[EXDEV]` The *WAN_interface* does not match the capabilities or mode of the hardware.

#### Compatibility

### 5.2.4 W_GETTUNE - Get Tunables

#### Argument

This input-output control takes an argument that is a pointer to a `wan_tnioc` structure, see Section 5.1.2.1 [wan˙tnioc Structure], page 26.

#### Description

The `W_GETTUNE` input-output control requests that the WAN driver get the tunable parameters associated with the *w_snid* contained in the passed-in structure. The WAN driver is to retrieve the tunable parameters and return them in the provided structure.

#### Return Value

When successful, the input-output control operation returns zero (0).
In addition to the errors that may be returned by `ioctl(2s)` and `streamio(7)`, errors that may be returned by this input-output control are as follows:

`[ENODEV]` The specified *w_snid* is unknown or invalid.

`[EINVAL]` The size of the structure corresponding to the passed-in argument is incorrect for the input-output control and *w_type* specified.

`[ENXIO]` The underlying device has encountered a fatal error.

#### Compatibility

### 5.2.5 W_PUTWANMAP - Put WAN Address Mapping

#### Argument

This input-output control takes an argument that is a pointer to a `wanmappf` structure, see Section 5.1.3.3 [wanmappf Structure], page 32.

#### Description

The `W_PUTWANMAP` input-output control requests that the WAN driver add a remote address to interface address mapping entry associated to the specified subnetwork identifier, *w_snid*.

#### Return Value

When successful, the input-output control operation returns zero (0).
In addition to the errors that may be returned by `ioctl(2s)` and `streamio(7)`, errors that may be returned by this input-output control are as follows:

`[ENODEV]` The specified *w_snid* is unknown or invalid.

`[EINVAL]` The size of the structure corresponding to the passed-in argument is incorrect for the input-output control and *w_type* specified.

`[ENXIO]` The underlying device has encountered a fatal error.

#### Compatibility

### 5.2.6 W_GETWANMAP - Get WAN Address Mapping

#### Argument

This input-output control takes an argument that is a pointer to a `wanmapgf` structure, see Section 5.1.3.1 [wanmapgf Structure], page 31.

#### Description

The `W_GETWANMAP` input-output control requests that the WAN driver retrieve a block of remote address to interface address mapping entries associated with the specified subnetwork identifier, *w_snid*.

#### Return Value

When successful, the input-output control operation returns zero (0).
In addition to the errors that may be returned by `ioctl(2s)` and `streamio(7)`, errors that may be returned by this input-output control are as follows:

`[ENODEV]` The specified *w_snid* is unknown or invalid.

`[EINVAL]` The size of the structure corresponding to the passed-in argument is incorrect for the input-output control and *w_type* specified.

`[ENXIO]` The underlying device has encountered a fatal error.

#### Compatibility

### 5.2.7 W_DELWANMAP - Delete WAN Address Mappings

#### Argument

This input-output control takes an argument that is a pointer to a `wanmapdf` structure, see Section 5.1.3.5 [wanmapdf Structure], page 33.

#### Description

The `W_DELWANMAP` input-output control requests that the WAN driver delete all remote address to interface address mapping entries associatedw with the specified subnetwork identifier, *w_snid*.

#### Return Value

When successful, the input-output control operation returns zero (0).
In addition to the errors that may be returned by `ioctl(2s)` and `streamio(7)`, errors that may be returned by this input-output control are as follows:

`[ENODEV]` The specified *w_snid* is unknown or invalid.

`[EINVAL]` The size of the structure corresponding to the passed-in argument is incorrect for the input-output control and *w_type* specified.

`[ENXIO]` The underlying device has encountered a fatal error.

#### Compatibility

### 5.2.8 W_STATUS - Get Interface Status

#### Argument

This input-output control takes an argument that is a pointer to a `wan_hdioc` structure, see Section 5.1.4.1 [wan˙hdioc Structure], page 34.

#### Description

The `W_STATUS` input-output control requests that the WAN driver return the status of the *w_snid* contained in the passed-in structure.

#### Return Value

When successful, the input-output control operation returns zero (0) when the associated *w_snid* is disabled, and one (1) when the associated *w_snid* is enabled.
In addition to the errors that may be returned by `ioctl(2s)` and `streamio(7)`, errors that may be returned by this input-output control are as follows:

`[ENODEV]` The specified *w_snid* is unknown or invalid.

`[EINVAL]` The size of the structure corresponding to the passed-in argument is incorrect for the input-output control and *w_type* specified.

`[ENXIO]` The underlying device has encountered a fatal error.

#### Compatibility

### 5.2.9 W_ENABLE - Enable Interface Data Transfer

#### Argument

This input-output control takes an argument that is a pointer to a `wan_hdioc` structure, see Section 5.1.4.1 [wan˙hdioc Structure], page 34.

#### Description

The `W_ENABLE` input-output control requests that the WAN driver enable data transfer for the *w_snid* contained in the passed-in structure.

#### Return Value

When successful, the input-output control operation returns zero (0).
In addition to the errors that may be returned by `ioctl(2s)` and `streamio(7)`, errors that may be returned by this input-output control are as follows:

`[ENODEV]` The specified *w_snid* is unknown or invalid.

`[EINVAL]` The size of the structure corresponding to the passed-in argument is incorrect for the input-output control and *w_type* specified.

`[ENXIO]` The underlying device has encountered a fatal error.

#### Compatibility

### 5.2.10 W_DISABLE - Disable Interface Data Transfer

#### Argument

This input-output control takes an argument that is a pointer to a `wan_hdioc` structure, see Section 5.1.4.1 [wan˙hdioc Structure], page 34.

#### Description

The `W_DISABLE` input-output control requests that the WAN driver disable data transfer for the *w_snid* contained in the passed-in structure.

#### Return Value

When successful, the input-output control operation returns zero (0).
In addition to the errors that may be returned by `ioctl(2s)` and `streamio(7)`, errors that may be returned by this input-output control are as follows:

`[ENODEV]` The specified *w_snid* is unknown or invalid.

`[EINVAL]` The size of the structure corresponding to the passed-in argument is incorrect for the input-output control and *w_type* specified.

`[ENXIO]` The underlying device has encountered a fatal error.

#### Compatibility

### 5.2.11 W_SETSIG - Set Signals and Leads

#### Argument

This input-output control takes an argument that is a pointer to a `wan_setsigf` structure, see Section 5.1.5.1 [wan˙setsigf Structure], page 35.

#### Description

The `W_SETSIG` input-output control requests that the WAN driver set the signals and leads as specified. This input-output control is equivalent to the `CD_MODEM_SIG_REQ(7)` primitive of the `cdi(7)`.

#### Return Value

When successful, the input-output control operation returns zero (0).
In addition to the errors that may be returned by `ioctl(2s)` and `streamio(7)`, errors that may be returned by this input-output control are as follows:

`[ENODEV]` The specified *w_snid* is unknown or invalid.

`[EINVAL]` The size of the structure corresponding to the passed-in argument is incorrect for the input-output control and *w_type* specified.

`[ENXIO]` The underlying device has encountered a fatal error.

#### Compatibility

### 5.2.12 W_GETSIG - Get Signals and Leads

#### Argument

This input-output control takes an argument that is a pointer to a `wan_setsigf` structure, see Section 5.1.5.1 [wan˙setsigf Structure], page 35.

#### Description

The `W_GETSIG` input-output control requests that the WAN driver retrieve the signals and leads. This input-output control is equivalent to the `CD_MODEM_SIG_POLL(7)` primitive of the `cdi(7)`.

#### Return Value

When successful, the input-output control operation returns zero (0).
In addition to the errors that may be returned by `ioctl(2s)` and `streamio(7)`, errors that may be returned by this input-output control are as follows:

`[ENODEV]` The specified *w_snid* is unknown or invalid.

`[EINVAL]` The size of the structure corresponding to the passed-in argument is incorrect for the input-output control and *w_type* specified.

`[ENXIO]` The underlying device has encountered a fatal error.

#### Compatibility

# Appendix A WAN Header Files

Applications using the Wide Area Network (WAN) Interface need to include several system header files:

```
<errno.h>

<sys/types.h>
<sys/ioctl.h>
<sys/stropts.h>
<sys/snet/wan_proto.h>
<sys/snet/wan_control.h>
```

## A.1 WAN Protocol Header File

```
<errno.h>

<sys/types.h>
<sys/ioctl.h>
<sys/stropts.h>
<sys/snet/wan_proto.h>
```

Note that on *IRIS SX.25* this file is located in `<sys/snet/wan_proto.h>`. Note that on *Solaris X.25* this file is located in `<sys/netx25/wan_proto.h>`.

```
#ifndef __SYS_SNET_WAN_PROTO_H__
#define __SYS_SNET_WAN_PROTO_H__

/*
 * This file contains a basic SpiderWAN like interface.    Source compatibility is
 * attempted.    Binary compatibility is not attempted (but may result, YMMV).
 * Portable applications programs, STREAMS drivers and modules, should use the
 * CDI interface instead.
 */

/*
 * Primitive types:  These constant values are used in the wan_type field of the
 * various primitives.
 */
#define WAN_SID         1       /* Set subnetwork identifier */
#define WAN_REG         2       /* Register subnetwork identifier */
#define WAN_CTL         3       /* control connection */
#define WAN_DAT         4       /* transfer data */
#define WAN_NTY         5       /* register and notify events */

/*
 * WAN_SID - one M_PROTO message block
 *
 * This primitive assigns a subnetwork identifier to the Stream upon which it
 * issued.    It is equivalent to an ATTACH.
 */
struct wan_sid {
        uint8_t wan_type;               /* always WAN_SID */
        uint8_t wan_spare[3];           /* spare for alignment */
        uint32_t wan_snid;              /* subnetwork ID */
};
```

```
/*
 * WAN_REG - one M_PROTO message block
 *
 * This primitive enables the subnetwork identifier for use.    The subnetwork
 * identifier does not need to match the Stream upon which the primitive is
 * issued.
 */
struct wan_reg {
        uint8_t wan_type;               /* always WAN_REG */
        uint8_t wan_spare[3];           /* spare for alignment */
        uint32_t wan_snid;              /* subnetwork ID */
};

/*
 * WAN_CTL Primitives - one M_PROTO message block
 *
 * These primitives come in eight forms:  WC_CONNECT, WC_CONCNF, WC_DISC and
 * WC_DISCCNF, each issued in either direction, to or from the WAN driver.
 */
/*
 * WAN_CTL command types:  These constant values are used in the wan_command
 * field of the WAN_CTL primitive.
 */
#define WC_CONNECT      1       /* connect */
#define WC_CONCNF       2       /* connect confirm */
#define WC_DISC         3       /* disconnect */
#define WC_DISCCNF      4       /* disconnect confirm */

/*
 * Address types:  These constant values are used in the wan_remtype field of the
 * WAN_CTL primitive.
 */
#define WAN_TYPE_ASC    1       /* digits are ASCII digits, length is octets */
#define WAN_TYPE_BCD    2       /* digits are BCD digits, length is nibbles */

/*
 * Status values:  These constant values are used in the wan_status field of the
 * WAN_CTL primitive.
 */
#define WAN_FAIL        0       /* operation failed */
#define WAN_SUCCESS     1       /* operation successful */

/*
 * WAN_CTL primitive, consists of one M_PROTO message block.
 */
struct wan_ctl {
        uint8_t wan_type;               /* always WAN_CTL */
        uint8_t wan_command;            /* command:  WC_CONNECT, WC_CONCNF, WC_DISC, WC_DISCCNF */
        uint8_t wan_remtype;            /* remote address type:  WAN_TYPE_ASC or WAN_TYPE_BCD */
        uint8_t wan_remsize;            /* size of remote address in octets or semi-octets */
        uint8_t wan_remaddr[20];        /* the remote address */
        uint8_t wan_status;             /* status:  WAN_SUCCESS or WAN_FAIL */
        uint8_t wan_diag;               /* diagnostic when failed */
};
```

```
/*
 * WAN_DAT Primitives - one M_PROTO and one or more M_DATA message blocks
 *
 * These primitives come in two forms:  WC_TX and WC_RX.  WC_RX are issued by
 * the WAN driver, and WC_TX are issued to the WAN driver.    Any addresses must
 * be implicit or included in the data.
 */
#define WC_TX           1       /* data for transmission */
#define WC_RX           2       /* received data */

/*
 * WAN_MSG primitive, consists of one M_PROTO message block followed by one or
 * more M_DATA message blocks.
 */
struct wan_msg {
        uint8_t wan_type;               /* always WAN_MSG */
        uint8_t wan_command;            /* WC_TX or WC_RX */
};

/*
 * WAN_NTY Primitive - one M_PROTO message block
 *
 * When the WAN_NTY primitive is sent to the WAN driver, the WAN driver marks
 * the bits that are masked and generates notifications for those events that
 * have a 1 in the corresponding bit location.    When the event occurs, the WAN
 * driver will issue a WAN_NTY primitive upstream.    By default, and for
 * compatibility, no events are registered for notification.
 */

/* definitions for wan_eventstat field */
#define W_RECEIVE_BUFFER_OVFL   (1<< 0)
#define W_FRAMING_ERROR         (1<< 1)
#define W_TIMEOUT               (1<< 2)
#define W_HD_OVERRUN            (1<< 3)
#define W_ATTACHED_DEV_INACT    (1<< 4)
#define W_ATTACHED_DEV_ACTIVE   (1<< 5)
#define W_FCS_ERR               (1<< 7)
#define W_CTS_ON                (1<< 8)
#define W_CTS_OFF               (1<< 9)
#define W_DCD_ON                (1<<10)
#define W_DCD_OFF               (1<<11)
#define W_DSR_ON                (1<<12)
#define W_DSR_OFF               (1<<13)
#define W_RI_ON                 (1<<14)
#define W_RI_OFF                (1<<15)
#define W_PARITY_ERROR          (1<<16)
#define W_BREAK_DETECTED        (1<<17)
#define W_SHORT_FRAME           (1<<18)
#define W_TX_UNDERRUN           (1<<19)
#define W_ABORT                 (1<<20)
#define W_RCL_NONZERO           (1<<21)
#define W_BSC_PAD_ERR           (1<<22)
#define W_CTS_UNDERRUN          (1<<23)

struct wan_nty {
        uint8_t wan_type;               /* always WAN_NTY */
```

```
        uint8_t wan_spare[3];           /* spare for alignment */
        uint32_t wan_snid;              /* subnetwork identifier */
        uint32_t wan_eventstat;         /* event status */
        uint32_t wan_reserved[2];       /* reserved for future use */
};

union WAN_primitives {
        uint8_t wan_type;
        struct wan_sid wsid;            /* WAN_SID primitives */
        struct wan_reg wreg;            /* WAN_REG primitives */
        struct wan_ctl wctl;            /* WAN_CTL primitives */
        struct wan_dat wdat;            /* WAN_DAT primitives */
        struct wan_nty wnty;            /* WAN_NTY primitives */
};

#endif                          /* __SYS_SNET_WAN_PROTO_H__ */
```

## A.2 WAN Control Header File

`<errno.h>`

`<sys/types.h>`
`<sys/ioctl.h>`
`<sys/stropts.h>`
`<sys/snet/wan_proto.h>`
`<sys/snet/wan_control.h>`
Note that on *IRIS SX.25* this file is located in `<sys/snet/wan_control.h>`. Note that on *Solaris X.25* this file is located in `<sys/netx25/wan_control.h>`.

```
#ifndef __SYS_SNET_WAN_CONTROL_H__
#define __SYS_SNET_WAN_CONTROL_H__

/*
 * This file contains a basic SpiderWAN like input-output control interface.
 * Source compatibility is attempted.    Binary compatibility is not attempted
 * (but may result, YMMV).  Portable applications programs, STREAMS drivers
 * and modules, should use the CDI interface instead.
 */

/* values for w_type field */
#define WAN_STATS       1       /* 0x34 wan_stioc structure */
#define WAN_TUNE        2       /* 0x35 wan_tnioc structure */
#define WAN_MAP         3       /* 0x36 wan_mpioc union */
#define WAN_PLAIN       4       /* 0x37 wan_hdioc structure */
#define WAN_SETSIG      5       /* 0x38 wan_setsigf structure */

/* WAN_STATS structures */

typedef struct hstats {
        uint32_t hc_txgood;             /* good frames transmitted */
        uint32_t hc_txurun;             /* transmitter underruns */
        uint32_t hc_rxgood;             /* good frames received */
        uint32_t hc_rxorun;             /* receiver overruns */
        uint32_t hc_rxcrc;              /* CRC or framing errors */
        uint32_t hc_rxnobuf;            /* no receive buffer */
```

```
        uint32_t hc_rxnflow;            /* frame received no flow control */
        uint32_t hc_rxoflow;            /* buffer overflows */
        uint32_t hc_rxabort;            /* received aborts */
        uint32_t hc_intframes;          /* tranmission failures */
} hdlcstats_t;

/* values for w_state field */
#define HDLC_IDLE       0       /*  0 */
#define HDLC_ESTB       1       /* 30 */
#define HDLC_DISABLED   2       /* 31 */
#define HDLC_CONN       3       /* 40 */
#define HDLC_DISC       4       /* 41 */

struct wan_stioc {
        uint8_t w_type;                 /* always WAN_STATS */
        uint8_t w_state;                /* HDLC state */
        uint8_t w_spare[2];             /* spare bytes for alignment */
        uint32_t w_snid;                /* subnetwork identifier */
        hdlcstats_t hdlc_stats;         /* statistics */
};

/* WAN_TUNE structures */

/* values for WAN_cptype field */
#define WAN_NONE        0
#define WAN_X21P        1
#define WAN_V25bis      2

/* for a description of timers and defaults, see X.21 Annex C DTE Timers */
struct WAN_x21 {
        uint16_t WAN_cptype;            /* Always WAN_X21P. */
        uint16_t T1;                    /* X.21 T1:  call-request to proceed-to-select timer */
        uint16_t T2;                    /* X.21 T2:  end-of-selection to ready-for-data timer */
        uint16_t T3A;                   /* X.21 T3A: addtn'l call prog or DCE provided info timer */
        uint16_t T4B;                   /* X.21 T4B: call-accept to ready-for-data timer */
        uint16_t T5;                    /* X.21 T5:  DTE clear-request to DCE ready timer */
        uint16_t T6;                    /* X.21 T6:  DTE clear-confirmation to DCE ready timer */
};

/* default values for the WAN_x21 structure */
const struct WAN_x21 WAN_x21_defaults = {
        .WAN_cptype = WAN_X21P, /* Always WAN_X21P */
        .T1 = 30,               /* 3.0 seconds */
        .T2 = 200,              /* 20.0 seconds */
        .T3A = 60,              /* 6.0 seconds */
        .T4B = 60,              /* 6.0 seconds */
        .T5 = 20,               /* 2.0 seconds */
        .T6 = 20,               /* 2.0 seconds */
};

/* for a description of timers and defaults, see V.25 bis Clause 5.2 */
struct WAN_v25 {
        uint16_t WAN_cptype;            /* Always WAN_V25bis.  */
        uint16_t callreq;               /* V.25 bis T1:  call init.  to call estab.  timer */
};
```

```
/* default values for the WAN_v25 structure */
const struct WAN_v25 WAN_v25_defaults = {
        .WAN_cptype = WAN_V25bis,       /* Always WAN_X21P */
        .callreq = 6000,        /* 600.0 seconds, 5 minutes */
};

/* values for WAN_interface field */
#define WAN_X21         0
#define WAN_V28         1
#define WAN_V35         2

/* values for WAN_phy_if field */
#define WAN_DTE         0
#define WAN_DCE         1

struct WAN_hddef {
        uint16_t WAN_maxframe;          /* WAN maximum frame size (octets).  */
        uint32_t WAN_baud;              /* WAN baud rate.  */
        uint16_t WAN_interface;         /* WAN physical interface.  */
        union {
                uint16_t WAN_cptype;    /* significant when WAN_NONE */
                struct WAN_x21 WAN_x21def;      /* significant when WAN_X21P */
                struct WAN_v25 WAN_v25def;      /* significant when WAN_V25bis */
        } WAN_cpdef;                    /* call procedure definitions */
};

/* values for WAN_options field */
#define TRANSLATE       0x0001

typedef struct wantune {
        uint16_t WAN_options;
        struct WAN_hddef WAN_hd;
} wantune_t;

struct wan_tnioc {
        uint8_t w_type;                 /* always WAN_TUNE */
        uint8_t w_spare[3];             /* spare bytes for alignment */
        uint32_t w_snid;                /* subnetwork identifier */
        wantune_t wan_tune;             /* WAN tunables */
};

/* WAN_MAP structures */

typedef struct wanmap {
        uint8_t remsize;                /* remote address size in octets */
        uint8_t remaddr[20];            /* remote address containing remsize octets */
        uint8_t infsize;                /* interface address size in octets */
        uint8_t infaddr[30];            /* interface address containing infsize octets */
} wanmap_t;

typedef struct wanget {
        uint16_t first_ent;             /* index of first entry in entries member */
        uint16_t num_ent;               /* number of entries in entries member */
        wanmap_t entries[1];
        /* followed by (num_ent - 1) * sizeof(wanmap_t) entry buffer */
} wanget_t;
```

```
/* used with W_GETWANMAP */
struct wanmapgf {
        uint8_t w_type;                 /* always WAN_MAP */
        uint8_t w_spare[3];             /* spare bytes for alignment */
        uint32_t w_snid;                /* subnetwork identifier */
        wanget_t wan_ents;              /* block of mapping entries */
};

/* used with W_PUTWANMAP */
struct wanmappf {
        uint8_t w_type;                 /* always WAN_MAP */
        uint8_t w_spare[3];             /* spare bytes for alignment */
        uint32_t w_snid;                /* subnetwork identifier */
        wanmap_t wan_ent;               /* single mapping entry */
};

/* used with W_DELWANMAP */
struct wanmapdf {
        uint8_t w_type;                 /* always WAN_MAP */
        uint8_t w_spare[3];             /* spare bytes for alignment */
        uint32_t w_snid;                /* subnetwork identifier */
};

union wan_mpioc {
        uint8_t w_type;                 /* always WAN_MAP */
        struct wanmapgf wan_getmap;     /* W_GETWANMAP structure */
        struct wanmappf wan_putmap;     /* W_PUTWANMAP structure */
        struct wanmapdf wan_delmap;     /* W_DELWANMAP structure */
};

/* WAN_PLAIN structures */

struct wan_hdioc {
        uint8_t w_type;                 /* always WAN_PLAIN */
        uint8_t w_spare[3];             /* spare bytes for alignment */
        uint32_t w_snid;                /* subnetwork identifier */
};

/* WAN_SETSIG structures */

/* definitions for w_ctrlsignal field */
#define W_RTS_HIGH      (1<<0)  /* set or indicate RTS high (X.21 C) */
#define W_DTR_HIGH      (1<<1)  /* set or indicate DTR high */
#define W_DCD_HIGH      (1<<2)  /* indicate DCD high */
#define W_DSR_HIGH      (1<<3)  /* indicate DSR high */
#define W_CTS_HIGH      (1<<4)  /* indicate CTS high (X.21 I) */
#define W_RI_HIGH       (1<<5)  /* indicate RI high */
#define W_RTS_LOW       (1<<6)  /* set RTS low */
#define W_DTR_LOW       (1<<7)  /* set DTR low */

typedef struct wan_setsig {
        uint8_t w_ctrlsignal;
        uint8_t w_reserved[3];
} wan_setsig_t;
```

```
struct wan_setsigf {
        uint8_t w_type;                 /* always WAN_SETSIG */
        uint8_t w_spare[3];             /* spare bytes for alignment */
        uint32_t w_snid;                /* subnetwork identifier */
        wan_setsig_t wan_setsig;        /* signals and leads set */
};

#define W_ZEROSTATS     (('W'<<8)|000)  /* zero statistics */
#define W_GETSTATS      (('W'<<8)|001)  /* get statistics */
#define W_SETTUNE       (('W'<<8)|002)  /* set tunables */
#define W_GETTUNE       (('W'<<8)|003)  /* get tunables */
#define W_PUTWANMAP     (('W'<<8)|004)  /* put address mapping */
#define W_GETWANMAP     (('W'<<8)|005)  /* get address mappings */
#define W_DELWANMAP     (('W'<<8)|006)  /* del address mappings */
#define W_STATUS        (('W'<<8)|007)  /* get interface status */
#define W_ENABLE        (('W'<<8)|010)  /* enable interface */
#define W_DISABLE       (('W'<<8)|011)  /* disable interface */
#define W_SETSIG        (('W'<<8)|012)  /* set signals and leads */
#define W_GETSIG        (('W'<<8)|013)  /* get signals and leads */
#define W_POLLSIG       (('W'<<8)|014)  /* poll signals and leads */

#endif                          /* __SYS_SNET_WAN_CONTROL_H__ */
```

# Appendix B WAN Drivers and Modules

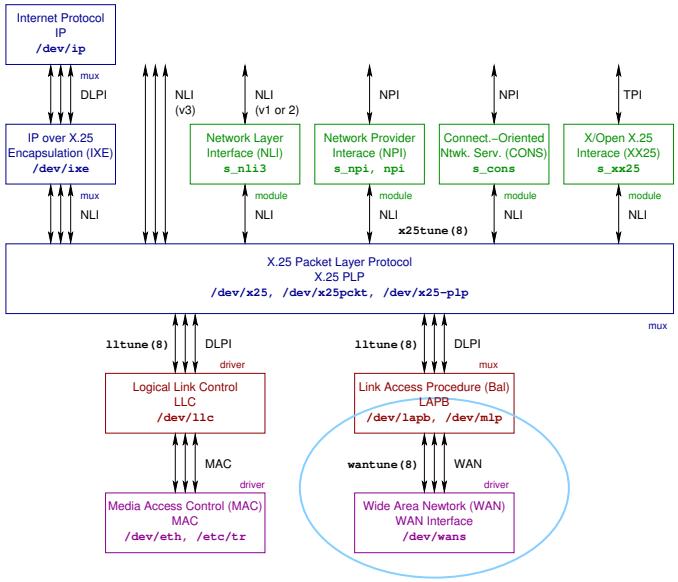


Figure B.1: *SpiderX.25 WAN Drivers and Modules*

## B.1 WAN Module

The *WAN Module* is a pushable STREAMS module named `s_wan`. Its purpose is to take an *OpenSS7* Communications Device Interface (CDI) Stream and convert it for use as a WAN interface Stream by applications programs, drivers or modules expecting the *SpiderX.25* interface. The insertion and use of this module is illustrated in Figure B.2.

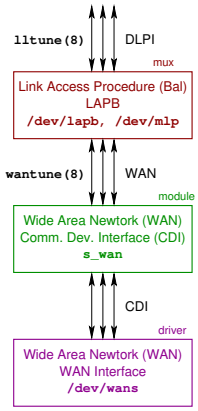


Figure B.2: *SpiderX.25 WAN Conversion Module:* `s_wan`

The `s_wan` pushable STREAMS module accepts a Communications Devce Interface (CDI) at its lower service boundary and provides a Wide Area Network (WAN) Interface at its upper service boundary.

Note that, as `s_wan` is a pushable module, it is possible to include an `autopush(8)` specification a driver providing the Communications Device Interface (CDI), to provide a specialized device minor or minor device name that clones Device Layers following the WAN approach.

```
#include <sys/types.h>
#include <sys/stropts.h>
#include <sys/errno.h>
#include <sys/error.h>
#include <sys/snet/wan_proto.h>
#include <sys/snet/wan_control.h>

int fd;

/* Open the communications style device. */

if ((fd = open("/dev/cd", O_RDWR)) < 0) {
        perror();
        exit(1);
}

/* Push the WAN style module. */

if (ioctl(fd, I_PUSH, "s_wan") < 0) {
        perror();
        exit(1);
}

/* At this point we can talk to the Stream using
 * the service primitives and input-output controls
 * of the WAN interface. */

struct wan_tnioc tune;

tune.w_type = WAN_TUNE;
```

```
tune.w_snid = snid;

if (ioctl(fd, W_GETTUNE, &tune) < 0) {
        perror();
        exit(1);
}
```

## B.2 CDI Module

The *CDI Module* is a pushable STREAMS module named `s_cdi`. Its purpose is to take a *SpiderX.25* WAN interface (WAN) Stream and convert it for use as a Communications Device utilizing the Communications Device Interface (CDI). The insertion and use of this module is illustrated in Figure B.3.

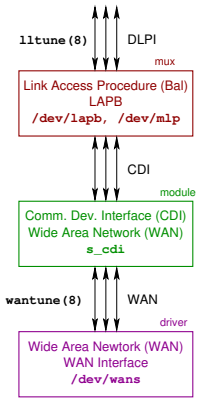


Figure B.3: *OpenSS7 CDI Conversion Module:* `s_cdi`

The `s_cdi` pushable STREAMS module accepts a Wide Area Network (WAN) Interface at its lower service boundary and provides a Communications Devce Interface (CDI) at its upper service boundary.

Note that, as `s_cdi` is a pushable module, it is possible to include an `autopush(8)` specification a driver providing the Wide Area Network (WAN) Interface, to provide a specialized device minor or minor device name that clones Device Layers following the CDI approach.

# Appendix C WAN Utilities

## C.1 WAN Tuning Utility

The WAN Tuning Utility is also documented as a manual page, `wantune(8)`.

### Name

`wantune` – manage WAN tunable parameters

### Synopsis

```
wantune [options] [-G] -s subnet_id [-d devname] [filename]
wantune [options]  -P  -s subnet_id [-d devname] [filename]
wantune {-h|--help}
wantune {-V|--version}
wantune {-C|--copying}
```

### Description

`wantune` is a configuration command intended to be executed from system configuration scripts, and, in particular, the `xnetd(8)` configuration daemon. Its purpose is to alter or interrogate the tunable parameters of a WAN data link connected to an identified subnetwork.

### Options

The `wantune` command accepts the following options:

### Command Options

The following command options are mutually exclusive (except for `-h`, `-V` and `-C` which never cause an error when specified with another command option). If no command option is given, `-G` is assumed.

`-G, --get` Retrieve configuration information for the specified *subnet_id*, from the default or specified *device*, and write the output to `stdout` (or *filename*, when given). This option can be used to create a properly formatted configuration file from an existing system configuration.

`-P, --put` Load configuration information for the specified *subnet_id*, to the default or specified *device*, getting the configuration input from `stdin` (or *filename*, when given). This option can be used to tune current system configuration.

`-h, --help` When this option is encountered, usage information is printed to `stdout`, option processing stops, and the program exists successfully without taking any further action.

`-V, --version`
When this option is encountered, version information is printed to `stdout`, option processing stops, and the program exits successfully without taking any further action.

`-C, --copying`
When this option is encountered, copying permissions are printed to `stdout`, option processing stops, and the program exits successfully without taking any further action.

### Non-Command Options

The following non-command options cab be combined together and with any command option. Non-command options that are not necessary for the specified command option do not generate an error by mere combination.

`-e, --extended`
: Normally `wantune` processes a fixed number of lines from `stdin` (or *filename*, when specified), and outputs a fixed number of lines to `stdout` (or *filename*, when specified). This fixed number of lines are strictly compatible with other implementations of `wantune`.

  When the `-e` option is specified, additional lines are accepted on input and are generated on output. For the format of the fixed lines and the additional lines, see Section D.2 [WAN Tuning File Format], page 70.

`-s, --subnet` *subnet_id*
: Specifies the subnetwork identifier, *subnet_id*, to which the tuning operation applies. *subnet_id* is normally an alphabetical character starting at '`A`' for the first subnetwork, '`B`' for the second subnetwork, and so on. This option must always be given when the `-P` or `-G` options are present or assumed.

`-d, --device` *devname*
: Specifies the device, *devname*, to open when tuning. When unspecified, the default is `/dev/wans`. See also *Devices*, below.

`-n, --dryrun`
: Execute the command (`-P` or `-G`) as a dry run. When this option is specified with the `-P` option, the input is read and checked for validity, but the configuration is not written to the device when specified with the `-G` option, information is read from the device, but configuration information is not output. The exit status and diagnostic output of the command still reflects the success or failure of the command.

`-q, --quiet`
: Suppresses normal output. This is the same as '`--verbose=0`'.

`-D, --debug [`*level*`]`
: Increase or specify the debug verbosity *level*. The default debug *level* is zero (0). This option may be repeated. Level zero (0) corresponds to no debugging output.

`-v, --verbose [`*level*`]`
: Increase or specify the output verbosity *level*. The default output *level* is one (1). This option may be repeated. Level zero (0) corresponds to no normal output.

### Arguments

The following non-option arguments may be provided on the command line:

*filename*
: Specifies the *filename* from which to read (`-P` option) or write (`-G` option) configuration information.

  This argument is optional. When the *filename* is not given and the `-P` option is specified, the values are read from `stdin`; for the `-G` option, values are written to `stdout`.

  If the *filename* is an absolute path (i.e. begins with '`/`'), then *filename* is assumed to be the exact path specified. Otherwise, the file required is assumed to be

`/etc/sysconfig/openss7/template/`*filename*.[1] See Section D.2 [WAN Tuning File Format], page 70, for the format of the file.

### Diagnostics

An exit status of zero (0) indicates that the command was successful; one (1) indicates that an error occured and a diagnostic message is printed to `stderr`; two (2) indicates that the option or argument syntax was in error and a diagnostic message is printed to `stderr`.
The `--quiet` option suppresses the printing of normal output to `stdout` and diagnostic messages to `stderr`.

### File Format

For the input file format, see Section D.2 [WAN Tuning File Format], page 70.

### Notices

On input, this implementation will handle fields that are separated by any whitespace (any number of blanks, horzontal tabs, new lines, carriage returns, vertical tabs, form feeds). On output, newlines are generated after fields.

### Devices

`/dev/streams/wans`
`/dev/wans` The Style 2 CDI device for WAN, `wans(4)`.

### Files

`/etc/sysconfig/openss7/template/`*filename*
The default directory location for configuration files used by this command.[2]

### See Also

Section D.2 [WAN Tuning File Format], page 70.

### Bugs

`wantune` has no known bugs.

### Compatibility

The `wantune` command is compatible with *Spider X.25*, and implementations based on *Spider X.25*, such as *AIXlink/X.25*, *HP-UX*, *IRIS SX.25*, *PT X.25*, *RadiSys WAN*, *SBE X.25*, *Solstice X.25*, and others, with the following portability considerations:

- A version of this command is provided by *OpenSS7* for compatibility with systems that require it. Neither this command nor the `xnetd(8)` are recommended for configuration of the *OpenSS7* subsystems. Use the SNMP agent instead.
- Options `-e`, `-n`, `-q`, `-v`, `-h`, `-V`, `-C`, and all long options, are specific to this *OpenSS7* implementation of `wantune` and will not be used by portable command scripts.

[1] Note that the precise location of the `/etc/sysconfig` directory varies depending upon whether the build was on a `dpkg(1)`-based or `rpm(1)`-based system.
[2] Note that the precise location of the `/etc/sysconfig` directory varies depending upon whether the build was on a `dpkg(1)`-based or `rpm(1)`-based system.

- No other implementation documents printing the output to a file when a *filename* is specified with the `-G` command option. This is an enhancement of this implementation.
- No other implementation documents the `-e`, `-n`, `-q`, `-v`, `-h`, `-V`, and `-C`, options. They will not be used by portable command scripts.
- Options `--help` and `--version` are provided for compatibility with GNU coding standards (GNITS); `--copying`, OpenSS7 coding standards.
- `wantune` attempts to be source (and script) compatible with historical implementations based on *Spider X.25*, however, binary compatibility is not attempted. Any binary compatibility acheived is likely to be removed in a subsequent release.

For additional compatibilty considerations, see Appendix E [WAN Compatibility and Porting], page 73.

### Conformance

*AIXlink/X.25*, *HP-UX*, *IRIS SX.25*, *PT X.25*, *RadiSys WAN*, *SBE X.25*, *Solstice X.25*, documentation. See [References], page 77.

### History

`wantune` first appeared in *Spider X.25*.

## C.2 WAN Address Mapping Utility

### Name

`wanmap` – manage WAN address mappings

### Synopsis

```
wanmap [options] -D -s subnet -r remote [-d device]
wanmap [options] -G -s subnet -r remote [-d device] [filename]
wanmap [options] -M -s subnet [-d device] [filename]
wanmap [options] -P -s subnet [-d device] [filename]
wanmap [options] -Z -s subnet [-d device] [filename]
wanmap {-h|--help}
wanmap {-V|--version}
wanmap {-C|--copying}
```

### Description

`wanmap` provides a user space command line program that permits alteration and management of the remote to interface address mapping tables that are associated with a given *subnet* identifier within the WAN driver. Command options are given to permit the deletion of individual entries, the retrieval of individual entries, the loading of the table from a file for a given *subnet*, and zeroing of the table for a given *subnet*.

### Options

The `wanmap` command accepts the following options:

### Command Options

The following command options are mutually exclusive: only one command option should be present on the command line at a time. The exceptions are the `-h`, `-V` and `-C` options that can be specified alone, or with any other option.

`-D, --delete`
: Deelte the address mapping identified by the *remote* argument to the `-r` option and the *subnet* argument to the `-s` option. The `-s` and `-r` options must be specified.

`-G, --get`
: Display the address mapping identified by the *remote* argument to the `-r` option and the *subnet* argument to the `-s` option. The `-s` and `-r` options must be specified.

`-M, --list`
: Display the address mappings identified by the *subnet* argument to the `-s` option. The `-s` option must be specified.

`-P, --load`
: Load the address mappings identified by the *subnet* argument to the `-s` option. The `-s` option must be specified.

`-Z, --zero`
: Delete all address mappings identified by the *subnet* argument to the `-s` option. The `-s` option must be specified.

`-h, --help`
: When this option is encountered, display usage information to `stdout`, stop options processing, and exit without taking further action.

`-V, --version`
: When this option is encountered, display version information to `stdout`, stop options processing, and exit without taking further action.

`-C, --copying`
: When this option is encountered, display copying information to `stdout`, stop options processing, and exit without taking further action.

### Non-Command Options

The following common options can be specified together along with a command option. It is not an error to specify options that are not necessary for the command option with which they are specified.

`-r, --remote` *remote*
: Specify the remote address, *remote*, for which to delete or retrieve an address mapping. The address mapping is deleted when the `-D` command option is given; retrieved for the `-G` option. This option must be specified whenever the `-D` or `-G` option is specified.

`-s, --subnet` *subnet*
: Specifies the subnetwork identifier, *subnet*, to which the command applies. This option must be specified whenever the `-D`, `-G`, `-M`, `-P` or `-Z` options is specified.

`-d, --device` *device*
: Specify the device name, *device*, upon which to operate. When this option is not specified, or *device* is not given, the default is `/dev/wans`, `/dev/streams/clone/wans` or `/dev/streams/wans/0`, whichever opens successfully first.

`-f, --file` *filename*
: Specify the configuraiton file name that holds mapping information to apply to the device when the `-P` command option is also specified. When this option is not specified, or the *filename* is not given, the default is `/etc/sysconfig/wanmapconf`.[1] For the format of this file, see Section D.1 [WAN Mapping File Format], page 69.

`-q, --quiet [`*level*`]`
: Suppress normal output. Only the return code oof the command is of interest. This has the same effect as '`--verbose=0`'.

`--debug [`*level*`]`
: Specify or increase the debugging verbosity *level*. Sets the debugging verbosity *level*, when given, or simply incresases the debug verbosity when *level* is not given. This option can be repeated. When *level* is specified, only the last repetition takes effect.

`-v, --verbose [`*level*`]`
: Specify or increase the output verbosity *level*. Sets the output verbosity *level*, when given, or simply incresases the output verbosity when *level* is not given. This option can be repeated. When *level* is specified, only the last repetition takes effect.

### Arguments

The `wanmap` command takes no non-option arguments.

### Diagnostics

An exit status of zero (0) indicates that the command was successful; one (1) indicates that an error occured and a diagnostic message is printed to `stderr`; two (2) indicates that the option or argument syntax was in error and a diagnostic message is printed to `stderr`.
The `--quiet` option suppresses the printing of normal output to `stdout` and diagnostic messages to `stderr`.

[1] Note that the precise location of the `/etc/sysconfig` directory varies depending upon whether the build was on a `dpkg(1)`-based or `rpm(1)`-based system.

### File Format

For the input file format, see Section D.1 [WAN Mapping File Format], page 69.

### Notices

On input, this implementation will handle fields that are separated by any whitespace (any number of blanks, horzontal tabs, new lines, carriage returns, vertical tabs, form feeds). On output, newlines are generated after fields.

### Devices

`/dev/streams/wans`
`/dev/wans` The Style 2 CDI device for WAN, `wans(4)`.

### Files

`/etc/sysconfig/openss7/wanmapconf`
The default directory location for configuration files used by this command.[2]

### See Also

Section D.1 [WAN Mapping File Format], page 69.

### Bugs

`wanmap` has no known bugs.

### Compatibility

`wanmap` is compatible with *Spider X.25*, and implementations based on *Spider X.25*, such as, *AIXlink/X.25*, *HP-UX*, *IRIS SX.25*, *PT X.25*, *RadiSyS WAN*, *SBE X.25*, *Solstice X.25*, and others, with the following portability considerations:

- Options `-q`, `-v`, `-h`, `-V`, `-C`, and all long options, are specified to this *OpenSS7* implementation of `wanmap` and should not be used by portable command scripts.
- No other implementation documents the `-q`, `-v`, `-h`, `-V`, and `-C`, options. They should not be used by portable command scripts.
- Options `--help` and `--version` are provided for compatibility with GNU coding standards (GNITS); `--coying`, OpenSS7 coding standards.

For additiona compatibility information, see Appendix E [WAN Compatibility and Porting], page 73.

### Conformance

*AIXlink/X.25*, *HP-UX*, *IRIS SX.25*, *PT X.25*, *RadiSyS WAN*, *SBE X.25*, *Solstice X.25*, documentation.

### History

`wanmap` first appeared in *Spider X.25*.

[2] Note that the precise location of the `/etc/sysconfig` directory varies depending upon whether the build was on a `dpkg(1)`-based or `rpm(1)`-based system.

# Appendix D WAN File Formats

## D.1 WAN Mapping File Format

The WAN mapping file format first appeared in *Spider X.25*.

## D.2 WAN Tuning File Format

### File Format

The WAN tuning file format corresponds closely to the `wan_tnioc` structure. Each line in the file typically corresponds to a member in the `wan_tnioc` structure. See Section 5.1.2.1 [wan˙tnioc Structure], page 26.

The file consists of 12 lines of data as follows:

1. *WAN_maxframe* specifies the maximum frame size for the WAN interface in octets. The value is a positive integer.
2. *WAN_baud* specifies the baud rate for the WAN interface in bits per second. When zero (0), an external clock must be provided. THe value is a positive integer.
3. *WAN_translate* specifies whether a remote address should be translated into an interface address using the address mapping function. See `wanmap(8)` for more information. When this value is '`Y`', '`y`' or '`1`', the address mapping will be used to translate the remote address.
4. *WAN_phys_int* specifies the physical interface type. This can be the integer numeric value '`0`', '`1`' or '`2`', wehre the interpretation of these values is described in the table below:

   | | | |
   |---|---|---|
   | 0 | `WAN_X21` | X.21 physical interface. |
   | 1 | `WAN_V28` | V.28 physical interface. |
   | 2 | `WAN_V35` | V.35 physical interface. |

5. *WAN_connect_proc* specifies the calling procedures to be used when generating outgoing calls on the WAN interface. This can be the values '`0`', '`1`', or '`2`', where the interpretation of these value is described in the table below:

   | | | |
   |---|---|---|
   | 0 | `WAN_NONE` | No calling procedures. |
   | 1 | `WAN_X21P` | X.21 calling procedures. |
   | 2 | `WAN_V25bis` | V.25 bis calling procedures. |

6. *WAN_x21_T1* specifies the time interval for the X.21 T1 Timer: the amount of time that the DTE will await *proceed-to-select* having signalled *call-request* to the DCE. The value is a short integer number of deciseconds (0.1 seconds), with a default value of 30 (3.0 seconds).
7. *WAN_x21_T2* specifies the time interval for the X.21 T2 Timer: the amount of time that the DTE wil await *ready-for-data* having signalled *end-of-selection*. The value is a short integer number of deciseconds (0.1 seconds), with a default value of 200 (20.0 seconds).
8. *WAN_x21_T3A* specifies the time interval for the X.21 T3A Timer: the amount of time that the DTE wil await additional *call-progress* or *DCE-provided-information* signals. The value is a short integer number of deciseconds (0.1 seconds), with a default value of 60 (6.0 seconds).
9. *WAN_x21_T4B* specifies the time interval for the X.21 T4B Timer: the amount of time that the DTE wil await *ready-for-data* having signalled *call-accept*. The value is a short integer number of deciseconds (0.1 seconds), with a default value of 60 (6.0 seconds).
10. *WAN_x21_T5* specifies the time interval for the X.21 T5 Timer: the amount of time that the DTE wil await *DCE-ready* having signalled *DTE-clear-request*. The value is a short integer number of deciseconds (0.1 seconds), with a default value of 20 (2.0 seconds).
11. *WAN_x21_T6* specifies the time interval for the X.21 T6 Timer: the amount of time that the DTE wil await *DCE-ready* having signalled *DTE-clear-confirm*. The value is a short integer number of deciseconds (0.1 seconds), with a default value of 20 (2.0 seconds).
12. *WAN_v25_callreq* specifies the time interval for the V.25 T1 Timer: the amount of time that the DTE will await successful call establishment afeter having initiated a call. The value is a

short integer number of deciseconds (0.1 seconds), with a default value of 6000 (600 seconds, or 5 minutes).

Note that V.25 Timer T1 is only necessary when CFI (Call Failure Indication) is not provided by the network.

# Appendix E WAN Compatibility and Porting

The typical *SpiderX.25* stack implementation is illustrated in Figure E.1. This stack profile has the following characteristics:

- The predominant implementation interface at the network layer is the Network Layer Interface (NLI), see [NLI], page 78.
- The predominant implementation interface at the data link layer is the Data Link Provider Interface (DLPI), see [DLPI], page 77.
- The predominant implementation interface at the media access or frame layer is the MAC interface for LAN and the WAN interface (for WAN), the later being the subject of this specification.

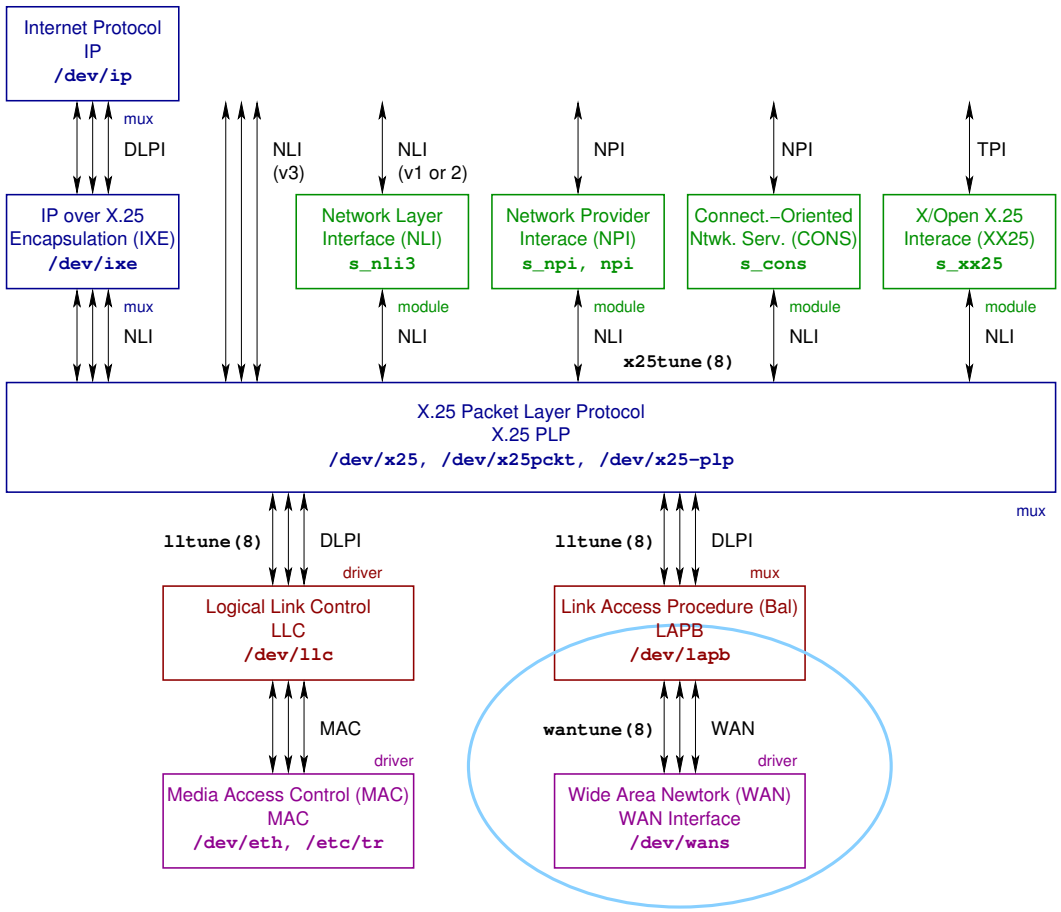


Figure E.1: *SpiderX.25 Overview*

The *SpiderX.25* stack implementation differs from the *OpenSS7* one in several fundamental ways:

- *OpenSS7* uses the Network Provider Interface (NPI), see [NPI], page 78, at the network layer.

- *OpenSS7* uses the Data Link Provider Interface (DLPI), see [DLPI], page 77, at the data link layer.
- *OpenSS7* uses the Communications Device Interface (CDI), see [CDI], page 77, at the media access or frame sub-layer.

For the purposes of providing compatibility between the *OpenSS7* implementation approach and the *SpiderX.25* implementation approach, *OpenSS7* provides a number of pushable "conversion" modules. See Appendix B [WAN Drivers and Modules], page 57.

## E.1 Compatibility with AIXlink/X.25

## E.2 Compatibility with HP X.25/9000

## E.3 Compatibility with IRIS SX.25

## E.4 Compatibility with PT X.25

## E.5 Compatibility with RadiSys WAN

## E.6 Compatibility with SBE X.25

## E.7 Compatibility with Solstice X.25

# Appendix F Glossary of WAN Terms and Acronyms

| | |
|---|---|
| *ANSI* | American National Standards Institute |
| *CCITT* | Old name for ITU-T |
| *CONS* | Connection-Oriented Network Service |
| *ENSDU* | Expedited Network Service Data Unit |
| *ETSI* | European Telecommunications Standards Institute |
| *IEEE* | Institute of Electrical and Electronics Engineers |
| *ITU* | International Telecommunications Union |
| *ITU-T* | ITU Telecom Sector |
| *LCI* | Logical Channel Identifier |
| *LLC1* | Logical Link Control Type 1 |
| *LLC2* | Logical Link Control Type 2 |
| *LLC* | Logical Link Control |
| *MAC* | Media Access Control |
| *NLI* | Network Layer Interface |
| *NPDU* | Network Protocol Data Unit |
| *NSAP* | Network Service Access Point |
| *NSDU* | Network Service Data Unit |
| *NSP* | Network Service Provider |
| *NS* | Network Service |
| *NSU* | Network Service User |
| *PDU* | Protocol Data Unit |
| *PVC* | Permanent Virtual Circuit |
| *SAP* | Service Access Point |
| *SDU* | Service Data Unit |
| *VC* | Virtual Circuit |
| *X.121* | ITU-T Recommendation X.121 |
| *X.25* | ITU-T Recommendation X.25 |
| *X.29* | ITU-T Recommendation X.29 |

# References


[AIXlink/X.25] *AIXlink/X.25 Version 2.1 for AIX: Guide and Reference*, No: SC23-2520-07, Eighth Edition, September 2006, (Bolder, CO), International Business Machine Corp., IBM. IBM Documentation Library.

[ARTIC WAN] *ARTIC STREAMS Support WAN Driver Interface Reference*, Release 1.7, June 2004, (Hillsboro, OR), RadiSys Corporation, RadiSys. [Doc No: 007-01232-0003], RadiSys Support Documentation.

[CDI] *OpenSS7 CAE Specification: Communications Device Interface (CDI) Specification*, Revision 0.9.2, Draft 2, July 15, 2007, (Edmonton, Canada), B. Bidulock, OpenSS7 Corporation. Distributed with package `strxns-0.9.2` and `openss7-0.9.2`. OpenSS7 Documents.

[DLPI] *Open Group CAE Specification: Data Link Provider Interface (DLPI) Specification*, Revision 2.0.0, Draft 2, August 20, 1992, (Parsippany, New Jersey), UNIX International, Inc., UNIX International Press. The Open Group, The OpenSS7 Project.

[IRIS SX.25] *IRIS SX.25 NLI Programmer's Guide*, 1995, (Mountainview, CA), Silicon Graphics, Inc., SGI Technical Publications. [No: 007-2268-002]. SGI Technical Publications.

[ISO3309] **ISO/IEC 3309:1988**, *Information Technology – Data Communications – High-Level Data Link Control procedures – Frame Structure*, 1988, ISO/IEC, International Orgianization for Standardization. International Organization for Standardization.

[ISO7776] **ISO/IEC 7776:1995**, *Information technology — Telecommunications information exchange between systems — High-level data link control procedures — Description of the X.25 LAPB-compatible DTE data link procedures*, Second Edition, July 1, 1995, International Organization for Standardization. International Organization for Standardization.

[ISO8208] **ISO/IEC 8208:2000**, *Information Technology — Data Communications — X.25 Packet Layer Protocol for Data Terminal Equipment*, Fourth Edition, November 1, 2000, (Geneva), ISO/IEC, International Organization for Standardization. International Organization for Standardization.

[ISO8802-2] **ANSI/IEEE Standard 802.2-1998 [ISO/IEC 8802-2:1998]**, *IEEE Standard for Information Technology — Telecommunications and Information Exchange Between Systems — Local and Metropolitan Area Networks — Specific Requirements — Part 2: Logical Link Control*, May 7, 1998, (New York), ANSI/IEEE, IEEE Computer Society. [ISBN 1-55937-959-6]. Institute of Electrical and Electronics Engineers.

[ISO8881] **ISO/IEC 8881:1989**, *Information Processing Systems — Data Communications — User of the X.25 Packet Level Protocol in Local Area Networks*, 1989, ISO/IEC, International Organization for Standardization. International Organization for Standardization.

[NLI] *OpenSS7 CAE Specification: Network Layer Interface (NLI) Specification*, Revision 0.9.2, Draft 1, June 2008, (Edmonton, AB), B. Bidulock, OpenSS7 Corporation. Distributed with package `strx25-0.9.2` and `openss7-0.9.2`. The OpenSS7 Project.

[NPI] *Open Group CAE Specification: Network Provider Interface (NPI) Specification*, Revision 2.0.0, Draft 2, August 17, 1992, (Parisppany, New Jersey), UNIX International, Inc., UNIX International Press. The OpenSS7 Project.

[Solstice X.25] *Solstice X.25 9.2 Administration Guide*, October 1999, (Palo Alto, CA), Sun Microsystems, Inc., Sun. [Part No: 806-1234-10], Solaris Documentation.

[TPI] *Open Group CAE Specification: Transport Provider Interface (TPI) Specification*, Revision 2.0.0, Draft 2, 1999, (Berkshire, UK), Open Group, Open Group Publication. The Open Group, The OpenSS7 Project.

[V.25 bis] **ITU-T Recommendation V.25 bis (10/96)**, *Synchronous and asyncrhonous automatic dialing procedrues on switched networks*, October 1996, (Geneva), ITU, ITU-T Telecommunication Standardization Sector of ITU, (Previously "CCITT Recommendation"), `http://www.itu.int/rec/T-REC-V.25bis/en/T-REC-V.25bis`.

[X.21] **ITU-T Recommendation X.21 (09/92)**, *Interface between Data Terminal Equipment (DTE) and Data Circuit-terminating Equiment (DCE) for synchronous operation on Public Data Networks*, September 1992, (Geneva), ITU, ITU-T Telecommunication Standardization Sector of ITU. (Previously "CCITT Recommendation"), T-REC-X.21.

[X.21 bis] **ITU-T Recommendation X.21 bis (03/88)**, *Use on Public Data Networks of Data Terminal Equipment (DTE) which is designed for interfacing to synchronous V-series modems*, March 1988, (Geneva), ITU, ITU-T Telecommunication Standardization Sector of ITU. (Previously "CCITT Recommendation"), T-REC-X.21bis.

[X.25] **ITU-T Recommendation X.25**. T-REC-X.25.

[X.75] **ITU-T Recommendation X.75**. T-REC-X.75.

[X.29] **ITU-T Recommendation X.29**. T-REC-X.29.

[XX25] *X/Open CAE Specification: X.25 Programming Interface using XTI (XX25)*, No. c411, November 1995, (Berkshire, UK), X/Open, Open Group Publication. [ISBN: 1-85912-136-5]. The Open Group.

# Licenses

All code presented in this manual is licensed under the [GNU Affero General Public License], page 79. The text of this manual is licensed under the [GNU Free Documentation License], page 89, with no invariant sections, no front-cover texts and no back-cover texts. Please note, however, that it is just plain wrong to modify statements of, or attribute statements to, the Author or *OpenSS7 Corporation*.

## GNU Affero General Public License

The GNU Affero General Public License.
Version 3, 19 November 2007


### Preamble

The GNU Affero General Public License is a free, copyleft license for software and other kinds of works, specifically designed to ensure cooperation with the community in the case of network server software.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, our General Public Licenses are intended to guarantee your freedom to share and change all versions of a program–to make sure it remains free software for all its users.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

Developers that use our General Public Licenses protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License which gives you legal permission to copy, distribute and/or modify the software.

A secondary benefit of defending all users' freedom is that improvements made in alternate versions of the program, if they receive widespread use, become available for other developers to incorporate. Many developers of free software are heartened and encouraged by the resulting cooperation. However, in the case of software used on network servers, this result may fail to come about. The GNU General Public License permits making a modified version and letting the public access it on a server without ever releasing its source code to the public.

The GNU Affero General Public License is designed specifically to ensure that, in such cases, the modified source code becomes available to the community. It requires the operator of a network server to provide the source code of the modified version running there to the users of that server. Therefore, public use of a modified version, on a publicly accessible server, gives the public access to the source code of the modified version.

An older license, called the Affero General Public License and published by Affero, was designed to accomplish similar goals. This is a different license, not a version of the Affero GPL, but Affero has released a new version of the Affero GPL which permits relicensing under this license.

The precise terms and conditions for copying, distribution and modification follow.

## Terms and Conditions

0. Definitions.

   "This License" refers to version 3 of the GNU Affero General Public License.

   "Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

   "The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

   To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

   A "covered work" means either the unmodified Program or a work based on the Program.

   To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

   To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

   An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

   The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

   A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

   The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

   The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the

source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

   a. The work must carry prominent notices stating that you modified it, and giving a relevant date.

b. The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

c. You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d. If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a. Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b. Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c. Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d. Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e. Convey the object code using peer-to-peer transmission, provided you inform other peers

where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a. Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b. Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c. Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d. Limiting the use for publicity purposes of names of licensors or authors of the material; or

e. Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f. Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

   You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

    Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

    An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

    You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

    A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

    A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

    Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

    In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

    If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to

downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

    If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Remote Network Interaction; Use with the GNU General Public License.

    Notwithstanding any other provision of this License, if you modify the Program, your modified version must prominently offer all users interacting with it remotely through a network (if your version supports such interaction) an opportunity to receive the Corresponding Source of your version by providing access to the Corresponding Source from a network server at no charge, through some standard or customary means of facilitating copying of software. This Corresponding Source shall include the Corresponding Source for any work covered by version 3 of the GNU General Public License that is incorporated pursuant to the following paragraph.

    Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the work with which it is combined will remain governed by version 3 of the GNU General Public License.

14. Revised Versions of this License.

    The Free Software Foundation may publish revised and/or new versions of the GNU Affero General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU Affero General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU Affero General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU Affero General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

**THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.**

16. Limitation of Liability.

**IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.**

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

**END OF TERMS AND CONDITIONS**

### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
one line to give the program's name and a brief idea of what it does.
Copyright (C) year name of author

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU Affero General Public License as published by
the Free Software Foundation, either version 3 of the License, or (at
your option) any later version.

This program is distributed in the hope that it will be useful, but
WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
Affero General Public License for more details.

You should have received a copy of the GNU Affero General Public License
along with this program.  If not, see http://www.gnu.org/licenses/.
```

Also add information on how to contact you by electronic and paper mail.

If your software can interact with users remotely through a network, you should also make sure that it provides a way for users to get its source. For example, if your program is a web application, its interface could display a "Source" link that leads users to an archive of the code. There are many ways you could offer source, and different solutions will be better for different programs; see section 13 for the specific requirements.

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU AGPL, see http://www.gnu.org/licenses/.

## GNU Free Documentation License

GNU FREE DOCUMENTATION LICENSE
Version 1.3, 3 November 2008



0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document *free* in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

   If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

   If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

   If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

   It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

   You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

   A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

   B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

   C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

   D. Preserve all the copyright notices of the Document.

   E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

   F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

   You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

   You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

   A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

   If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

   Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

   If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

   You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

## ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C)  year  your name.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.3
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover
Texts.  A copy of the license is included in the section entitled ``GNU
Free Documentation License''.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with. . . Texts." line with this:

```
with the Invariant Sections being list their titles, with
the Front-Cover Texts being list, and with the Back-Cover Texts
being list.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

# Index

## S



## T



## W

## X