Bull DPX/20

OSI Communications Porting Guide

AIX

ORDER REFERENCE
86 A2 44AP 03

Bull DPX/20

OSI Communications Porting Guide

AIX

Software

June 1996

Bull Electronics Angers S.A.
CEDOC
Atelier de Reprographie
331 Avenue Patton
49004 ANGERS CEDEX 01
FRANCE

ORDER REFERENCE
86 A2 44AP 03

Printed in France

Suggestions and criticisms concerning the form, content, and presentation of this book are invited. A form is provided at the end of this book for this purpose.

Trademarks and Acknowledgements

We acknowledge the right of proprietors of trademarks mentioned in this book.

AIX® is a registered trademark of International Business Machines Corporation, and is being used under licence.

UNIX is a registered trademark in the USA and other countries licensed exclusively through X/Open.

The information in this document is subject to change without notice. Groupe Bull will not be liable for errors contained herein, or for incidental or consequential damages in connection with the use of this material.

About this Book

This book has three objects:

- describe the new functions and limits of OSI communications products associated with AIX Version 4.1.4 and their differences compared to previous versions (AIX Version 3.2.5).
- describe the process of software product migration, i.e.

  packaging: how the various OSI communications products are delivered,

  software product migration: how to install those products with preserving configuration information.
- describe how to migrate user applications: user interfaces changes are described for each of OSI products.

Warning:

The expression ”**OSI communications products**” used all along this document refers to these software products:

- **OSI Stack** low and high layers, and **CIS**,
- APIs, **MAX25-3**, **XTI**, **Session API**, **XAP** and **ROSE**,
- Applications, **TPAD-HPAD**, **FTAM** and **UFT**, **OTM**, **CPI-C OSI**, **CMIP** agent, **X500**,
- Miscellaneous products, **bullx25**, **NetShare** and **ASN.1**.

It covers more than OSI communications, as some products, such as bullx25 or XTI, may be associated with TCP/IP.

OSI components as OSI-TP are not mentioned as they were not available previously. Though it was not available previously, XAP is mentioned because it comes in place of ACSE/P API.

Warning:

AIX Version 3.2.5 and AIX Version 4.1.4 are related to previous and new versions of OSI stack:

- AIX Version 3.2.5 means 1. 2.17 OSI Stack version number and later.
- AIX Version 4.1.4 means 02.03.05 OSI Stack version number and later.

Figure 1 page 3-2 gives OSI Stack and associated products version numbers for AIX Version 3.2.5 and AIX Version 4.1.4.

Who Should Use this Book

This guide is intended for system administrators and programmers who have experience using previous versions of OSI products (AIX Version 3.2.5) and wish to migrate their installation and applications to the new OSI products available with AIX Version 4.1.4.

Consequently, knowledge of OSI communications products and terminology, installation procedures, and the use of the SMIT tool is assumed to be acquired.

Note: This guide is of no interest for anyone who just begins to work with OSI communications products. For each product, reference documentation is available. Refer to **Bibliography**, on page A-1.

How to Use this Book

Overview of Contents

This guide contains the following chapters and appendixes:

Chapter 1. **OSI Communications New Functions and Limits**
Describes functional improvements and additions to OSI products, and changes in limits (such as data unit sizes and number of connections).

Chapter 2. **OSI Communications Packaging**
Describes the delivery media, prerequisites, and organization of OSI communication products.

Chapter 3. **OSI Communications Migration**
Describes how to migrate previous OSI products to new versions, along with configuration information when applicable.

Chapter 4. **OSI Communications User Interface Changes**
Describes the commands, files and subroutines changes for each of the OSI products.

Appendix A. **Bibliography**

Glossary

Index

Highlighting

The following highlighting conventions are used in this book:

Bold Identifies commands, subroutines, keywords, files, structures, directories, and other items whose names are predefined by the system. Also identifies graphical objects such as buttons, labels, and icons that the user selects.

Italics Identifies parameters whose actual names or values are to be supplied by the user.

`Monospace` Identifies examples of specific data values, examples of text similar to what you might see displayed, examples of portions of program code similar to what you might write as a programmer, messages from the system, or information you should actually type.

Conventions and Terminology

The following terms are used throughout this guide as follows:

DPX/20 Refers to DPX/20 mono-processor

ESCALA or DPX/20 ESCALA, refers to DPX/20 multi-processor

machine Refers to DPX/20, mono or multi-processor

operating system
Refers to the AIX operating system.

API Application Programming Interface.
Other manuals may use the terms “library”, or more generally “interface”.

stack Short name for “OSI stack”, or “OSI communications stack“. Refers to the core layers of communications services used by all OSI products.
The previous version refers to AIX version 3.2.5 and the new version to AIX Version 4.1.4.

When comparing differences, a command, subroutine, or file for AIX version 3.2.5 is simply called a **previous version** and a version for AIX Version 4.1.4 is called a **new version**.

Table of Contents

Chapter 1. New Functions and Limits

This chapter describes the functional improvements and additions to OSI products, and changes in limits (such as data unit sizes and number of connections).

- Differences Common to OSI Communications Products on page 1-1
- OSI Stack on page 1-3
- OSI APIs: MAX25-3, XTI, Session API, XAP and ROSE on page 1-5
- OSI Applications: TPAD-HPAD, OTM, CPI-C OSI, UFT, FTAM, X500 and CMIP Agent on page 1-9
- OSI Miscellaneous Products: bullx25, NetShare and ASN.1 on page 1-12

OSI Communications Products – Common Differences

Distribution Media and Licensing

AIX Version 3.2.5

Each OSI product was delivered on a specific media. Licensing was associated with purchase of this media.

AIX Version 4.1.4

Except for XAP, ROSE and Session API, the OSI communications products and associated SRBs are delivered on a single CD-ROM. They can be installed and configured on any machine, on which AIX Version 4.1.4 is installed.

Use of each product or set of products is subsequently controlled by a license.

Licensing is managed using iFOR/LS (information For Operation Retrieval/License System). Each iFOR/LS license has an encrypted key, a license type (Nodelocked for the OSI communications products) and an expire date.

Refer to:

- *AIX iFOR/LS System Management Guide*, for details about licensing and iFOR/LS,
- the SRB specific to the product for details on how the product uses the license key.

Note: XAP and the associated SRB is available on a specific CD-ROM, as well as for ROSE and Session API which are delivered together with the associated SRBs on a specific CD-ROM. Licensing is associated with purchase of the corresponding media.

osisnap Command

The **osisnap** command gathers information needed when reporting a problem to Technical Support. It is helpful for error diagnostics and troubleshooting.

Previously, it only collected stack information. Now, **osisnap** can collect stack as well as OSI APIs or applications information. For instance XTI and NetShare problems can be reported using **osisnap**.

On a system, the **–l** option of **osisnap** allows to list the OSI products which can be snapped through **osisnap**.

Refer to *OSI Diagnostic Interactive Toolkit (ODIT) User's Guide* for a complete information about **osisnap**.

CIS (Connection Information Service)

CIS manages a local database, called the **CIS** Local Database and accessed by OSI components such as XAP, FTAM, OSI–VT, etc. This database uses the AIX Object Data Manager (ODM) to store information needed at connection time:

- Addressing information (PSEL, SSEL, TSEL NSAP or SNPA)
- Connection profiles (Network type, Negotiation profile, ...)

In this version, **CIS** is a package associated with the OSI high layers and useful for XAP, FTAM, CMIP agent, X500, MMS.

CIS is accessed through either a configurator based on SMIT or commands. Refer to *Connection Information Service (CIS) Installation, Administration & User's Guide* for further information.

BASIC-COM and GCOS-COM Easy Configuration

Easy Configuration provides an easy way to configure both bundles while maintaining the access to the specialized and detailed configurator dedicated to each product.

It configures:

- X25 lines, and public X25 networks,
- TCP/IP over a LAN
- TCP/IP over X25 networks,
- the OSI stack,
- NetShare,
- TPAD–HPAD,
- UFT,
- OTM,
- CPI–C SS.

MAX3, XTI/XX25 and PAD printing are not configured by the Easy Configurator.

Individual product configurators can reuse existing parameters previously set by the Easy Configurator.

Note: 1. Configuration performed via the easy configurator is seen by the specialized configurators.
2. Configuration performed using the specialized configurators is not seen by the easy configurator. In particular, avoid using the easy configurator after having configured with the specialized ones.

OSI Stack

The OSI Stack for AIX Version 4.1 is upwardly–compatible with the previous version on AIX 3.2.5, i.e. informations about previous configuration are preserved, and APIs are generally not changed. The coexistence of several OSI Stack versions on the same machine is not possible, except in the case of a server of diskless stations, which may have several SPOT versions.

Improved Stack services result from a number of changes to its structure and user interface:

- Architecture
- Configuration
- Help on–line
- Addressing
- New features
- Service tools
- Maintainability framework
- Resource limits

Architecture

Designed to take advantage of multi–processor architecture, with modular concept and clearly–defined standardized upper layer STREAM interfaces. Lower layers are integrated into a STREAM driver providing a TPI interface.This STREAM driver contains a set of modules, which implement all the protocols below the transport interface. Components running in the MAD environment make use of a new memory manager supporting a dynamic memory allocation. These changes result in improved throughput of data.

Configuration

A simplified SMIT user interface permits easier configuration. The compilation and generation phases are no longer needed. Extensive use is made of parameter default values, including the provision of a minimum configuration menu. Dynamic configuration is provided and parameters, stored in databases, maintain the latest configuration across system reboots. Static RIB entries are displayed in detail and can now be changed dynamically without stopping communications. All the new features are configurable through SMIT.

Help On–line

Contextual help under SMIT is extended to all the menus.

Addressing

At Application level, addresses are defined using AE Titles, made up of an AP Title and an AE Qualifier. Coded using ASN.1 rules, both Form 1 and Form 2 AE Titles are supported. Session Selectors are fully used at Presentation, Session and Transport levels. Hierarchical addressing is used down to the Network level, where the network address is defined as a set of NSAPs.

For incoming Service Access Point (SAPs), addresses can be partially specified using a wildcarding feature, extending to all high level selectors and to NSAPs.

Application behavior can be affected by these changes. See **Addressing** in the *OSI Services Reference Manual*.

New features

The routing capability of OSI stack version for AIX Version 3.2.5 was limited to CLNP relay. The new version provides also two new transport relays:

- MSDSG relay, which conforms to ISO TR 10172 and is reserved for NSAP addressing,
- a transport relay, which supports all the addressing modes of OSI stack, but which is not transparent to user from an addressing point of view.

All the relays can coexist on a same system. OSI stack on AIX Version 4.1.4 introduces filtering capabilities for incoming connections or data packets. It allows to define lists of authorized calling/called addresses.

CLTP protocol is now supported, as well as special multicast features for TCP/netbios support.

XTI over CLTP is supported.

Dynamic Adding/Removing of communications adapters is supported.

Service Tools

In addition to configuration tools, new service tools are provided:

- Stack Applications Support (SAS), for disciplined starting and stopping of OSI Applications when the stack is loaded and unloaded,
- Problem determination, with enhancement of trace management and Stack status display, in the ODIT Toolkit. For details, see the *OSI Diagnostic Interactive Toolkit (ODIT) User's Guide*.

Errlog is also used for problem reporting.

Maintainability Framework

This new SMIT menu permits to easily diagnose and validate OSI communication products, concerning both installation and configuration. See *Communications Maintenance Tools* for details.

Resource Limits

The new OSI Stack has more generous limits for both the resources and addressing used. The limits are not intrinsic to the stack, but vary according to available system resources. Most fixed limits are controlled by the stack configurator. The following table compares some resource and addressing limits for the stacks on previous version and new version:

Stack Resource Limit Comparison		
Limit	**AIX 3.2 Stack**	**AIX 4.1 Stack**
communications adapters		
FDDI	4	8
Ethernet	4	8
Token Ring	4	8
HiSpeed WAN Comm. (X25)	4	6
connections		
Session	512	2048
Transport	512	2048
Permanent Virtual Circuits (PVCs)	8 per line (224 max.)	1024 per line
data		
allocation method for unlimited main memory	static	dynamic
SSDU size	32 KB – 1	64 KB – 1
TSDU size (through session)	4 KB	8000 bytes

OSI APIs

An OSI API provides access to a specific OSI stack level and relative services. It allows, for application needs to bypass all stack layers above the required services so, avoiding the overhead of unneeded higher stack levels, if not needed.

MAX25-3	Access to X.25-3 services.
XTI	X/Open Transport Interface with Bull enhancements, a transport layer API.
XX25	X/Open access to X25.3 services using XTI interface.
Session API	Access to Connection Oriented Session Protocol (COSP).
XAP	X/Open ACSE/Presentation Services API.
ROSE	Remote Operations Service Element.

The APIs offer improved trace services using the AIX trace system. Except for XTI and XX25, **ODIT** can be used to configure API trace levels, start and stop tracing, and generate trace reports.

MAX25-3

IPS accesses X.25 services, not through a specific library, but using the **MAD** driver and the **X.25-3 Access Method.** The driver itself has changed, but most of these changes are not externally visible. Changes in the MAX25-3 interface include:

- new **errno** values.
- the destination unit field of the UCB (the **tamd** field of the **Tucb** structure (**max3.h**) and the **un_dest** field of the **ucb_util** structure (**maduif.h**)) is now a reserved field.
- some error codes in the interface file **maduif.h** are no longer used but have been retained to maintain previous source code compatibility.
- improved dispatch of incoming calls, switching now takes into account the called addresses and subaddresses.
- the maximum number of MAX25-3 connection contexts is now dynamically determined using OSI configuration information: MAX25-3 calculates the maximum number of virtual circuits (SVCs and PVCs) for all X25 lines, and adds 10 additional contexts for each X25 line to handle the waiting of incoming calls. Previously, the maximum number of connections was a static limit, depending on the MAX25-3 version (256 connections for versions up to and including 1.2.12; 4096 connections for version 1.2.16 and later).
- thread-safe.

XTI (including XX25)

In this document, whenever **XTI** is used without further qualification, it refers to Bull-enhanced XTI.

XTI accesses the services of the following transport or network providers:

- OSI with Connection-Oriented mode of service (**COTS**),
- OSI with ConnectionLess mode of service (**CLTS**),
- Internet with Connection-Oriented mode of service (**TCP/IP**) or Connectionless mode of service (**UDP/IP**),
- OSI onto TCP/IP (**NetShare (RFC 1006)**),
- X25 with Connection–Oriented mode of service (**XX25**).

A basic XTI is now embedded with AIX, but it does not provide all the facilities of Bull-enhanced XTI, in particular only access to TCP/IP is supported.

- The commands **chxti** and **lsxti** allow the choice of basic XTI Bull-enhanced XTI or **XX25** API to be determined when compiling and linking the application programs.

XTI conforms to *X/Open Transport Interface XPG4 CAE Specification Version 2*, whereas the previous version conformed to the XPG3 specification for the previous version.

- The *XTI Porting Guide*, helpful for the programmers, describes the XPG3-XPG4 differences, as well as the differences related to the new OSI stack, TCP/IP and NetShare versions.

Other new functionalities

- Instead of hard-coding providers, applications can use the network selection and address mapping services of the XTI name server. Provider names and priorities can be placed in a shell environment variable (**XTINETPATH**) or in the file **/etc/xtiprotocols**. This information can be accessed using the **t_gettp** subroutine.
- The name server can be used to store option profile names and associated XPG4 options in the database file **/etc/xtiopts**. The database can be managed using the XTI configurator. Applications can read this information using a new subroutine **t_getopt**.
- XTI is an internationalized product which uses NLS services. Message catalogs for XTI are available in English.
- A new statistics and diagnostics tool **xtistat** can be used to display XTI connection information (context addresses, numbers of input and output packets, internal structures and counters). It can also be used to display kernel statistics and analyze core dumps.
- XTI takes advantage of stack support of PVCs. X25 facilities are allowed using two new options: `TCO_X25FAC` and `TCO_X25CALLDT`.
- Two shared libraries are provided: standard and thread-safe.

XX25 allows to access X25 services through XTI:

- It uses the same functions as XTI. Flags and options have been added to support X25 functions such as bit D, bit Q, reset or X25 facilities.
- To use **XX25**, the product **bullx25** is required. This product includes the network provider, packaged in **bullx25.xpi**.

Session API

The session API is an interface to session layer services. The new version provides these additional functionalities:

- Support of activity management and capability data exchange functional units, which implies:
  - New subroutines.
  - **Se_ReadEvent** subroutine returns signals corresponding to the above events.
  - The *option* field of the negotiation profile accepts two new values for the functional units: `OP_ACTIVITY` and `OP_CAPACITY`.

- About session level addressing:
  - Acceptation of connect indications, TSAPs (TSELs and NSAPs) are now meaningful for both the calling and the called side. Previously, only SSELs were used for both sides).
  - Indications for unknown TSAPs are no longer systematically transmitted from the transport to the session layer. A session user must be bound to a specific TSAP, using the **Se_aListenid_rq** subroutine, in order to receive connect indications from it. Wildcarding may be used at the SSEL, TSEL and NSAP levels to approach former behavior.
- A new subroutine **Se_OpenBis** can be used instead of **Se_Open,** for improved performance, when only one link is required (no multiplexing of connections on one **fd**).
- X25 PVCs are supported for session connections.
- The new API uses a runtime (shared library).

To help in debugging of **Session** applications, the previous **trapises** command has been replaced by traces using **ODIT**.

XAP

The XAP API conforms to *ACSE/Presentation Services API (XAP) CAE Specification* and provides applications with access to services implemented in upper layers of OSI stack:

- **ACSE** (Application layer) provides association control services for establishing and releasing associations between applications entities.
- **Presentation** layer is responsible for the representation of data in transit between applications.
- **Session** layer provides services which allow applications to organize and synchronize their interactions and data transfers.

It replaces the previous ACSE/P API, a proprietary ACSE/Presentation interface.

The XAP services are a superset of the previous ACSE/P API services, but the interface and the implementation are quite different.

> **Warning: Previous applications which used the ACSE/P interface must be ported. It is advisable to refer to** *ACSE/P to XAP Interface Porting Guide*. See **Bibliography**, on page A-1.

A major change lies in the necessity for an application using XAP instead of ACSE/P to **encode data**, whereas the ACSE/P interface performed the encoding itself. Applications can use the services of the **ASN.1** compiler for ASN.1 encoding and decoding.

Changes are relative to supported services, addressing, error codes, tracing, configuration and limits.

From an architecture point of view, the APi uses a runtime (shared library).

Services

XAP supports new services compared to ACSE/P.

- **ACSE level**
  - Support of AE title form 1 and form 2 formats, previously only form 2 was supported.
- **Presentation level**
  - Default context, capability data exchange, and activity management.
  - Exception reporting.
  - Support of normal and X.410 (1984) modes.
  - Support of fully encoded user data (PDV lists).
  - Support of PSELs.
- **Session level**

  - data separation functional unit
  - SPDU concatenation.

Addressing

- XAP takes advantage of the stack support of wildcarding of PSELs, SSELs, TSELs, and NSAPs. XAP wildcarding follows the rules defined in the X/Open standard.
- Support of X25 PVCs.

Error Codes

XAP error codes are completely different from the ACSE/P error codes.

The **ap_error** subroutine provides the error message corresponding to an error code.

Tracing

To help in debugging of **XAP** applications, the previous **asedebug** command has been replaced by traces using **ODIT**.

Environment and Configuration

The concepts used in XAP differ from those used in ACSE/P:

- *objects* in function interfaces for ACSE/P,
- *environment attributes* for XAP, which may be configured using the **ap_set_env** or **ap_init_env** subroutines.
  Bull-specific connection information (connection profiles) is configured using **CIS** or defining a specific environment attribute **AP_CONNECT_INFO**.

Limits

XAP has no intrinsic limits. Limits are defined by OSI stack limits.

- The associations within a process are limited to 1024 session-level connections, instead of 512 for ACSE/P API.
- No limit for the user data length.
- Unlimited number of transfer syntaxes in a Presentation Context Definition List (PCDL), instead of one transfer syntax per PCDL for ACSE/P API.
- Unlimited number of contexts in the Defined Context Set (DCS), instead of ten contexts per DCS for ACSE/P API.

ROSE

The new **ROSE** API:

- takes advantage of the stack improvements in addressing (calling and called selectors and NSAPs supported) and wildcarding.
- manages some new error codes, whereas some others have been removed.
- memory consumption by memorizing a limited number of outstanding operations.

To help in debugging of **ROSE** applications, the previous **asedebug** command has been replaced by traces using **ODIT**.

OSI Applications

OSI applications take advantage of the new stack feature called Stack Application Support (SAS). It allows an administrator to configure a list of commands to automatically start or stop applications when the stack itself is loaded or unloaded.

TPAD-HPAD Terminal PAD/Host PAD, an asynchronous terminal management service.

OTM Open Terminal Manager, a terminal manager (local to remote connections). Associated with a special *Starter Set* version of **CPI-C**.

CPI-C OSI Common Programming Interface for Communications, an application-to-application communication service.

UFT Unified File Transfer, to transfer files to Bull and IBM proprietary systems.

FTAM File Transfer, Access, and Management, a file service for OSI systems.

X500 A directory service based on the X-500 standard.

CMIP Agent A network management agent for supervising OSI stack resources.

TPAD-HPAD

TPAD-HPAD allows a user of an asynchronous terminal to connect to and communicate with a remote host application using X.25 packet switched communications.

New functionalities have been added:

- CCITT PAD88 support:

  New parameters/values are now available to upgrade the PAD protocol to 1988 recommendations. Their values concern:

  - the management of the procedure between the start-stop DTE mode and PAD
  - the management of the assembly-disassembly of packets
  - a number of additional functions related to the operational characteristics of the start-stop DTE mode.

- SPAG Y/13 profile:

  Updates have been made in protocol and configuration to support LAN/WAN relays, making easy a connection between a user on a LAN network and a remote host on a WAN network.

- update PAD printing functionalities and configuration:

  This function enables ro include in a local site printers accessed via a PAD connection over a LAN/WAN network.

- support of PVCs using the X.25.3 Access Method:

  On PVC the virtual circuit is permanent. The link is established once and for all when the board is loaded. There is no notion of connection/disconnection.

- new traces and a new interface have been added, as requested by the Prime Support:

  While previous trace tools used a level bracket, the new interface works with a mask so that any combination of levels is allowed. Now it is possible to run the trace using dialog screens of SMIT on HPAD.

- maintenance improvements:

  Tools are provided to link PAD features to the 'OSI and GCOS connectivity problem determination' item in SMIT menus.

OTM

OTM consists of a terminal manager itself, and an optional product called **CPIC-SS** BULL (**C**ommon **P**rogramming **I**nterface for **C**ommunications **S**tarter **S**et for BULL).

OTM has these new functionalities:

- multi–provider architecture,
- support of X25 facilities and PVCs over WANs,
- support of NetShare and full ISO addressing,
- the OTM server can handle up to 255 PCs using **Affinity**,
- enhanced configuration menus,
- enhanced traces (DSA record dump included),
- 1024 connections support,
- **Affinity** performances improved.

No new functionality for **CPIC-SS**.

CPI-C OSI

CPI-C is an application to application communications interface that conforms to the X/Open standard API. It integrates the eXtended Cooperative Protocol Level 2 (XCP2), which is SNA compatible and applies to the Bull DSA environment, into the OSI environment.

One new functionality of CPI-C is the delivery of a library for PCs running Windows V3. This allows to develop cooperative applications running under Windows and interoperating with peer CPI-C applications running under Bull AIX platforms, GCOS mainframes, or IBM mainframes (the latter through OSF gateway).

The 4.1 CPI-C/OSI brings performance, configurability and robustness improvements, with in particular:

- the removal of the session and side information redirection access
- modifications of the SMIT menus presentation
- help screens for configuration menus have been added
- inclusion of CPI-C/OSI in **osisnap** command
- the maximum size of buffers exchanged between two user applications is now 32767 bytes.
- improvement of the tuning possibilities. Now it is possible to configure the size of the messages exchanged between applications and the node server up to 32767 bytes instead of 512.
- a two-way facility has been included, it is possible to configure in a side information entry, two ways to access a remote application (two modes, local and remote LUs and two configuration files). The CPI-C library is in charge of choosing the way to use.

UFT

UFT allows files to be transferred between DPX/20 systems and/or other systems (Bull GCOS systems or IBM MVS systems). New features include:

- file compression to speed transfers,
- a trace facility with four levels,
- a mechanism to recover aborted transfers,
- a verbose mode which allows the number of bytes transferred to be continuously displayed,

- no intrinsic limits concerning the number of open connections (stack limits, 1024 connections) instead of limitation to 64 in the previous version.

FTAM

FTAM is used to transfer files among different computer systems which may be based on different operating systems, architectures, or file systems. This machine independence is obtained using a Virtual File Storage (VFS) system. A specific API is associated with **FTAM**.

The new functionalities of **FTAM** are:

- Configuration is performed using SMIT menus and CIS services, so the **ftcnf** configuration tool is no longer delivered.
- The management of incoming connections has been improved by keeping connection information in the stack session layer, which allows an FTAM responder to spend more time processing connections. In the previous version, connections could be lost if several arrived simultaneously.
- Previous restrictions about the recover mechanism (for FTAM-2 and FTAM-4 document types) have been removed.

X500

X500 is a directory service, typically used to store names, phone numbers, addresses and other information for people or things.

The new functionalities of **X500** (version 3.0) are:

- Configuration and monitoring now performed using SMIT. The configuration tools also use CIS services for management of X25 network facilities and calling user data.
- On-line help is available for the Global Directory Services (GDS) commands.

CMIP Agent

The CMIP agent for OSI layers has new functionalities:

- Management of the X25 network sublayer.
- Configuration performed using SMIT menus and CIS services, instead of using a text file.
- Browser, based on SMIT menus, to display the objects supported by CMIP.

OSI Miscellaneous Products

These products are not strictly speaking OSI products.

bullx25	HiSpeed WAN Comm. software.
NetShare	Transport provider allowing OSI higher layers to access TCP/IP.
ASN.1	A tool to encode and decode data in the ASN.1 communication format.

bullx25

bullx25 manages the HiSpeed WAN Comm. adapters (4Port HiSpeed WAN Comm. and 1Port HiSpeed WAN Comm. adapters) and provides OSI stack and TCP/IP with X.25-3 services implemented on these adapters.

The improvements associated with OSI stack on AIX Version 4.1.4 are:

- enhancements to benefit from multiprocessor platforms,
- a new command **xdping**, to test HiSpeed WAN Comm. adapters and connections,
- a new option for **funstat** command, to test the availability of the bullx25 license,
- support of the use of **XX25** new API (new package **bullx25.xpi**),
- the possibility to clear the current configuration of a line, and get default values,
- the possibility to tune explicitly the incoming calls routing between OSI, XPI and HVX,
- an SNMP agent integrated in AIX SNMP environment.

NetShare

NetShare (RFC 1006) allows OSI upper layers (session, presentation and application) to use TCP/IP communications instead of OSI low layers.

The improvements associated with OSI stack on AIX Version 4.1.4 are:

- enhancements to benefit from multiprocessor platforms,
- dynamic configuration (can be changed when NetShare is running) instead of static configuration,
- a larger limit on the number of simultaneous open connections, 1024 instead of 512.

ASN.1

ASN.1 allows to encode and decode data in the ASN.1 communication format.

The new version provides these additional functionalities:

- **Support of new types**
  A **subtype** is defined by the notation of a parent type followed by an appropriate subtype specification. This specification reduces the values of the subtype to a restricted number of values of the parent type.
  Three notations are supported by the **ASN.1** compiler:
  – size constraint,
  – permitted alphabet constraint,
  – value range constraint.
  Only the type notation for subtype is supported.
- **Rules of extensibility**
  The **ASN.1** notation supports an extension marker which signals to the encoding rules that the **ASN.1** specification of the involved data structure may differ between two releases of the same protocol.
  This notation may be used only on a restricted number of built-in ASN.1 types: SET/SEQUENCE, BIT STRING, ENUMERATED, CHOICE.

If a value with additional (undecodable) extensions is received, this results in an error specification on the marker extension, which returns these undecoded values.
If an extension addition value is absent, this results in an error specification on this element only.

- **Improvement of code generation**

  – **Portability**
  The code generator embeds basic C types in specific *typedef* prefixed by *asn1_* or *ASN1_*. These types are generated according to the target machine by setting optional compilation flags in *api/asn1.h*.
  For example, an INTEGER is generated as C type *asn1_int32* which means it is 32 bits long. It is defined in *api/asn1.h* as an *int* or a *long* according to the length of the integer on the target machine.

  – **Optional components management**
  Each time a component is represented by an *asn1_field* structure, the optionality is managed by a negative length preferably to a NULL pointer to the structure. This permits to minimize and simplify memory allocation especially at decoding.

  These evolutions in code generation may lead to slight differences between C structures generated by the previous ASN.1 compiler and by the new one.
  The interface of the coding procedures generated is not modified.
  Compilation flags related to the target machine are generally set by the C compiler itself.

- **Suppression of restrictions**
  – Empty SET and CHOICE components are accepted.
  – REAL components can use the 0 (zero), PLUS-INFINITY and MINUS-INFINITY values.
  – DEFAULT clauses can be used on SET, SEQUENCE, and string components.

- **Addition of restrictions**
  Subtypes of ENUMERATED types are not correctly processed by the compiler.
  Replace the ENUMERATED subtype declaration by the declaration of an equivalent ENUMERATED type.

Chapter 2. Packaging

The OSI communications products and associated SRBs are delivered together on a CD-ROM, except for XAP and ROSE and Session API which are delivered on two specific CD-ROMs.

An OSI communication product may be delivered as one or more packages, the SRB specific to the product is delivered as a package as well.

For each OSI product the corresponding packages and eventually the options are listed. The options are listed only when they correspond to a specific usage, for instance an option which would be useful only with TCP/IP.

OSI products packaging listed:

- **OSI Stack, on page 2-2,**
- **OSI APIs, on page 2-3,**
  - **MAX25-3**
  - **XTI**
  - **Session API** and **ROSE**
  - **XAP**
- **OSI applications, on page 2-4,**
  - **TPAD-HPAD**
  - **OTM**
  - **CPIC-C OSI**
  - **UFT**
  - **FTAM**
  - **X500**
  - **CMIP Agent**
- **OSI miscellaneous products, on page 2-5,**
  - **bullx25**
  - **NetShare**
  - **ASN.1**
- **BASIC-COM and GCOS-COM Bundles, on page 2-5.**

Note: The SRB packages are not listed hereafter.

OSI Stack Packaging

The OSI stack (low layers and high layers) packaging is different between versions running on **AIX Version 3.2.5** and **AIX Version 4.1.4**:

- **OSI Stack version for AIX Version 3.2.5** contained 3 LPPs (License Program Products)

  osi_frame, general software and configuration tools, prerequisite for the 2 other LPPs,

  osi_low, OSI layers 1 to 4,

  osi_high, OSI layers 5 to 7.

- **OSI Stack version for AIX Version 4.1.4** contains 6 packages:

  osisnap, for the **osisnap** command,

  osi_frame, general software and configuration tools, with **osisnap** as prerequisite,

  osi_low, OSI layers 1 to 4 and MAX25-3 environment, with **osi_frame** as prerequisite,

  osi_high, OSI layers 5 to 7, trace and configuration elements, and runtime for XAP et Session APIs, with **osi_frame** and **osi_low** or **netshare** as prerequisite,

  cis, for **CIS** (Connection Information Service),

  com_diag, for maintenance framework.

OSI APIs Packaging

MAX25-3

Previously packaged with TPAD-HPAD, **MAX25-3** is now packaged with **osi_low**.

XTI

XTI consists of a single package **xti_api** containing three options:

- **xti_api.com**, a **mandatory** option containing the software common to all transport providers and software specific to TCP/IP.
- **xti_api.cotp**, option to be used with the transport providers OSI or NetShare, with **osi_low** or **netshare** as prerequisite.
- **xti_api.xx25**, option to be used over X25 with **bullx25.xpi** as a prerequisite.

Session API and ROSE

Previously, the **osi_api** LPP contained a single OPP (**osi_api.obj**) including Session API, ROSE and ACSE/P.

The main packaging differences between previous and new version are due to the removal of the ACSE/Presentation library (replaced by XAP), and the inclusion of trace files into the **osi_high** package.

The **Session** and **ROSE** APIs are provided on a specific CD-ROM and included in the **osi_api** package which contains the following options:

- for **ROSE**:
  - **osi_api.rose.adt**, contains the ROSE include files and libraries.
  - **osi_api.rose.samples**, contains ROSE API examples.
- for **Session API**:
  - **osi_api.apises.adt**, contains the session include files and libraries.
  - **osi_api.apises.samples**, contains session API examples.
- common to **ROSE** and **Session API**:
  - **osi_api.ext**, for future use.

The prerequisites for **Session API** and **ROSE** are **osi_high** and **asn1comp**.

(**ASN.1** runtime is used by **ROSE**, but **ASN.1** license is not necessary).

XAP

Previously **ACSE/P** was included in the **osi_api** LPP.

XAP consists of:

- a package **xap_api**, provided on a specific CD-ROM.
- trace and configuration files included in **osi_high**.
- ODIT files included in **osi_frame**.

The **XAP** prerequisites are **cis**, **osi_high** and **asn1comp**.

(**ASN.1** runtime is used by **XAP**, but **ASN.1** license is not necessary).

OSI Applications Packaging

TPAD-HPAD

As previously, **TPAD-HPAD** consists of a single package **tpad_hpad**, with **osi_low** as prerequisite.

The difference between previous and new versions is that **MAX25-3** is no more packaged with **tpad_hpad** but with **osi_low**.

OTM

OTM consists of a single package **otm**, with **osi_high** as prerequisite.

CPIC–SS consists of a single package **cpic_ss**, with **otm** as prerequisite.

CPI-C OSI

CPI-C OSI consists of a single package **xcp2**, with **osi_high** as prerequisite.

UFT

UFT consists of a single package **uft**.

FTAM

FTAM consists of a single package **ftamx**, with **cis** and **osi_high** as prerequisites.

X500

X500 has a user part (DUA) and a server part (DSA). The use of the server part requires the user part to be installed.

Previously there were two different LPPs:

- **x500_dua** contained only the user part,
- **x500_ds** contained both the user and server parts.

Now, there are still two packages, **x500_dua** and **x500_ds**:

- **x500_dua** contains the user part,
- **x500_ds** contains only the server part, **x500_dua** as prerequisite.

X500 is used:

- either with **TCP/IP** with **CIS** as prerequisite,
- or with **XTI** onto NetShare or OSI low layers; with **CIS**, **XTI** plus either **NetShare** or **OSI low layers** as prerequisites.

CMIP Agent

CMIP consists of a single package **cmip_agent**, with **cis** and **osi_high** as prerequisite.

OSI Miscellaneous Products Packaging

bullx25

bullx25 was previously packaged in a single LPP called **bullx25**.

Now it comes in three packages:

- **bullx25.board**, which is mandatory for the use of HiSpeed WAN Comm. adapters, with the prerequisite **osi_low** if the adapters are accessed by OSI stack or **TCP/IP** if they are accessed by TCP/IP.
- **bullx25.tcpip**, which is necessary for **TCP/IP** support using HiSpeed WAN Comm. adapters.
- **bullx25.xpi**, which is necessary to use the **XX25 API**,
- **bullx25.adm**, which is necessary to use X25 SNMP agent.

NetShare

NetShare is split in:

- **netshare.rte**, the runtime part,
- **netshare.smit**, the configuration part,
- **netshare.tools**, traces and rfcstat tools.

ASN.1

ASN.1 is provided in a single package **asn1comp** which is made of three options:

- **asn1comp.adt**, code generator,
- **asn1comp.rte**, run-time,
- **asn1comp.sample**, programming examples.

BASIC-COM and GCOS-COM Bundles

Two new packages of communications products are available, to bundle the existing features:

BASIC-COM

BASIC-COM is dedicated to OSI, NetShare and TCP/IP communications on both LAN and WAN networks. It includes the following communication products:

- Bullx25,
- OSI Stack,
- NetShare (RFC 1006),
- XTI API/XX25,
- TPAD–HPAD,
- PAD Printing,
- MAX25-3 API.

GCOS-COM

GCOS-COM is dedicated to the mainframe integration over OSI/DSA network. It includes the following communication softwares, with the pre-requisite BASIC-COM:

- UFT,
- OTM,
- CPI-C SS.

Chapter 3. Migration

OSI Communications Migration Overview

Concerning the OSI communications products and relative applications, the migration must be done step by step.

They are described in:

1. **Preparation Phase, on page 3-2,**
2. **AIX Migration, on page 3-3,**
3. **Communication Migration, on page 3-3,**
4. **Post–migration Phase, on page 3-4,**
5. **Preparation of Restarting Applications, on page 3-5.**

Migration from DPX/20 (3.2.5) to DPX/20 (4.1.4)

The following tasks, to be done in sequence, are described in this chapter:

- Preparation Phase
- AIX migration
- OSI Communications installation and configuration
- ”Post–migration”
- Preparation of restarting applications

Preparation Phase

- First of all, you must ensure that the OSI products migration you want to perform is possible and that the versions of your products are at the right level.
  Migration can be performed only:

  from AIX 3.2.5, OSI Stack 1.2.17 and associated products to
  AIX Version 4.1.4, OSI Stack 02.03.05 and associated products.

  The table below gives the OSI stack and products version numbers for a migration from AIX 3.2.5 to AIX 4.1.4 as they must be.

  Note that the products for which problems have been ”fixed by software” should have been updated.

Packages	AIX 3.2.5	AIX 4.1.4
OSI Stack layers 1 to 4	01.02.17	02.03.05
OSI Stack layers 5 to 7	01.02.17	02.03.05 (1)
XTI–API	01.02.17	02.03.05
OSI–API	01.02.17	02.03.05 (2)
XAP	01.02.17	02.03.05
TPAD–HPAD	01.03.05	01.05 .04
OTM	01.02.03	01.05.02
CPIC–OSI	02.07.10	02.09.06
UFT	02.01.06	03.01.01
FTAM	04.00.09	04.02.04
X500	02.00.04	03.00.06
CMIP	01.00.02	01.01.03
CPIC–SS	01.02.03	01.05.02
BULLX25	01.02.23	02.03.05
NetShare	01.02.17	02.03.05
ASN1	01.02.01	01.03.05

Table 1. OSI Stack and associated products version numbers in AIX 3.2.5 & AIX 4.1.4

(1) : contains MAX25–3.

(2) : contains Session and Rose interfaces.

- Then, although OSI Stack and most of OSI products migrate while preserving user communication information, (so far as you have chosen the AIX migration installation mode), you must ensure you are not concerned with the following restrictions:
  a. CMIP: no migration mechanism is provided.
     You must de–install before installing the new version.
  b. X500: migration is not automatically performed because of the disk space and the CPU timed needed by such a migration. If a migration is needed, X500 asks the operator at launch time to perform it. X500 provides the operator with the way to do it.
  c. BULLX25: user data do not migrate: you must backup manually the configuration of the line(s) of your X25 adapter(s).
     Concerning X25 adapters, only HiSpeed WAN comm. adapters are supported at present. See the SRB of AIX 4.1.4 for additional information.
  d. Configuration names: in case the name of your Stack configuration includes characters different from { [a..z]   [A..Z] [0..9] _ \ / . }, you must change its name before AIX Migration phase, so as to be able to use it again.

Note:

– Obviously, you must **unload the Stack** before starting AIX migration.

– Installation procedures themselves will not be described in detail. For each product you want to migrate, refer to the associated documentation (installation section) and to the SRB, in particular to estimate the **disk size needed for the new version.**

About AIX Migration

The migration of OSI communications products can be performed only if AIX itself is correctly migrated. Only the **migration installation mode** allows afterwards to migrate OSI products.
So, you must choose the migration option, that preserves user data, configuration files (/**etc**), binary files and product definitions (/**usr** and /**var**).

Refer to *AIX Installation Guide* for more details.

Warning:

– Even if you choose the AIX 4.1 migration installation mode, remember that X25 configuration must be backed up manually: for instance, you may run the command: *”lsattr –E –l x25l<n>*” and keep the result to use it afterwards

– Remember that the name of your stack configuration must have a correct syntax. If it has not, give it another name before starting AIX migration.

Note:

– TCP/IP migrates at the time when AIX migrates.

– It is advised to perform AIX migration in english, (LANG=C (POSIX)), and with enough disk space in /tmp. See SRB of AIX 4.1.4 for more information.

OSI Communications Installation and Configuration

1. List the OSI products, previously installed on the DPX/20, using the command:

   ```
   lslpp –l
   ```

2. Do not de-install the OSI products, except those which will be no more used in this new configuration.

3. Install each of the OSI products, respecting the sequence of prerequisites described in the associated documentation and using the command:

   ```
   smit install_latest
   ```

   Refer to **OSI Communications Packaging**, on page 0. to get information about software delivery and the names of the OSI products LPPs which have to be installed.

   The sequence of installation may be, according to the needs:

   - OSI stack (framework, low layers and high layers) including MAX25–3
   - bullx25
   - TPAD–HPAD
   - NetShare
   - XTI
   - OTM
   - UFT
   - Other applications (FTAM, ...)

   No new configuration is necessary.

   For each product, refer to the associated documentation to check whether any special tasks have to be performed.

4. List the OSI products, currently installed on the DPX/20, using the command:

   ```
   lslpp –l
   ```

   `lslpp –L` displays with an O (as Obsolete) the previous version of each product (except TPAD–HPAD).

Post–Migration Phase

Once the OSI products have been installed, you may have, according to the products you have migrated, to perform:

- X25 lines configuration, by using the configuration parameters you have manually saved in the previous phase.

At this point, check you have installed the License keys you need, and load the stack.
See *AIX iFOR/LS System Management Guide*

If UFT is to be used, you will have to perform:

- UFT starting, to be done once (with smit or by running the command **rc.uft start**).

Before Restarting Applications

Before restarting the applications, some other points have to be checked referring to **User Interface Changes**, on page 0:

- Some previously developed specific tools, such as shell scripts, may use commands whose user interface (options, parameters) have changed. In this case they must be modified.
- Applications developed using OSI Communications API, need to be ported in some cases.

Chapter 4. User Interface Changes

On the previous version, the user may have developed application programs or tools (shell scripts) accessing **subroutines**, **commands** or **files**. This chapter highlights the changes occurring between OSI stack versions on AIX 3.2.5 and AIX Version 4.1.4 for each product.

In tables describing changes, the status changes are indicated as follows:

ss	The new version is a superset of the previous version.
as	The new version is almost the same as the previous version.
d	The new version is very different from the previous version.
r	The old version has been removed (does not exist in the new version).

You can find more information about these changes in the following sections:

- **OSI Stack** Interface Changes on page 4-2
- **OSI APIs** Interface Changes on page 4-3
  - **MAX25-3**
  - **XTI**
  - **Session API**
  - **XAP**
  - **ROSE**.
- **OSI Applications** Interface Changes on page 4-7
  - **TPAD-HPAD**
  - **OTM**
  - **CPI-C OSI**
  - **UFT**.
  - **FTAM**
  - **X500**
  - **CMIP Agent**.
- **OSI Miscellaneous Products** Interface Changes on page 4-11
  - **bullx25**
  - **NetShare**
  - **ASN.1**.

Warning: Only the commands and files which have been modified between **previous version** and **new version** are documented here.

OSI Stack Interface Changes

Command Changes

New Commands	
Command	**Comments**
osisas	Manages the Stack Application Support (SAS) feature.
osisnapshot	Displays Dynamic RIB, EI–IS protocol and Memory buffer data.
OSIdefault	Sets the minimum stack configuration based on installed adapters.
osifilter	Sets the configuration of authorized calling/called addresses.
osinetbios	Sets the configuration for NETBIOS multicast environment.
osirelay	Sets the transport relay configuration.
osiroute	Sets the protocol parameter configuration for transport relay.

Modified Commands		
Command	**Status**	**Comments**
osiadapt	as	The **–c** flag uses ODM codes and replaces the **–a** flag.
osiadd	r	Some functionalities now moved to **osisnapshot**
osiconf	d	The syntax of the **–r**, **–i**, and the **–t** flags has changed. The **–mz** and **–st** flags are new. The **–g** and **–p** flags have been removed.
osiconfgen	r	
osifunct	r	
osiload	as	A new daemon is used to start up the stack.
osirib	as	The **–m** and the **–p** flags have been removed.
osisnap	as	Extended functionalities for use by OSI applications.
osisubnet	as	The **–p** and **–s** flags have been removed.
osiunload	as	A new daemon is used to stop the stack.
osix25	as	The **–c** flag uses ODM codes and replaces the **–a** flag.
pmaderror	ss	A new **–e** flag can be used to display messages for **errno** codes.
trmad	d	The interactive menus and commands have been changed.
trmem	r	Functionalities now moved to **trmad**.
osinsap	ss	A new option **–L** has been added to deal with Local/Group NSAP display. The **–a** option is extended to deal with Local/Group NSAP definition.

ss	The new version is a superset of the previous version.
as	The new version is almost the same as the previous version.
d	The new version is very different from the previous version.
r	The old version has been removed (does not exist in the new version).

OSI APIs Interface Changes

MAX25-3

The MAX25-3 subroutines are unchanged. Previously, MAX25-3 contained a library stored in the file **/usr/lib/drivers/osi/max3/lib.a**. This library is no longer delivered. In the new version, the MAX25-3 API is delivered directly integrated into a driver contained in the file **/usr/lib/drivers/pse/osi/osi_low**.

MAX25-3 offers binary application compatibility. But if recompilation is needed, source compatibility is also assured.

XTI

Programming Changes

The *XTI Porting Guide* provides all the information needed to port an XTI application from XPG3-XTI to XPG4-XTI.

The link options have changed, the simplest way to compile and link an XTI application (`file.c`) using the Bull-enhanced XTI library and XTI Name Server library is to use the standard C Compiler resource file **/etc/xlC.cfg** and run the following commands:

- to use the shared libraries:

```
cc file.c –F:xticc
```

- to use the thread-safe libraries:

```
cc file.c –F:xticc_r
```

- to use the non-shared libraries:

```
cc file.c –F:xticc –bnso
```

The same way of compiling is to be applied to **XX25** applications with the `–F:xx25cc` option:

```
cc file.c –F:xx25cc –bnso
```

Command Changes

New Commands	
Command	**Comments**
chxti	Change current XTI attributes.
lsxti	List current XTI attributes.
xtiopt	Manage various XTI administrative options.
xtistat	Display XTI statistics and diagnostics information.

Modified Commands		
Command	**Status**	**Comments**
xtihost	d	The syntax for the **–r** and **–l** flags has changed. The **–ls** and **–p** options have been added. The network type is a now a name (previously it was a number).
xtiserv	as	The syntax for the TSEL value (**–n** flag) has changed.

File Changes

New Files	
File	**Comments**
/etc/xtiopts	Options and associated profiles in TLV format.
/usr/lib/xti_api/libxti.exp	Regular XTI export file.
/usr/lib/xti_api/libxti_ns.exp	Regular XTI name server library.
/usr/lib/xti_api/libxti_r.a	Reentrant, thread-safe XTI library.
/usr/lib/xti_api/libxti_r.exp	Reentrant XTI export file.
/usr/lib/xti_api/libxti_ns_r.a	Reentrant XTI name server library.
/usr/lib/xti_api/libxti_ns_r.exp	Reentrant XTI name server export file.
/usr/lib/xti_xx25	XX25 program production directory.
/usr/lpp/xti_api/examples/binop_c/cots_r	Example use of XTI with threads.

Modified Files		
File	**Status**	**Comments**
/usr/include/sys/ticom.h	as	Moved to **/usr/include/xti_api/sys/**.
/usr/include/sys/xti.h	d	Moved to **/usr/include/xti_api/sys/**.
/usr/include/sys/xti4.h	r	Contents integrated into **sys/xti.h**.
/usr/include/xti.h	d	Moved to **/usr/include/xti_api/**.
/usr/include/xti4.h	r	No longer needed (**sys/xti4.h** removed).
/usr/lib/libxti.a	r	Only the shared library is delivered.
/usr/lib/libxti_s.a	d	Moved to **/usr/lib/xti_api/libxti.a**.
/usr/include/xti_ns.h	d	Moved to **/usr/include/xti_api/**.
/usr/lib/libxti_ns.a	r	Only the shared library is delivered.
/usr/lib/libxti_ns_s.a	d	Moved to **/usr/lib/xti_api/libxti_ns.a**.
/usr/lib/drivers/pse/xti3mod	r	Replaced by **/usr/lib/drivers/pse/xti4mod**.

In the above table,

ss	The new version is a superset of the previous version.
as	The new version is almost the same as the previous version.
d	The new version is very different from the previous version.
r	The old version has been removed (does not exist in the new version).

Session API

For previous and new versions, Session API provides two libraries, the link options have been changed:

```
Shared Library Options:
old stack: -L/usr/lib/api -lses_s
new stack: -bI:/lib/libses.exp

Normal Library Options:
old stack: -L/usr/lib/api -lses
new stack: -L/usr/lib/api -lses
```

Source compatibility is maintained from previous to new version of Session API (except for some error codes which are still defined in the include file, but no longer returned), so applications only need to be recompiled.

malibse.h contains some new error codes and signals.

Modified Commands		
Command	**Status**	**Comments**
trapises	r	Trace configuration. Integrated in **odit.**

r The old version has been removed (does not exist in the new version).

Subroutine Changes

The new subroutines are relative to the activity management and capacity data exchange functional units.

New Subroutines	
Subroutine	**Comments**
Se_aCapData_rq	Send capability data.
Se_CapData_rp	Reply to a send capabilty data.
Se_aDiscardAct_rq	Discard an activity.
Se_DiscardAct_rp	Reply to a discard activity.
Se_aEndAct_rq	End an activity.
Se_EndAct_rp	Reply to an activity end.
Se_GControl_rq	Give control.
Se_aIntrAct_rq	Interrupt an activity.
Se_IntrAct_rp	Reply to an activity interrupt.
Se_StartAct_rq	Start an activity.
Se_ResumeAct_rq	Resume an activity.
Se_OpenBis	Optimized routine for opening a link that does not use multiplexing.

The **Se_Read_Event** has changed: it now returns a new exception report signal and other signals which correspond to the new subroutines listed above.

XAP

The *ACSE/P to XAP Interface Porting Guide* provides all the information needed to port an application from ACSE/P to XAP.

XAP as well as ACSE/P provides two libraries, the link options have been changed from ACSE/P to XAP:

```
Shared Library Options:
ACSE/P: –L/usr/lib/api –lacse_s
XAP: –bI:/lib/libxap.exp

Normal Library Options:
ACSE/P: –L/usr/lib/api –lase –lacse –lrose2
XAP: –L/usr/lib/api –lxap
```

ROSE

For previous and new versions, ROSE provides two libraries, the link options have been changed:

```
Shared Library Options:
old stack: –L/usr/lib/api –lase –lrose_s –lacse2
new stack: –bI:/lib/librose.exp


Normal Library Options:
old stack: –L/usr/lib/api –lase –lrose –lacse2
new stack: –L/usr/lib/api –L/usr/lib/osi –lrose –lasn1
–losi_trace –bI:/lib/pse.exp
```

Source compatibility is maintained from previous to new version of ROSE API (except for some error codes which are still defined in the include file, but no longer returned), so applications only need to be recompiled.

rose.h contains some new error codes.

Modified Commands		
Command	**Status**	**Comments**
asedebug	r	Trace configuration. Integrated in **odit.**

r	The old version has been removed (does not exist in the new version).

OSI Applications Interface Changes

TPAD-HPAD

File Changes

New File	
File	**Comments**
/usr/lib/lpd/pio/etc/pad.attach	Attachment type

Modified Files		
File	**Status**	**Comments**
/usr/lib/lpd/pio/etc/pad_proc	r	This SMIT configuration file is now part of a stanza file.
/var/spool/lpd/pio/custom	ss	New format and moved to **/var/spool/lpd/pio/@local**/.

OTM

CPIC-SS offers binary application compatibility and has no interface changes.

The **OTM** interface changes are listed hereafter.

Command Changes

Modified Command		
Command	**Status**	**Comments**
tws2107	as	Addition of a new option (**–f**) which allows up to four X25 facility fields to be defined.
dumps	d	Full DSA record dump.
inftms	d	Reporting of multi–provider architecture.
vipemu	as	Keeps track of original environment variables.

File Changes

Modified Files		
File	**Status**	**Comments**
/usr/lib/iso/site	ss	A field for X25 facilities has been added.
/usr/bin/tws2107	ss	It is both an executable file and a configuration file. The **–f** flag has been added to define X25 facilities.

In the above table,

ss	The new version is a superset of the previous version.
as	The new version is almost the same as the previous version.
d	The new version is very different from the previous version.

CPI-C OSI

The **CPI-C OSI** configurator has new SMIT menus for managing the following features: the node server, the current general and OSI session configuration, the logging and trace information, the listeners, and the display information for PC support.

CPI-C OSI offers application source compatibility.

UFT

Command Changes

UFT contains commands that can be run from the operating system prompt and subcommands (user commands) that can only be run from the **UFT** prompt.

New Subcommands	
Subcommand	**Comments**
cmp	Enables or disables compression of data before transfer.
noabort	Enables or disables the full restart functionality.
trace	Sets the trace level and trace file.
verbose	Enables or disables the display of the number of transfered kilobytes.

Modified Commands and Subcommands		
Command	**Status**	**Comments**
rc.uft	ss	New parameters for trace, verbose, and multiple server support.
uft	ss	New parameters for trace support.
uftreqp	r	The requestor process is no longer needed to run the **uft** command.
uftserp	d	Renamed **uftser** and accepts a trace level parameter.
Subcommand	**Status**	**Comments**
connect	as	Provides default values for login information.
who	as	Provides additional status information such as the trace level.

In the above table,

ss	The new version is a superset of the previous version.
as	The new version is almost the same as the previous version.
d	The new version is very different from the previous version.
r	The old version has been removed (does not exist in the new version).

The **rc.uft** command has been extended to support the new functionalities, but SMIT is the recommended method for activating UFT.

FTAM

The **FTAM** API offers application source code compatibility.

Configuration is now performed using SMIT menus and CIS services, instead of the **ftcnf** command.

Compared to configuration using **ftcnf**, the new **FTAM** configurator uses objects with different names and meanings. The differences between these new and previous objects are described in the following sections.

Tools are provided (**isoadre_to_cis, isohosts_to_cis**) to migrate old configuration files (like **/etc/osi/iso***) to CIS. See the **/usr/lpp/ftamx/isofiles_to_cis/README** file after installing the new product. These tools are automatically launched at installation time to migrate the /etc/osi/iso* files if present.

Remote Site Object

A *remote site* object describes a remote application, including the AE Title, stack selectors, and the site name.

In the previous version of **FTAM**, the object was called a remote site in both the FTADM administrative interface and in the **ftcnf** configurator.

In the new version of FTADM, the object is still called a remote site, but in the FTAM CIS configurator it corresponds to an identification object, which contains the following information:

- AE Title (AP Title and AE Qualifier).
- Presentation, session, and transport selectors.
- The *FULL_ISO Site* name (if NSAPs are used), or the *PRE_ISO Site* name (for SNPAs).
- The session descriptor name, which holds session negotiated parameters.
- The transport profile name, which defines the network type.
- The Network Calling Facilities profile name, which describes X25 parameters.

Local Application

In the previous version of **FTAM**, a local application was identified by a local port and a local address.

In the new **FTAM** CIS configurator, there is no difference between a local or remote application. But FTAM can only use the default local applications that are delivered: ftam, ftam_snpa, ftamresp. These names cannot be changed.

Connection

Previously, a connection was a link between a local port and a remote site. In CIS, a connection occurs between two identification objects. A connection can also be *implicit* if two conditions are satisfied:

– Only NSAP addresses are used (not SNPAs).
– Local and remote sites each have only one NSAP address.

In this case, the calling and called site identification objects contain sufficient information to establish a route. If these conditions are not satisfied, the connection route must be explicitly defined using a Route Selection object.

X500

- The configurator now uses SMIT menus, but the configuration commands still exist.
- New example programs have been added.
- Applications using the XDS or XOM APIs must be recompiled due to changes in the operating system. Application source compatibility is assured.

CMIP Agent

- Previously, configuration was performed with the text file **resol.conf**.

  The new configuration interface is based on SMIT menus and CIS services.
- A new command **cmipinfo** is provided to use directly the **CMIP** browser.

OSI Miscellaneous Products Interface Changes

bullx25

New Command	
Command	**Comments**
xdping	Test HiSpeed WAN Comm. adapters

Modified Command		
Command	**Status**	**Comments**
funstat	as	A new option **–n** is available to test the state of the bullx25 licence.

NetShare

Command Changes

Because the **tpises** driver is no longer delivered, its menus have been removed from the **rfcstat** command. Also, the **osiconfgen** and **osiload** commands used for NetShare configuration on AIX Version 3.2.5 must not be used for configuration with NetShare versions on AIX Version 4.1.4.

File Changes

New Files	
File	**Comments**
/usr/bin/tpt6trl	Trace facility management.
/usr/bin/tpt6conf	Dynamic configuration management.
/usr/bin/osisnap.netshare	NetShare extension for **osisnap** support.

Modified Files		
File	**Status**	**Comments**
/etc/tsapd_sess	r	**tpises** driver management.
/etc/tsapd_sess.conf	r	**tpises** driver configuration management.
/usr/lib/drivers/osi/tpises/lib.a	r	**tpises** driver code.

r	The old version has been removed (does not exist in the new version).

ASN.1

Command Changes

A new command **asn1** has been added. It is a shell script that calls the **asn1pp** preprocessor and the **asn1c** compiler. Makefiles or shells scripts used to build applications may be updated to use this **asn1** command.

File Changes

The major file changes occur in the **asn1.h** header file.

Source files may be reused insofar as the calls to generated functions have not been modified and the differences relative to the generated code are managed by the *#define* statements of **asn1.h**.

Subroutine Changes

New Subroutines	
Generated Subroutines	**Comments**
Type_cmp	Performs a byte-to-byte comparison of two values of type *Type*.
Type_cpy	Copies source data of type *Type* to a target.
Type_hash	Returns a simple hash code based on a value of type *Type*.
Type_stp	Checks subtype values of type *Type*.
Generic Subroutines	**Comments**
asn1_int_oid	Converts an object identifier expressed as a table of integers in a compact INTEGER form.
asn1_oid_int	Converts an object identifier from a compact INTEGER form to a table of integers.

Hints on Porting ASN.1 Applications

Recompiling the ASN.1 Specifications

Rebuild the coding functions using the **asn1pp** and **asn1c** commands. Instead of these commands, you can also use the new **asn1** shell script to call the preprocessor and the compiler. If you use shell scripts or make files to build your applications, you may want to modify them to use the new **asn1** shell script.

Because the new preprocesor is less permissive than the previous version, previously undetected warnings may now be displayed.

Consider the following extract of an ASN.1 code specification, where line numbers have been added to the left:

```
5   Transfer-syntax-name ::= OBJECT IDENT
6   Presentation-context-identifier ::= INTEGER
7   PDV-list ::= SEQUENCE
8       { Transfer-syntax-name OPTIONAL,
9       Presentation-context-identifier,
10      presentation-data-values CHOICE
11          {single-ASN1_type  [0] EXPLICIT ANY,
12          octet-aligned      [1] IMPLICIT OCTET STRING,
13          arbitrary          [2] IMPLICIT BIT STRING
14          }
15      }
```

For the above code, the preprocessor produces the following warnings:

```
line 8:warning: This component should be named (OBJECT IDENT)
line 8:warning: Automatic allocation of name (x_0)
line 9:warning: Automatic allocation of name (x_1)
```

The C code is correctly generated using automatic naming of variables. To avoid warnings like these, correct the ASN.1 code.

Rebuilding the Application Libraries

Use the C compiler to compile ASN.1 compiler generated files. The makefile created by the **asn1c** command can be used as a model and adapted as necessary. Compilation errors may be detected in the following cases:

- The name of the ASN.1 module is not the same as the name of its file.
- The ASN.1 module name contains non-alphanumeric characters such as dashes (–), underlines (_), or colons (:).
- The C identifiers are too long. This may occur for identifiers automatically generated by concatenating lengthy ASN.1 identifiers or when the ASN.1 structures have many levels. The AIX C compiler has generous limits on identifiers, but other compilers may have restrictive limits. For example, ANSI C only requires support for the first 31 characters of identifiers. Older compilers and linkers may have limits as low as 7 characters.

The compiler constants that identify the target machine environment are generally set by the compiler itself. If not, you may need to set these constants yourself. As an example, the compiler flags for setting the constants for AIX on a DPX/20 are shown below:

```
-D_ANSI_C_SOURCE -D_AIX -D_POWER
```

Modify the Application

Adapt your application to the new C structure definitions. Most differences come from:

- Simplification of generated structures, where indirections are removed whenever possible.
- Use of portable C types (for example INTEGER produces an asn1_int32 type).

Portions of the application code which directly access the C structures (or their elements) for data transfer or analysis may require modification. However, the interfaces to coding functions (both names and parameters) have not changed.

Compile the application source files and link them with the ASN.1 object files and the new **libasn1.a** library. If you had to set compiler flags to identify the environment in order to build the application library, you should use the same flags when compiling the application source files.

Appendix A. Bibliography

This appendix gives information about the document references used in this guide.

- AIX Documentation on page A-1
- LIst of Related Documents on page A-1
- Standards on page A-2

AIX Documentation

For more information about AIX and related products such as iFOR/LS, refer to the following documents:

AIX Installation Guide	86 A2 60AP
AIX iFOR/LS System Management Guide	86 A2 11AQ
AIX iFOR/LS Tips and Techniques	86 A2 12AQ

List of Related Documents

The related documents for the OSI products described in this guide are as follows:

OSI Services Reference Manual	86 A2 05AQ
OSI Diagnostic Interactive Toolkit (ODIT) User's Guide	86 A2 39WG
Connection Information Service (CIS) Installation, Administration & User's Guide	86 A2 26AP
MAX25-3 API Programmer's Guide	86 A2 45AP
XTI/XX25 Administrator and User Guide	86 A2 04AP
XTI Porting Guide	86 A2 25AP
Session Access Library Programmer's Guide	86 A2 48WE
XAP API Reference Manual	86 A2 27AP
ACSE/P to XAP Interface Porting Guide	86 A2 46AP
ROSE Interface Library Programmer's Guide	86 A2 31AP
TPAD–HPAD Reference Manual	86 A2 65WE
CPI-C Administrator's Guide	86 A2 92QP
CPI-C Programmer's Guide	86 A2 91QP
UFT Reference Manual	86 A2 10CB
FTAM Installation, Administration & User's Guide	86 A2 03AQ
CMIP Agent for OSI Layers Installation, Administration & User's Guide	86 A2 87WG
HiSpeed WAN Comm. Installation and Service Guide	86 A1 81WG
NetShare User's Guide	86 A2 95AP
ASN.1 Compiler Programmer's Guide	86 A2 50WE
Communications Maintenance Tools	86 A2 36GX
BASIC-COM-MP and GCOS-COM-MP Installation Guide	86 A2 74AT
BASIC-COM and GCOS-COM Installation Guide	86A246AT

Standards

This section provides a list of standards, grouped by organization body:

ISO Standards

Refer to the bibliography of the *OSI Services Reference Manual* for a full list of ISO standards. The most important standards for the products described in this document are shown in the table below:

ISO 8326	Connection–oriented Session Service Definition
ISO 8822	Connection–oriented Presentation Service Definition
ISO 8649	Service Definition for the ACSE
ISO 9072–1	Remote Operations Part 1: ROSE Model, Notation and Service Definition
ISO 8878	Use of X.25 to provide OSI Connection–mode Network Service
ISO 8824	Abstract Syntax Notation One (ASN.1)
ISO 8571	OSI File Transfer, Access, and Management.

X/Open Standards

40 A2 49AS	*X/Open Transport Interface XPG4 CAE Specification Version 2*
86 A2 30AP	*ACSE/Presentation Services API (XAP) CAE Specification*

Request For Comments

The Request For Comments (RFCs) are listed in the following table:

RFC 1006	ISO Transport Services on top of the TCP Version 3. Marshall T. Rose, Dwight E. Cass, Northrop Research and Technology Center (May 1987).
RFC 1277	Encoding Network Addreses to support operation over non-OSI lower layers. S.E. Hardcastle–Kille, University College London (November 1991).

Glossary

This glossary contains the abbreviations, key words and phrases used in this document.

ACSE
Association Control Service Element: Single consistent means for establishing and terminating all associations.

ACSE/P
Abbreviation commonly used to designate the stack API used to access the stack ACSE and Presentation entities.

API
Application Programming Interface, allowing a high–level language application program to use specific data or services.

ASN.1
Abstract Syntax Notation One: A syntax commonly used for communication of data and type information.

CIS
Connection Information Service: An internal product and name server used by several OSI communications products.

CLNP
ConnectionLess Network Protocol: Protocol where no acknowledgement is returned to the originating source.

CLNS
ConnectionLess Network Service.

CMIP Agent
Short name for “CMIP agent for OSI layers”, a network management agent.

CONP
Connection Oriented Network Protocol.

CONS
Connection Oriented Network Service: Service with protocol provided by X.25.3 protocol.

COPP
Connection Oriented Presentation Protocol.

COSP
Connection Oriented Session Protocol.

COTP
Connection Oriented Transport Protocol.

CPI–C
Common Programming Interface for Communications: API allowing X/Open–compliant systems to communicate with systems implementing SNA or XCP2 protocols.

FTAM
File Transfer Access and Management: ISO file service function enabling user application processes to manage and access a file system.

HPAD
Host PAD, Server side in the PAD client/server model.

ICB
Internal Control Block.

iFOR/LS
Information For Operation Retrieval/License System.

LPP
Licensed Program Product

LSAP
Link Service Access Point.

MSDSG
Multi-System Distributed System Gateway.

NSAP
Network Service Access Point: A string of 20 bytes maximum that identifies a machine on a network.

NSDU
Network Service Data Unit.

NSEL
Network SELector.

OTM
Open Terminal Manager.

PAD
Packet Assembler Disassembler, Functional device enabling un–equipped Data Terminal Equipments to access a packet switching network.

PVC
Permanent Virtual Circuit, Virtual Circuit (X.25) with a logical channel permanently assigned to it at each Data Terminal Equipment.

RFC
Request For Comments. A process used to created industry standards, often used for TCP/IP.

RIB
Routing Information Base, Network database which contains all the required routing information to remote NSAPs. It gives the remote SNPA and local subnetwork to use for NSAPs or groups of NSAPs.

ROSE
Remote Operations Service Element.

SDU
Service Data Unit.

Session
Session protocol: Virtual relationship permitting communications between two network addressable units.

SMIT
System Management Interface Tool, Menu–driven command–building system management facility.

SNPA
Sub Network Point of Attachment: Information for accessing the system within the domain (Transpac or Ethernet address).

SSAP
Session Service Access Point.

SSDU
Session Data Unit.

SSEL
Session SELector.

SVC
Switched Virtual Circuit, requested by a virtual call and released when the call is cleared.

TCP–IP
TCP and IP are the two fundamental protocols of the Internet protocol suite. (Acronym for this suite). TCP provides reliable transfer of data, while IP transmits.

TPAD
Terminal PAD, Client side in the PAD client/server model.

TPDU
Transport Protocol Data Unit.

TPI
Transport Protocol Interface, an internal transport interface included in the lower stack layer module called **osi_low**.

TSAP
Transport Service Access Point.

UCB
User Control Block.

UFT
Unified File Transfer.

XAP
X/Open ACSE/Presentation Services API.

XTI
X/Open Transport Interface Definition.

X.25
CCITT recommendation defining an interface for connection to a packet–switched network with virtual circuit service. Defines the three lowest ISO layers: Electrical interface, HDLC procedure application, Packet structure.

X.25.3
Packet level protocol in layer three of X.25.

X.400
Message handling service. CCITT recommendation defining the message handling services which can be provided by telecommunications authorities to their subscribers enabling them to exchange messages in the store and forward mode.

X.500
Open System directory management service applying CCITT recommendations.

Index

Vos remarques sur ce document / Technical publication remark form

Titre / Title : Bull DPX/20 OSI Communications Porting Guide

Nº Reférence / Reference Nº : 86 A2 44AP 03

Daté / Dated : June 1996

ERREURS DETECTEES / ERRORS IN PUBLICATION

AMELIORATIONS SUGGEREES / SUGGESTIONS FOR IMPROVEMENT TO PUBLICATION

Vos remarques et suggestions seront examinées attentivement
Si vous désirez une réponse écrite, veuillez indiquer ci-après votre adresse postale complète.

Your comments will be promptly investigated by qualified technical personnel and action will be taken as required.
If you require a written reply, please furnish your complete mailing address below.

NOM / NAME : ____________________ Date : __________

SOCIETE / COMPANY : ____________________

ADRESSE / ADDRESS : ____________________

____

Remettez cet imprimé à un responsable BULL ou envoyez-le directement à :

Please give this technical publication remark form to your BULL representative or mail to:

Bull Electronics Angers S.A.
CEDOC
Atelier de Reprographie
331 Avenue Patton
49004 ANGERS CEDEX 01
FRANCE

PLACE BAR CODE IN LOWER
LEFT CORNER

Bull Electronics Angers S.A.
CEDOC
Atelier de Reprographie
331 Avenue Patton
49 004 ANGERS CEDEX 01
FRANCE

ORDER REFERENCE
86 A2 44AP 03

Utiliser les marques de découpe pour obtenir les étiquettes.
Use the cut marks to get the labels.

DPX/20
AIX
OSI
Communications
Porting Guide

86 A2 44AP 03

DPX/20
AIX
OSI
Communications
Porting Guide

86 A2 44AP 03

DPX/20
AIX
OSI
Communications
Porting Guide

86 A2 44AP 03