

1-1-1991

ISODE System Installation Manual

Margaret L. Loper

Find similar works at: <https://stars.library.ucf.edu/istlibrary>
University of Central Florida Libraries <http://library.ucf.edu>

This Research Report is brought to you for free and open access by the Digital Collections at STARS. It has been accepted for inclusion in Institute for Simulation and Training by an authorized administrator of STARS. For more information, please contact STARS@ucf.edu.

Recommended Citation

Loper, Margaret L., "ISODE System Installation Manual" (1991). *Institute for Simulation and Training*. 130.
<https://stars.library.ucf.edu/istlibrary/130>

INSTITUTE FOR SIMULATION AND TRAINING

May 1991

ISODE System Installation Manual

Margaret Loper • David Shen • Jack Thompson • Dr. Henry Williams

IST

Institute for Simulation and Training
12424 Research Parkway, Suite 300
Orlando FL 32826

University of Central Florida
Division of Sponsored Research

B 87

IST-TR-91-15

INSTITUTE FOR SIMULATION AND TRAINING

ISODE System Installation Manual

Margaret Loper • David Shen • Jack Thompson • Dr. Henry Williams

May 1991

IST-TR-91-15

Institute for Simulation and Training
12424 Research Parkway, Suite 300
Orlando FL 32826

**University of Central Florida
Division of Sponsored Research**

ISODE System Installation Manual

Written and Compiled by:

Margaret Loper
David Shen

Jack Thompson
Dr. Henry Williams

*Institute for Simulation and Training
University of Central Florida*

TABLE OF CONTENTS

LIST OF FIGURES	iii
LIST OF TABLES	iv
PREFACE	v
1.0 OVERVIEW	1
1.1 ISODE ORGANIZATION	3
1.1.1 Libraries	5
1.1.2 Databases	6
1.1.3 ISODE Directory Tree	7
1.2 SYSTEM ENVIRONMENTS FOR ISODE	9
1.2.1 Operating Systems	9
1.2.2 IST Configuration	9
2.0 INSTALLATION PROCEDURES	12
2.1 CONFIGURATION	13
2.1.1 Software	13
2.1.2 Hardware	16
2.2 COMPILATION	18
2.2.1 Generation	18
2.2.2 Compilation Time	19
2.2.3 Files Generated	19
2.3 INSTALLATION	20
2.4 POST-INSTALLATION	21
2.5 INSTALLATION TEST	23
3.0 PROBLEMS ENCOUNTERED	24
3.1 Differences in UNIX	24
3.2 Deamons	24
3.3 Bugs/Questions	25
4.0 SAMPLE PROGRAMS	26
4.1 RTF	27
4.1.1 Description	27
4.1.2 RTF Flow Diagram	29
4.2 LOOKUP	31
4.2.1 Description	31
4.2.2 Lookup Flow Diagram	34

5.0	DOCUMENTATION	36
5.1	THE ISO DEVELOPMENT ENVIRONMENT: USER'S MANUAL	36
5.2	DISCUSSION GROUPS	38
5.3	BOOKS	39
5.4	REQUEST FOR COMMENTS (RFCs)	39
5.5	ARTICLES	39
5.6	TUTORIALS	40
5.7	ISODE RELEASES	40
	REFERENCES	46
	APPENDIX A ISODE DIRECTORY DESCRIPTIONS	47
	APPENDIX B ISODE PORTS	53
	APPENDIX C ISODE TEST PROGRAM AND OUTPUT	64
	APPENDIX D ISODE BUG INFORMATION	71

LIST OF FIGURES

1.0-1	OSI TRANSPORT SERVICES ON TOP OF THE TCP	2
1.1-1	ELEMENTS OF ISODE 6.0	4
1.1.3-1	ISODE DIRECTORY TREE	8
1.2.2-1	THE IST ISODE NETWORK	10
1.2.2-2	NETWORKING AND COMMUNICATIONS TECHNOLOGY LABORATORY	11
2.1.2-1	ISODE MINIMUM REQUIREMENTS	17
4.1.2-1	FLOW DIAGRAM FOR THE RTF PROGRAM	30
4.2.2-1	FLOW DIAGRAM FOR THE LOOKUP PROGRAM	35

LIST OF TABLES

4.0-1	ISODE Demo Programs	26
-------	-------------------------------	----

PREFACE

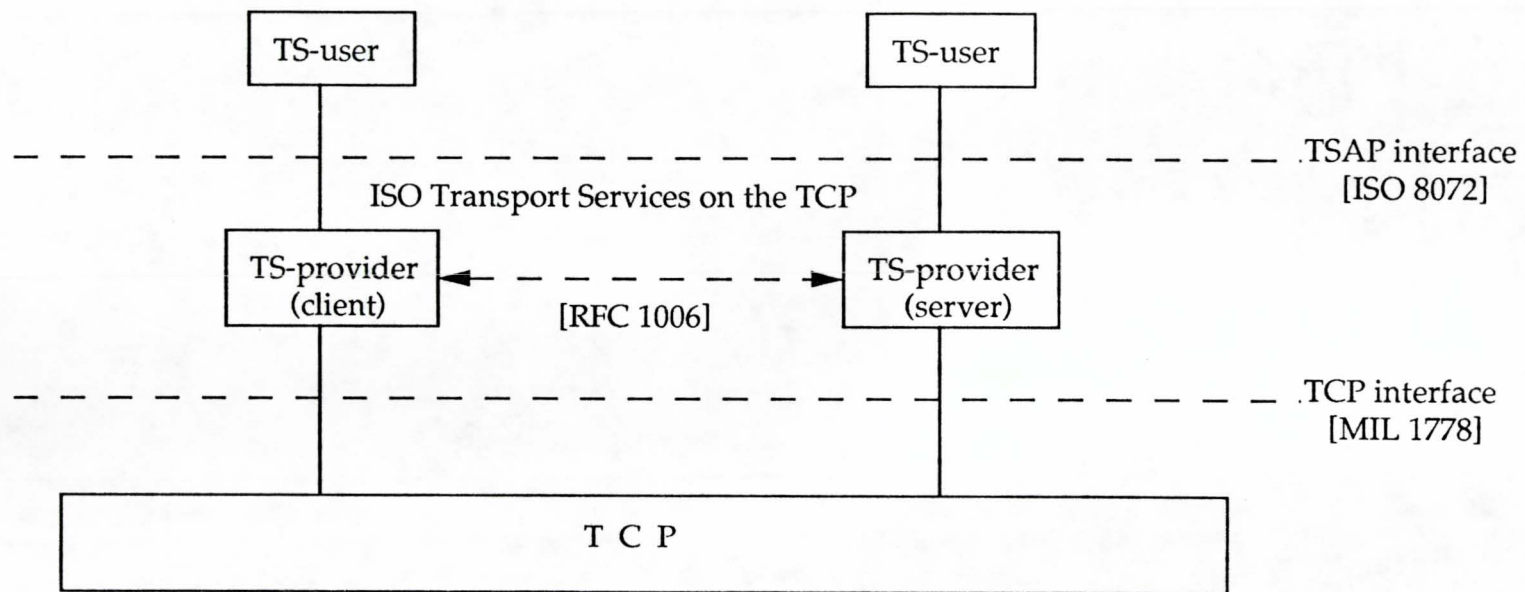
The Institute for Simulation and Training (IST) is currently involved in research aimed at investigating Open Systems Interconnection (OSI) network protocols to provide network communication services for real-time Distributed Interactive Simulation (DIS). As a part of this effort, IST has centered its research around the study and dissection of a public domain implementation of some of the protocols defined by ISO. This "quasi-OSI" software application is named the **ISO Development Environment (ISODE)**. This manual will describe how ISODE was obtained, its organization (libraries, databases, etc.), installation procedures, problems encountered, and available documentation. It is hoped that this document will assist the novice with the ISODE installation as well as provide the reader with insight about ISODE's organization and operation.

Revision 1.0, January 3, 1991

1.0 OVERVIEW

The ISO Development Environment (ISODE) is a non-proprietary software implementation of some of the protocols defined by the International Organization for Standardization and International Electrotechnical Commission (ISO/IEC), the International Telegraph and Telephone Consultative Committee (CCITT), and the European Computer Manufacturer Association (ECMA).

ISODE can support several different types of network services below the transport service access point (TSAP). One of these network services is the Department of Defense (DoD) Transmission Control Protocol (TCP), as shown in Figure 1.0-1. This implementation permits the development of higher level protocols in a robust and mature internet environment, while allowing the luxury of not having to recode anything when moving to a network where the OSI Transport Protocol (TP) is used to provide the TSAP. However, this software also operates over pure OSI lower levels of software.



2

Figure 1.0-1 OSI TRANSPORT SERVICES ON TOP OF THE TCP

1.1 ISODE ORGANIZATION

ISODE implements the upper layers of the ISO protocol stack on top of the TCP, as shown in Figure 1.1-1. These protocols are implemented through a set of libraries and databases written in the C programming language. The libraries implement application protocols and layer services, while the databases contain mappings of application entity information, object identifiers, service descriptions, and network and presentation addresses. A brief description of the libraries and databases follow, as well as a directory structure.

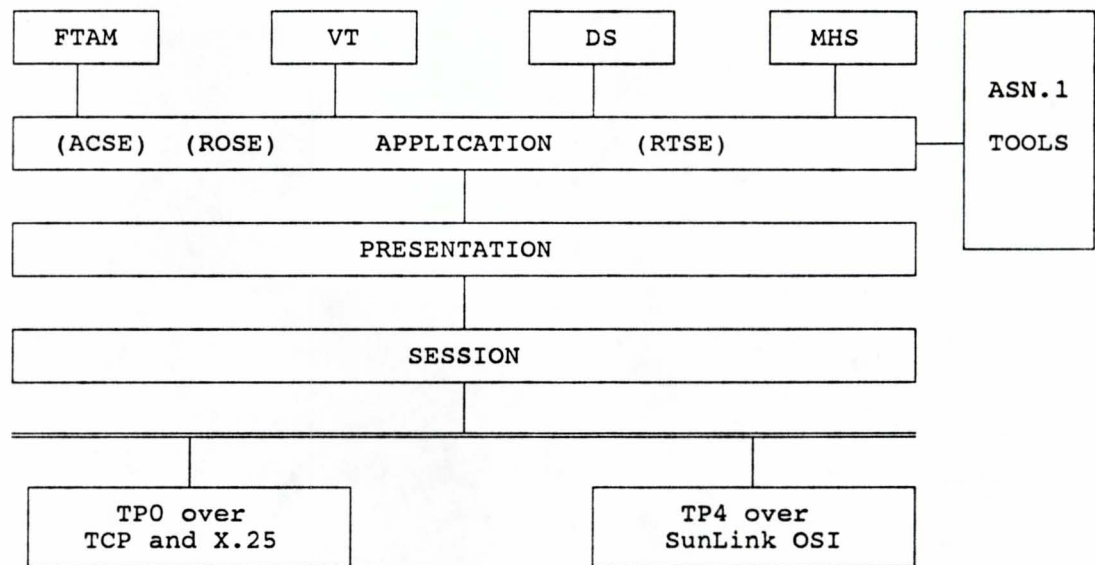


FIGURE 1.1-1 ELEMENTS OF ISODE 6.0

1.1.1 Libraries

Each library name follows the convention `libxxxx` where `xxxx` is the name of a particular ISODE service or facility. For example, the library `libacsap` implements the association control service (acs).

libacsap - implements the Association Control Service (ACS). The Association Control Service Element (ACSE) is concerned with starting and stopping the network for the application. That is, an application uses the ACSE to establish a connection, or an association, between two users. The association binds the two users, which are referred to as the initiator and responder.

libdsap - implements the Directory Access Protocol (DAP) for OSI applications wishing to look up information, or an interactive Directory user-interface.

libftam - implements the filestore-independent parts of the International Standard of the OSI file service, FTAM. Currently supported are: the no-recovery FTAM Quality-of-Service (QOS); the transfer and management service classes; the kernel, read, write, access, limited file management, enhanced file management, grouping, and fadu-locking functional units; and, the kernel, storage, and security attribute groups.

libcompat - is used as an aid for porting ISODE from one system to another.

libpepy - implements and manipulates presentation elements, from either a presentation stream or a presentation list.

libpsap - implements presentation syntax abstractions for machine-independent exchange of data structures. There are two objects which are manipulated: Presentation Elements (PE), which represent a particular, arbitrarily complex, data structure; and, Presentation Streams (PS), which represent an I/O path of these data structures.

libpsap2 - implements the presentation service. This library supports whatever session requirements the user wishes to employ, negotiates presentation contexts on connection establishments, and utilizes abstract transfer notations to transmit data structures in a machine-independent manner.

libpsap2-lpp - implements the lightweight presentation protocol for TCP/IP-based internets.

libquipu - provides routines for managing the data returned by the Directory User Agent (DUA) procedure calls.

librosap - implements the Remote Operations Service (ROS). Three service disciplines are implemented: the basic service discipline is used for the ECMA interpretation of ROS; the advanced service discipline implements the CCITT X.400 interpretation of ROS; and, the complete service discipline is used for the new ISO and CCITT MOTIS interpretation. Once an association is established, the initiator requests the responder to perform remote operations. The responder attempts these operations, returning either a result or error.

librosy - implements the run-time environment for applications using the distributed applications paradigm.

librtsap - implements the Reliable Transfer Service (RTS). RTS provides an application with an association for the reliable transfer of data.

libssap - implements the session service.

libtsap - contains a set of routines which implement the Transport Services Access Point (TSAP).

1.1.2 Databases

Each database name starts with **isoxxxx** where **xxxx** relates to the type of information contained in the database. For example, the database **isobjects** contains information about object descriptors and object identifiers.

isoaliases - contains a simple mapping between user-friendly strings and distinguished names. This database is used by the directory services element when accessing the "higher performance" nameservice.

isoentities - contains a simple mapping between application-entity information and presentation addresses. This database is used by the stub-directory service.

isomacros - contains a simple mapping between user-friendly strings and network addresses. This database is used to resolve textual representations of network addresses for use with the network.

isobjects - contains a simple mapping between object descriptors and object identifiers.

isoservices - contains a simple mapping between textual descriptions of services, service selectors, and local programs.

isotailor - contains a simple run-time configuration mechanism for programs loaded with the -lisode library.

isodocuments - contains a simple mapping between textual descriptions of FTAM document types and the various object identifiers which compose each document type.

1.1.3 ISODE Directory Tree

The ISODE software system is distributed as a collection of files which are organized into various directories. Each directory contains the files which are related by a common purpose or function. Figure 1.1.3-1 shows the tree structure for the ISODE directories. See Appendix A for a description of directories by function or purpose.

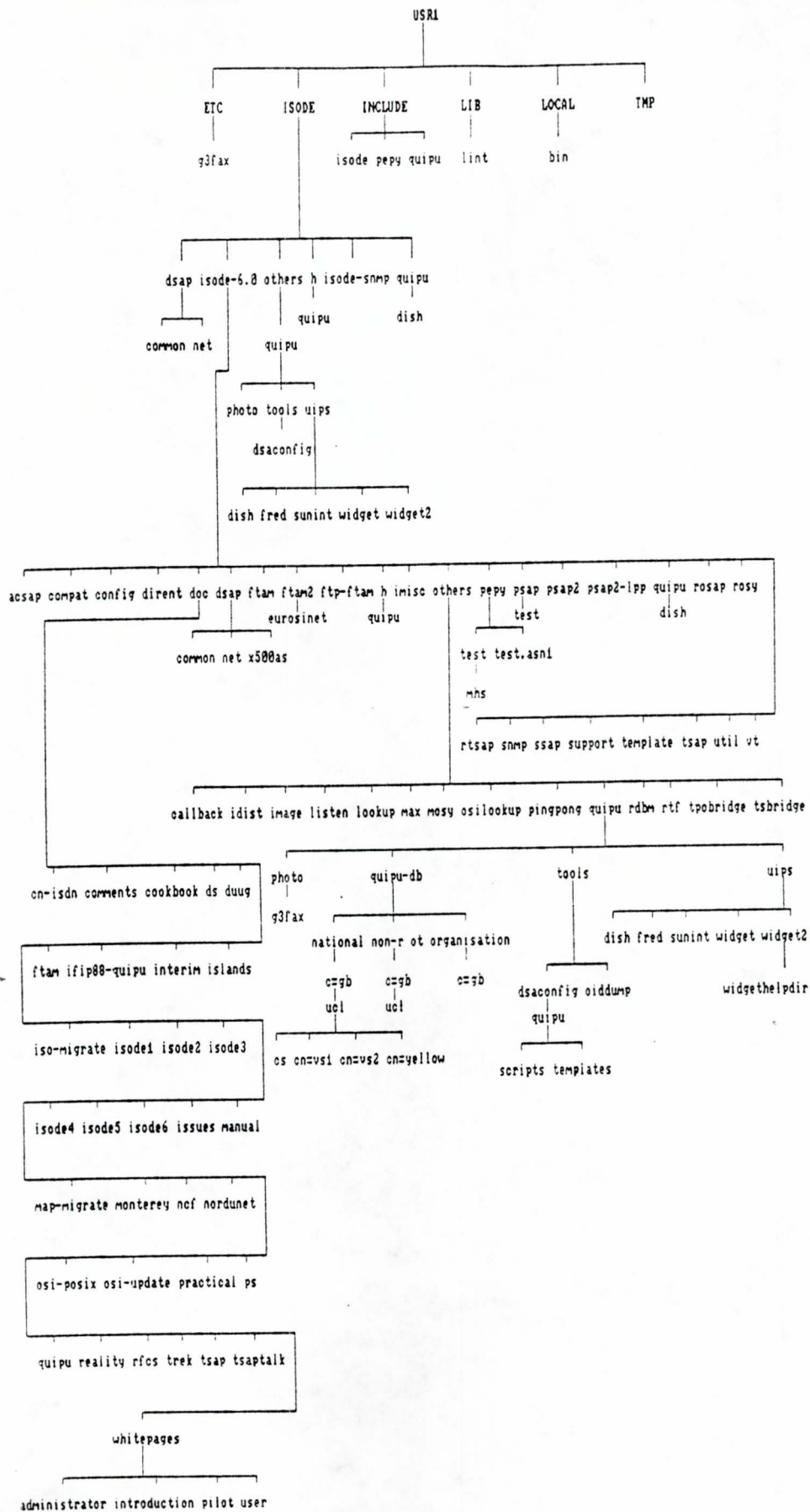


FIGURE 1.1.3-1 ISODE DIRECTORY TREE

1.2 SYSTEM ENVIRONMENTS FOR ISODE

1.2.1 Operating Systems Platforms

ISODE is written entirely in the C programming language and runs under the following operating systems: Berkeley UNIX, 4.2BSD, 4.3BSD, or 4.4BSD; AT&T UNIX, SVR2 or SVR3; AIX (IBM's UNIX-based derivative of AT&T System V); HP-UX (HP's UNIX-like operating system); ROS, the Ridge operating system; and, Pyramid OsX, a Pyramid computer running OsX. ISODE needs either a K&R or ANSI C compiler. There is no support for C++.

ISODE has also been recently ported to a DEC/VMS environment. The ASN.1 compilers, pepy, posy and rosy, have been ported also to a variety of different platforms, including a Macintosh. Information on these projects can be found in Appendix B.

1.2.2 IST Configuration

Currently, IST is running ISODE on two SUN SPARC 1 workstations running Berkeley UNIX and on two Motorola VME Development workstations running AT&T System V UNIX. These computers are connected on the Ethernet network in the Networking and Communications Technology Laboratory (NCTL). The computers running ISODE are shown in Figure 1.2.2-1, which are a part of the NCTL network that is shown in Figure 1.2.2-2.

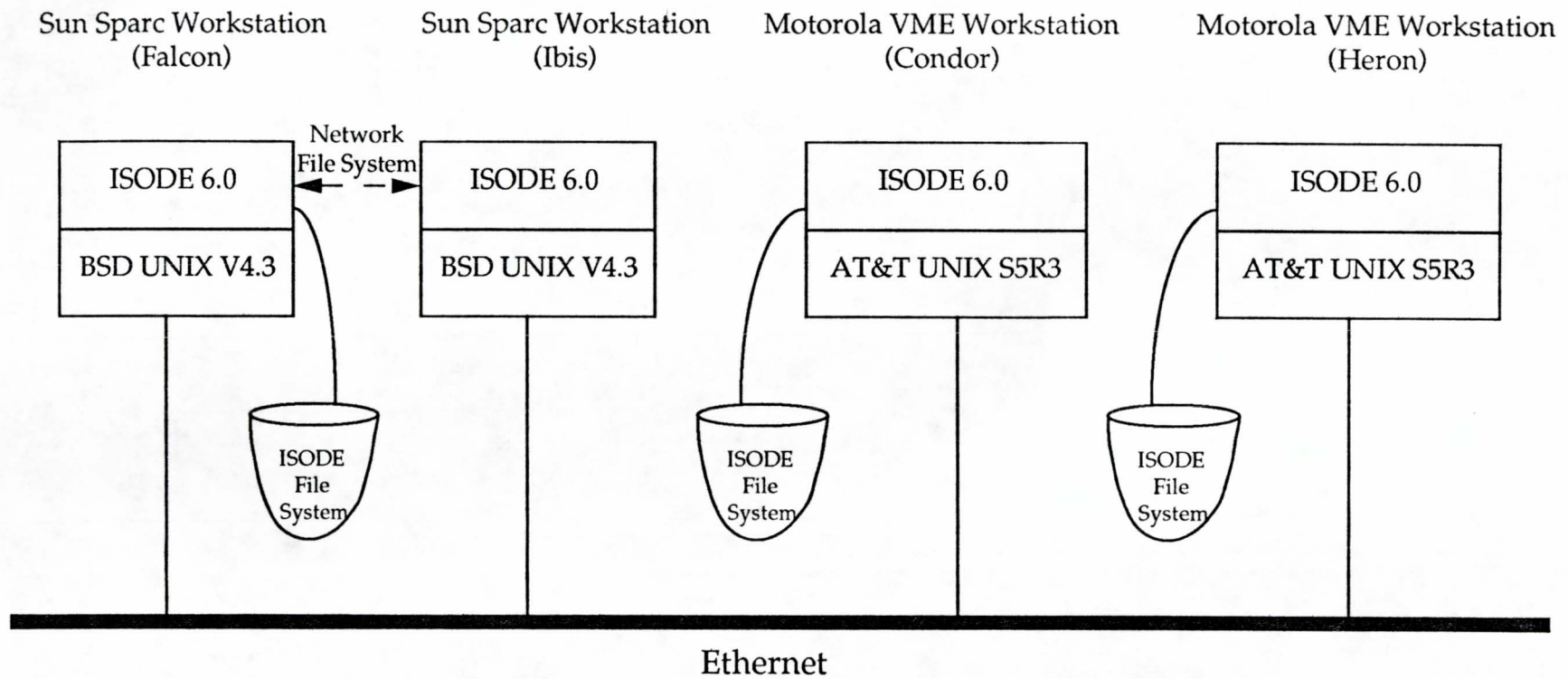


Figure 1.2.2-1 The IST ISODE Network

*NCTL's Internal
Developmental Ethernet*

Realtime (SIMNET) Ethernet

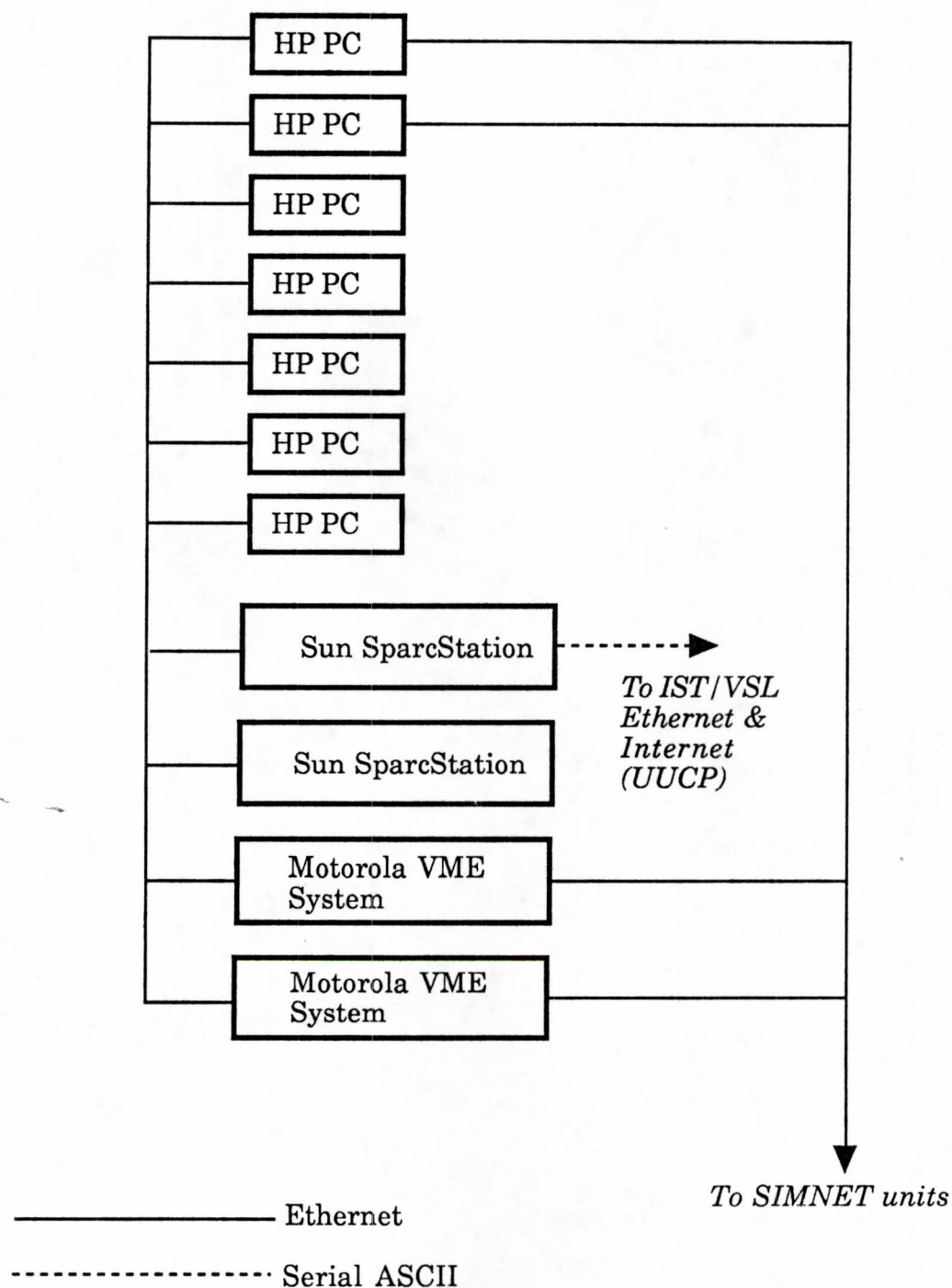


FIGURE 1.2.2-2

Networking and Communications Technology Laboratory

2.0 INSTALLATION PROCEDURES

A set of installation procedures are included with the ISODE and are located in `usr1/isode/isode-6.0/README`. This file includes instructions to configure, generate, and install ISODE. The following sections elaborate on those instructions. However, the user is advised to review the README file along with the installation procedures.

According to Marshall Rose, there are some common "Gotcha's". These include: C compiler/loader must handle names that are 1) longer than 32 characters, 2) non-unique in first 32 characters, and 3) pre-process must handle over 3000 defines (also see Section 3.0 Problems Encountered); need `fork` and `exec` to do dynamic servers; need synchronous multiplexing (e.g., `select`); need dynamic memory allocation (e.g., `malloc`); and, need `make` with 1) `-f` file capability and 2) `sh expr` capability.

2.1 CONFIGURATION

2.1.1 Software

The following sections describe the ISODE installation procedures on a UNIX-based computer. UNIX commands such as `cd`, `ln`, `%`, etc., will be used. However, these commands should be replaced with the appropriate commands for the actual operating environment being used.

The ISODE software is configured for a system by completing the following set of steps in the indicated order.

Step 1: Go to the config/ directory

`% cd config`

Step 2: Select the Makefile and include-file skeleton which most closely matches your system. The current choices are :

<u>FILE</u>	<u>CONFIGURATION</u>
apollo	Apollo
aux	A/UX release 1.1
bsd42	generic 4.2BSD UNIX
bsd43	generic 4.3BSD UNIX
bsd43-rt	RT/PC with 4.3BSD
bsd44	4.4BSD UNIX with OSI
hpux	HP-UX
mips	MIPS RISC/OS
osx	Olivette LSX 30xx
ros	Ridge Operating Syetm
sunlink3	SunOs release 3 with SunLink OSI release 5.2
sunlink4	SunOs release 4 with SunLink OSI release 6.0
sunos3	SunOs release 3

sunos4	SunOs release 4
sys52-exos	SVR2 UNIX with EXOS
sys52-rt	RT/PC with AIX
sys52-sun	SVR2 UNIX emulation on SunOs release 3
sys52-win	SVR2 UNIX with WIN/TCP
sys53	generic SVR3
ultrix	Ultrix 3.1

The Makefile skeleton has the extension `.make`, whereas the include-file skeleton has the extension `.h`.

Step 3: Copy the Makefile skeleton of your choice to **system.make**, where "system" is the name of your system. Now edit this file to set the following make variables:

<u>VARIABLES</u>	<u>DEFAULT</u>	<u>SPECIFIES</u>
BINDIR	/usr/local/bin	where to install user programs
ETCDIR	/usr/etc/	where to install administrator files
INCDIR	/usr/include/isode/	where to install include files
LIBDIR	/usr/lib/	where to install object libraries
LINTDIR	/usr/lib/lint/	where to install lint libraries
LOGDIR	/usr/tmp/	where to install log files
LSOCKET		libraries to link in (e.g., -lcci)
MANOPTS		see util/inst-man.sh for detail
OPTIONS		options to cc and lint(e.g., -l../h)
SBINDIR	/usr/etc/	where to install administrator pgms

Note: For our application, IST has installed the ISODE under directory `usr1`.

Step 4: Type:

In `system.make` **CONFIG.make**

Both of these files are in the `usr1/isode/isode-6.0/config/` directory. The latter file is the one which the software uses to configure itself during generation.

Step 5: Copy the include-file skeleton of your choice to **system.h**, where "system" is the name of your system. Now add any additional definitions you like (usually none). Consult the file **config.OPTIONS** for a list.

Step 6: Now,

% In **system.h** ../h/config.h

The latter file is the one which the software uses to configure itself during generation.

Step 7: Typically, sites run with the default databases used by the OSI directory. These databases include: aliases, services, entities, macros, and objects. To use the default databases, simply copy the default local configuration file to the support/ directory:

```
% cp aliases.local ../support/  
% cp services.local ../support/  
% cp entities.local ../support/  
% cp macros.local ../support/  
% cp objects.local ../support/
```

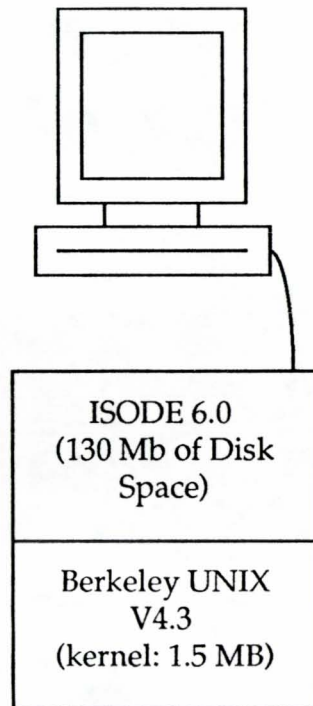
If you have local modifications to make, either copy in your own file or edit the existing file (ex: **support/aliases.local**) as appropriate.

There are a few differences between AT&T and Berkeley UNIX and also some bugs found in version 6.0. It is advisable that you read Section 3.0 (Problems Encountered) before proceeding with the installation.

2.1.2 Hardware

There is no specific hardware configuration in which ISODE can be installed, although there is a set of minimum requirements. A 4.5 Mb hard-disk is required for the initial download of the compressed ISODE system, which decompressed is 14Mb. After ISODE has been compiled and installed, it will require 130 Mb of memory on a SUN Sparc Station with Berkeley UNIX or 275 Mb on a Motorola VME with AT&T UNIX. These minimum requirements are shown in Figure 2.1.2-1.

Sun Sparc Workstations



Motorola VME Workstations

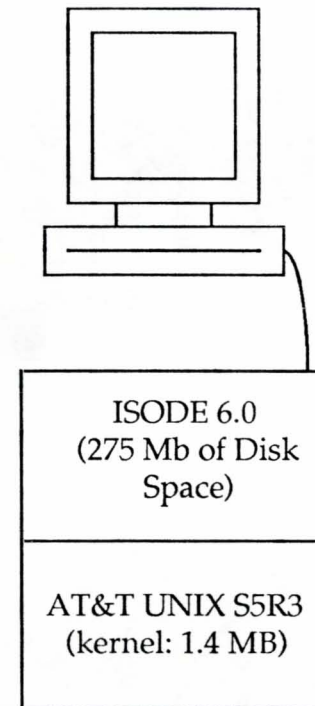


Figure 2.1.2-1 ISODE Minimum Requirements

2.2 COMPILATION

2.2.1 Generation

Once the ISODE has been configured for your system, it must be compiled.

The following commands are used to generate the ISODE code:

Step 1: Go to the `usr1/isode/isode-6.0/` directory,

```
% cd usr1/isode/isode-6.0
```

Step 2: Now reset the date of the configuration files for the system. This is done only once per source-tree.

```
% ./make once-only
```

Step 3: Generate the basic system,

```
% ./make
```

If you are using SunOs, do not use the make program supplied with the SunPro package. Contrary to any claims, it is not compatible with the standard make facility. Further, note that if you are running a version of SunOs 4.0 prior to release 4.0.3, then you may need to use the make program found in `/usr/old/`, if the standard make you are using is the SunPro make. In this case, you will need to put the old, standard make in `/usr/bin/`, and you can keep the SunPro make in `/usr/bin/`.

If you are using SVR3, then you will probably have to type this command before starting the compilation:

```
% ulimit 32768
```

The make command from the top-level directory will cause a complete generation of the basic system. If all goes well, proceed with the installation.

While compiling, some files may produce the messages:

```
warning: statement not reached or
```

type ObjectDescriptor: Warning: Can't find file DSE.ph failed

This is normal. Also, when building a loader library, you may see several

ranlib: warning: ../libisode.a(aetdbm.o): no symbol table

messages. This is also normal. Messages like:

*** Error code 1, or perhaps Exit

may also appear. Again, this is normal.

2.2.2 Compilation Time

For a Sun Sparc Station 1, the compilation time is about 45 minutes. For a Motorola VME System with a MC68030 microprocessor, the compilation time is about 3 hours.

2.2.3 Files Generated

The following files and libraries are generated on the top-level of the source

tree:

FILES

libacsap.a
libcompat.a
libdirent.a
libisode.a
libpepy.a
libpsap.a
libpsap2.a
librosap.a
librosy.a
librtsap.a
libssap.a
libtsap.a

LIBRARIES

llib-lacsap
llib-lcompat
llib-ldirent
llib-lisode
llib-lpsap
llib-lpsap2
llib-lrosap
llib-lrosy
llib-lrtsap
llib-lssap
llib-ltsap

Many other files and libraries are created within each of the basic system subdirectories. The basic system is composed of the files in the following subdirectories:

acsap	psap2
compat	rosap
dirent	rosy
dsap	rtsap
h	ssap
imisc	support
pepy	tsap
psap	

2.3 INSTALLATION

This phase requires the SuperUser capabilities to be executed.

Step 1: First, create the directories defined in the **system.make** file, i.e., BINDIR, SBINDIR, ETCDIR, LOGDIR, INCDIR, LIBDIR and LINTDIR. The mode of the LOGDIR directory should be 777, the mode of the other directories should be 755.

Step 2: Then, from the **usr1/isode/isode-6.0/** directory, type
% ./make inst-all
which will initiate the installation.

2.4 POST-INSTALLATION

After installation, there are two kinds of activities that occur: once-only activities, which are performed the first time the software is installed; and each-time activities, which are performed every time the software is installed. The following once-time and each-time activities should be initiated after ISODE has been installed:

Step 1: The tsap daemon, tsapd, is a process that runs in the background of UNIX and handles all incoming connections for ISODE. It is necessary for two machines in the network to be able to communicate with each other using the ISODE connection services. The first once-only activity is to verify that the tsapd daemon will run when the machine goes multi-user. On Berkeley UNIX systems, add these lines to the `usr/etc/rc.local` file:

```
if [ -f $(SBINDIR)tsapd ]; then
    $(SBINDIR)tsapd & (echo -n ' tsapd') > /dev/console
fi
```

On other systems, a similar procedure is followed. For example, on systems derived from AT&T UNIX, the file `usr/etc/rc2` script might be edited.

Step 2: The next once-only activity is to verify that systems with a native `usr1/etc/services` file contain an entry for the tsap service (if you have configured the ISODE to run over TCP). If not, add the line:

```
tsap                102/tcp
```

to the `usr/etc/services` file. If your system does not have such a file, the software automatically compensates for this.

Step 3: Next, on Berkeley UNIX systems, add a line to the `/usr/lib/crontab` file to invoke a shell-script that will re-cycle the log files. Usually the line you add looks something like this:


```
0 4 * * * su daemon < (SBINDIR)isologs
```

This means that the shell-script `$(SBINDIR)isologs` should be invoked at 4:00am each morning. On other systems, a similar procedure is followed. For example, on systems derived from AT&T UNIX, the file `/usr/spool/cron/crontabs/root` might be edited followed by the command

```
% crontab root
```

Step 4: The last once-time activity is the installation of manual pages. If you are installing the manual pages into a non-standard area, then set the environment variables `$MANDIR` accordingly, e.g.,

```
# sentenv MANDIR /usr/local/man
```

Step 5: The each-time activity is to kill and restart the `tsapd` daemon in order to reinitialize correctly. To start the daemon from the CShell, the command might be:

```
# $(SBINDIR)tsapd > & /dev/null
```

The daemon will automatically detach. If you do not redirect the daemon's standard-error, then it will not detach. Instead, the daemon will print messages as to what actions it is taking.

That's about it. This will install everything.

Step 6: To clean-up the source tree, use:

```
% ./make clean
```

Note: If you are planning to generate or install FTAM, VT, or QUIPU, do not clean-up the source tree until after you have finished installing these.

Each time an initiator or a responder program is compiled, it needs to be installed into its proper subdirectory (`$(SBINDIR)/local/bin` or `$(SBINDIR)/etc` respectively), otherwise the system will not be able to execute it (see step 3 of section 2.1.1 for SBINDIR definition).

There are additional programs in the `isode/isode-6.0/others/` directory. These programs are not an integral part of the system and they assume that the ISODE has been installed. Two of these demo programs are described in Section 4.0 (Sample Programs).

2.5 INSTALLATION TEST

Before proceeding with the installation test, be sure to read Sections 2.1 thru 2.4 for configuration and installation procedures. The `isode-test` program is located in directory `usr1/isode/isode-6.0/`. The program is initiated by typing `isode-test` while in the `/isode-6.0` directory. By default, two services are installed. The first service uses program `imisc` to support utilities such as `time`, `finger`, `who`, `ping`, and so forth. The second service uses program `isoc` to test the installation of the ISODE. The script `isode-test` uses `isoc` to verify the proper connection of the system. A copy of the `isode-test` program and its output can found in Appendix C.

3.0 PROBLEMS ENCOUNTERED

3.1 Differences in UNIX

One problem was identified when compiling the ISODE on machines running System V UNIX. Many of the files in the basic system do not include the header file `internet.h` (UNIX standard library), therefore during compilation errors resulted. The solution was to add the line `#include internet.h` to those files. A list of those files is located in Appendix D.

Another problem with implementing ISODE on System V UNIX is the maximum length allowed for file names. System V UNIX limits names to 14 characters. During the ASN.1 compilation, files with names longer than 14 characters are truncated, causing duplicate file names. The solution was to shorten the names of the files before compiling them.

3.2 Daemons

The `/dev/null` is a special directory where all the message dumps directed to it are discarded without comment. The output of the `tsapd` daemon should be directed to `/dev/null` when it is run manually, this way the output will not be displayed on the screen. If it is desired that the output be displayed on the screen, then the daemon should be killed before leaving the terminal where the daemon was initiated. This allows the next user the option of redirecting the output.

3.3 Bugs/Questions

Several bugs have been found in version 6.0 of ISODE. These bugs were made apparent by other users of ISODE through the ISODE mail reflector (see Section 5.2 for reporting ISODE bugs). Problems and questions with the code can be found in Appendix D.

4.0 SAMPLE PROGRAMS

Several demo programs are included with the ISODE software. These programs are not an integral part of the system and they assume that the ISODE has been installed. The following sections describe two of the available demo programs. These programs emphasize reliable transfer and remote operations.

The demo programs are located in the `usr1/isode/isode-6.0/others` directory and are listed in Table 4.0-1.

ISODE DIRECTORY	DESCRIPTION OF DEMO PROGRAM
callback/	The ISODE callback demo
idist/	The ISODE idist utility
image/	The ISODE image utility
listen/	The ISODE listen demo
lookup/	The ISODE password lookup demo
max/	Directory browser (X-based)
mosy/	Managed object syntax-compiler (yacc-based)
osillookup/	ISODE/SunLink OSI conversion
pingpong/	Asynchronous connection establishment
quipu/	Utility directory for QUIPU directory
rdbm/	The ISODE remote DBM demo
rtf/	RT-file transfer utility
tsbridge/	The ISODE TS bridge
tp0bridge/	The ISODE TCP/X.25 TP0 bridge

Table 4.0-1 ISODE Demo Programs

4.1 RTF

The Reliable Transfer File (RTF) program utilizes the Reliable Transfer service implemented in ISODE to perform bulk data transfer from one host to another.

4.1.1 Description

The RTF program is invoked by typing `rtf <host> <get/put> <source> <destination>`, with the user supplying the full path to the source and destination file. This transfer process is accomplished through an initiator, which sends/receives a file to/from the remote host using the put/get commands, and a responder, which requests and checks the initiator's password to ensure proper access to the remote host.

The initiator performs four operations: association establishment, operation invocation, association release, and error handling. During association establishment, the application-entity information and presentation address for the desired service are computed, along with the application context and default presentation context information for the service. If the initiator is to send (or put) a file, the source file is opened and the tsap daemon is contacted using the **RtBeginRequest** routine. If the initiator is to receive (or get) a file, the tsap daemon will be contacted immediately to invoke the responder. If an association is established with the responder, and the initiator is sending a file, the transfer (or operation) is initialized using the **RtSetDownTrans** routine. The

RtTransferRequest routine is then called to request the reliable transfer of a data structure. When the transfer is complete, the association is released using **RtEndRequest**.

If the initiator is to receive the file from the remote host, the operation is initialized by the **RtSetUpTrans** routine. The initiator will then wait for a transmission from the responder with the **RtWaitRequest** routine. At this point, transmission occurs by managing the turn (or token) between the initiator and responder. The user with the token is permitted to send data to the other user. Two routines are used to synchronize this communication; **RtGTurnRequest** is used to pass the token to the responder and **RtPTurnRequest** is used to ask for the token from the responder. When transmission is complete, the association is released using the **RtEndRequest** routine.

Any time an error is encountered, an **adios** or **advise** routine will report the error and terminate if appropriate.

The responder is responsible for three functions: association management, operation response, and error handling. The first action performed by the responder is to set up the association using the **RtBInit** routine. On a successful call, the responder should either accept or reject the association. This is accomplished using the **RtBeginResponse** routine. If the association is

rejected, the association is released and the program terminates, otherwise association establishment is completed. If the responder is sending the file, the source file is opened and the transmission is initialized by the **RtSetDownTrans**. When the user has the token, it can use the **RtTransferRequest** routine to request the reliable transfer of a data structure. If the responder is receiving the file from the initiator, the transmission is initialized by the **RtSetUpTrans** routine. The responder waits for an event (usually incoming data) to occur with **RtWaitRequest**. Transmission occurs by managing the turn (or token) between the initiator and responder. When the transfer is complete, the association is released by the **RtEndResponse** routine.

Again, error handling is accomplished by the **adios** and **advise** routines. These functions report errors and terminate if appropriate.

4.1.2 RTF Flow Diagram

The flow diagram for the RTF program can be found in Figure 4.1.2-1.

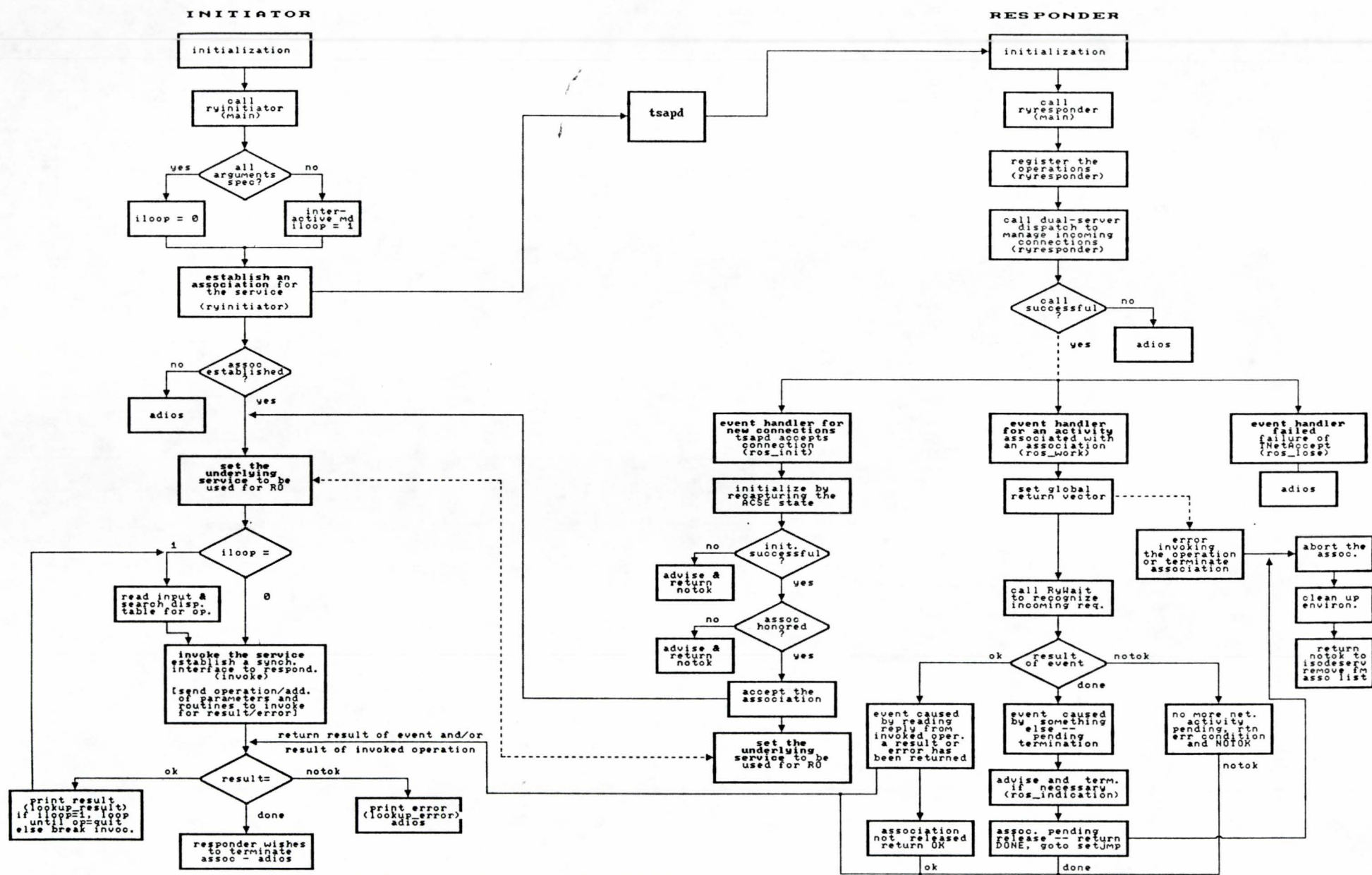


Figure 4.2.2-1 Flowchart for Lookup Program

4.2 LOOKUP

The Lookup program makes use of the remote operations services implemented in ISODE. This program requests password information, **UserName** or **UserID**, from remote nodes. If the password information is located in the remote node's password file, it is returned to the initiating node and displayed on the screen. If it is not found, an error message is returned and displayed.

4.2.1 Description

The Lookup program can be invoked several ways: by typing **Lookup**, which will put the user in an interactive loop allowing the lookup of more than one password; by typing **LookupUser username**, which allows the user to access password information for one username; or by typing **LookupUID userid**, which allows the user to access password information for one userid. If the user initiates the program with **Lookup**, the operations **help** and **quit** are also available. This lookup process is accomplished through an interactive initiator, which manages the association and invokes the operations required, and a static responder, which performs the operation.

The initiator performs four operations: association establishment, operation invocation, association release, and error handling. During association establishment, the application-entity information and presentation address for the desired service are computed, along with the application context and

default presentation context information for the service. Also, a session reference identifier is chosen. This is done in the **ryinitiator** routine using the ISODE **AcAssocRequest** routine. At this time, the tsap daemon is contacted to invoke the responder. If an association is established with the responder, the underlying service to be used for the remote operation (the presentation service) is set using routine **RoSetService**.

At this time, either the interactive loop is entered or the lookup operation is invoked. If the interactive loop is entered, a line is read from the input and a search is performed to determine which lookup operation is requested. The invocation is performed through a synchronous interface implemented in routine **RyStub**. The operation will return one of three results: error, done, or the result of the lookup operation. The result of the operation (password or error) is displayed on the screen, and the association is released. If the user has chosen the interactive loop, another username or userid can now be entered. The association will be released when the operation invoked is quit.

Any time an error is encountered, an **adios** or **advise** routine will report the error and terminate if appropriate.

The responder is responsible for three functions: association management, operation response, and error handling. Association management is

implemented in the routine **ryresponder**. After initializing the invoked program, each lookup operation (i.e., **LookupUser**, **LookupUserID**, **Help**, and **Quit**) is registered with the ISODE **RyDispatch** routine. The routine **isodeserver** is then called to set the addresses of event-handlers and to manage any associations. If the call to **isodeserver** is successful, then the program may terminate immediately.

When an event associated with a new connection occurs, the event-handler **ros_init** will be invoked. This routine first calls **AcInit** to re-capture the Association Control Service Element (ACSE)-state. If the initialization is successful, the routine **AcAssocResponse** is called to deal with the incoming association from the initiator. If the association was accepted, the underlying service for remote operations is set using the **RoSetService** routine.

If activity associated with an association occurs, the event-handler **ros_work** will be invoked. This routine sets a global return vector using **setjmp(3)** and then calls ISODE routine **RyWait** to poll for the next operation-related event. This usually results in one of the registered operations being performed (i.e., **LookupUser**, **LookupUserID**, **Help**, or **Quit**). Next, the operation is attempted. If it is successful, a result is returned to the initiator by the **RyDsResult** routine. Otherwise, the error is returned by the **RyDsError** routine. The **RyWait** will then indicate that no more network activity is pending. If extraordinary conditions

exist for the association, routine **ros_indication** is called. This routine will process any errors that have occurred, cause control to return to the **setjmp** call, and terminate the association.

The **isodeserver** routine uses the **TNetAccept** routine to wait for the next event on existing associations and for new connections. If failure occurs during this operation (i.e., network listening fails), the **ros_lose** routine will be advised. This routine logs the error condition and terminates the operation with one of the **adios** or **advise** routines.

4.2.2 Lookup Flow Diagram

The flow diagram for the Lookup program can be found in Figure 4.2.2-1.

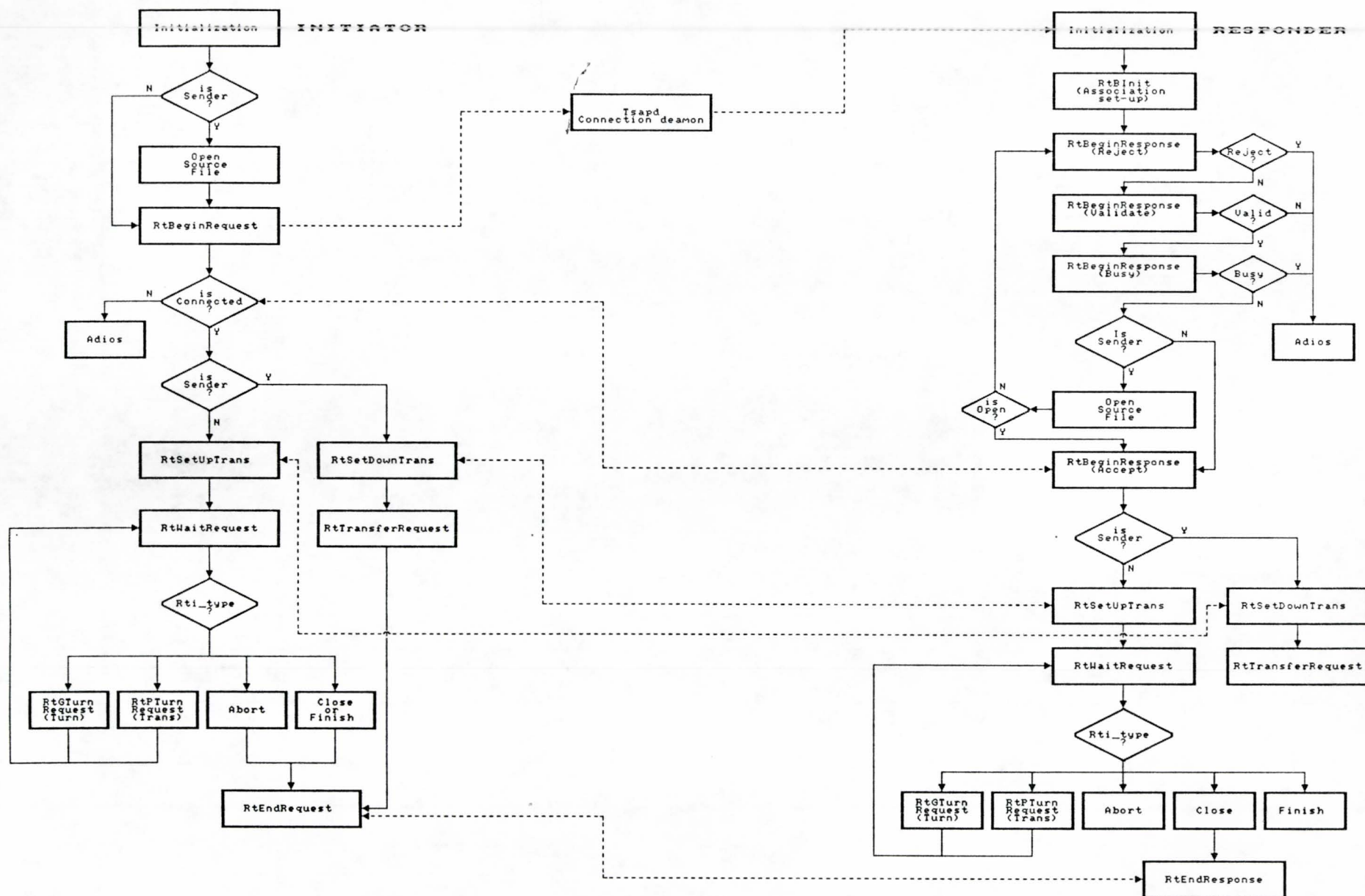


Figure 4.1.2-1 Flowchart of RTF Program

5.0 DOCUMENTATION

There is very limited documentation available for ISODE. The following sections list all known sources. Included in the ISODE software are five User Manuals, a set of UNIX manual pages, and a number of notes, papers, and presentations. This documentation is in LaTeX or S_{Li}TeX format and is located in the `usr1/isode/isode-6.0/doc/` directory. It is also rumored that Marshall Rose has plans to write an ISODE book.

5.1 THE ISO DEVELOPMENT ENVIRONMENT: USER'S MANUAL

There are five volumes of documentation available for ISODE. The User's Manuals are in LaTeX format and are located in the `usr1/isode/isode-6.0/doc/` directory. A hardcopy of these manuals can also be obtained from the University of Pennsylvania for \$375.00 (see Section 5.7).

In **Volume One: Application Services**, the raw facilities available to applications are described. This includes the following libraries: `libacsap(3n)`, which implements the OSI Association Control Service (ACS); `librosap(3n)`, which implements different styles of the OSI Remote Operations Service (ROS); `librtsap(3n)` which implements the OSI Reliable Transfer Service (RTS); and, `libpsap(3)`, which implements the OSI abstract syntax and transfer mechanisms.

Volume Two: Underlying Services, describes the services upon which the application facilities are built. This is accomplished with the following three libraries: **libpsap2(3n)**, which implements the OSI presentation service; **libssap(3n)**, which implements the OSI session service; and, **libtsap(3n)**, which implements an OSI transport service access point. The library **libpsap2-lpp(3n)** is also described, which can replace library **libpsap2(3n)** when implementing the lightweight presentation protocol for TCP/IP-based internets. This volume also contains information on how to configure the ISODE for your network.

Application programs are contained in **Volume Three: Applications**. The applications described include: an implementation of the ISO File Transfer, Access and Management (**FTAM**), which runs on Berkeley or AT&T UNIX; an implementation of an FTAM/File Transfer Protocol (FTP) gateway, which runs on Berkeley UNIX; an implementation of the ISO Virtual Terminal (**VT**), which runs on Berkeley UNIX; an implementation of the "little services", which are often used for debugging; and, an implementation of a simple image database service.

In **Volume Four: The Applications Cookbook**, a "cooked" interface for applications using remote operations is described. The following programs and library are described: the **rosy(1)** compiler, which is a stub-generator for specifications of Remote Operations; the **posy(1)** compiler, which is a structure-

generator for ASN.1 specifications; the **pepy(1)** compiler, which reads a specification for an application and produces a program fragment which is communicated by that application; and, the **librosy(3n)** library, which is used by applications using the distributed applications paradigm.

In **Volume Five: QUIPU**, the QUIPU directory is described. This currently consists of the following programs and library: **quipu(8c)**, which is a program for the Directory System Agent (DSA); the **dish(1c)** family of programs, which are a set of Directory SHell commands; and, **libdsap(3n)**, which is a library for applications using the Directory.

5.2 DISCUSSION GROUPS

The Internet discussion group **ISODE@NIC.DDN.MIL** is used as a forum to discuss ISODE. To be added to this list, contact the Internet mailbox **ISODE-Request@NIC.DDN.MIL**. To report bugs use **BUG-ISODE@NISC.PSI.NET**. There is also a discussion group for QUIPU users. To be added to this list, contact **QUIPU-Request@CS.UCL.AC.UK**. X.400/ISODE users can be added to a discussion group by contacting **PP-PEOPLE-Request@CS.UCL.AC.UK**.

5.3 BOOKS

The author of ISODE, Marshall Rose, has written the definitive book on OSI, entitled **The Open Book, A Practical Perspective on OSI**. This publication focuses on the pragmatic aspects of OSI: what OSI is, how OSI is implemented, and how OSI is integrated with existing networks. The book also provides examples and references to the ISODE.

5.4 REQUEST FOR COMMENTS (RFCs)

There are three RFCs which further document services utilized by the ISODE. The Lightweight Presentation Protocol for TCP/IP-based internets, as described in Volume Two: Underlying Services, is further documented in **RFC1085**. The implementation of ISO transport services on top of the TCP is further documented in **RFC1006**. Documentation on the ISO-TPO bridge between TCP and X.25 can be found in **RFC1086**. These RFC's are located in the `usr1/isode/isode-6.0/doc/rfcs/` directory.

5.5 ARTICLES

Actual documentation on ISODE is limited to the five user's manuals, however, Marshall Rose has written two articles about ISODE. These articles, entitled **ISODE: Horizontal Integration in Networking and Building Distributed**

Applications in an OSI Framework describe the rationale for ISODE and building remote operations with respect to ISODE, respectively. These articles were published in *ConneXions, The Interoperability Report* Volume 1, No. 1, May 1987 and in Volume 2, No. 3, March 1988.

5.6 TUTORIALS

INTEROP, Inc. offers a tutorial on ISODE internals which has been taught by Marshall Rose.

5.7 ISODE RELEASES

ISODE version 6.0 was released in January 1990. The new release, ISODE 7.0, is due out in 1991. An announcement about the new release will be made in January. The following pages describe the current release and how ISODE can be obtained from the Internet.

ANNOUNCEMENT

The next release of the "ISO Development Environment" will be available on 24 January 1990. This release is called

ISODE 6.0

This software supports the development of certain kinds of OSI protocols and applications. Here are the details:

- The ISODE is not proprietary, but it is not in the public domain. This was necessary to include a "hold harmless" clause in the release. The upshot of all this is that anyone can get a copy of the release and do anything they want with it, but no one takes any responsibility whatsoever for any (mis)use.
- The ISODE runs on native Berkeley (4.2, 4.3) and AT&T (SVR2, SVR3) systems, in addition to various other UNIX-like operating systems. No kernel modifications are required.
- Current modules include:
 - OSI transport service (TPO on top of TCP and X.25; TP4 for SunLink OSI)
 - OSI session, presentation, and association control services
 - ASN.1 abstract syntax/transfer notation tools, including:
 - remote operations stub-generator (front-end for remote operations)
 - structure-generator (ASN.1 to C)
 - element-parser (basic encoding rules)
 - OSI reliable transfer and remote operations services
 - OSI file transfer, access and management
 - FTAM/FTP gateway
 - OSI directory services
 - OSI virtual terminal (basic class, TELNET profile)
- ISODE 6.0 consists of final "IS" level implementations with a few exceptions: ROSE and RTSE are current to the last circulated drafts (March, 1988); VT is a DIS implementation. The ISODE also contains implementations of the 1984 X.400 versions of ROS and RTS. ISODE is aligned with the U.S. Government OSI Profile (GOSIP).
- Modules planned for future releases include:
 - OSI message handling system
 - MHS/SMTP gateway

- Although the ISODE is not "supported" per se, it does have a problem reporting address, Bug-ISODE@NISC.NYSER.NET. Bug reports (and fixes) are welcome by the way.
- The discussion group ISODE@NIC.DDN.MIL is used as an open forum on ISODE. Contact ISODE-Request@NIC.DDN.MIL to be added to this list.
- The primary documentation for this release consists of a five volume User's Manual (approx. 1000 pages) and a set of UNIX manual pages. The sources to the User's Manual are in LaTeX format. In addition, there are a number of notes, papers, and presentations included in the documentation set, again in either LaTeX or SLiTeX format. If you do not have LaTeX, you should probably get a hardcopy from one of the distribution sites below.

For more information, contact:

PSI, Inc.
 PSI California Office
 Attn: Marshall T. Rose
 420 Whisman Court
 Mountain View, CA 94043-2112
 USA

+1-415-961-3380

DISTRIBUTION SITES

1. FTP

If you can FTP to the Internet, then use anonymous FTP to nisc.nyser.net [192.33.4.10] to retrieve the file isode-6.tar.Z in BINARY mode from the pub/isode/ directory. This file is the tar image after being run through the compress program and is approximately 4.5MB in size.

2. NIFTP

If you run NIFTP over the public X.25 or over JANET, and are registered in the NRS at Salford, you can use NIFTP with username "guest" and your own name as password, to access UK.AC.UCL.CS to retrieve the file <SRC>isode-6.tar. This is a 14MB tar image. The file <SRC>isode-6.tar.Z is the tar image after being run through the compress program (4.5MB).

3. NORTH AMERICA

For mailings in NORTH AMERICA, send a check for 375 US Dollars to:

University of Pennsylvania
Department of Computer and Information Science
Moore School
Attn: David J. Farber (ISODE Distribution)
200 South 33rd Street
Philadelphia, PA 19104-6314
USA

+1-215-898-8560

The tape will be written in tar format at 1600bpi, and returned with a documentation set. Do not send tapes or envelopes. Documentation only is the same price.

4. EUROPE (tape and documentation)

For mailings in EUROPE, send a cheque or bankers draft and a purchase order for 200 Pounds Sterling to:

Department of Computer Science
Attn: Natalie May/Dawn Bailey
University College London
Gower Street
London, WC1E 6BT
UK

For information only:

Telephone: +44-1-380-7214

Fax: +44-1-387-1397

Telex: 28722

Internet: natalie@cs.ucl.ac.uk, dawn@cs.ucl.ac.uk

Specify either (a) 1600bpi 1/2-inch tape, or (b) Sun 1/4-inch cartridge tape. The tape will be written in tar format and returned with a documentation set. Do not send tapes or envelopes. Documentation only is the same price.

6. EUROPE (tape only)

Tapes without hardcopy documentation can be obtained via the European UNIX User Group (EUUG). The ISODE 6.0 distribution is called EUUGD14.

EUUG Distributions
c/o Frank Kuiper
Centrum voor Wiskunde en Informatica
Kruislaan 413
1098 SJ Amsterdam
The Netherlands

For information only:

Telephone: +31-20-5924121 (or: +31-20-5929333)
Telex: 12571 mactr nl
Telefax: +31-20-5924199
Internet: euug-tapes@cw.nl

Specify one of:

- 1600bpi 1/2-inch tape: 130 Dutch Guilders
- 800bpi 1/2-inch tape: 130 Dutch Guilders
- Sun 1/4-inch cartridge tape (QIC-24 format): 190 Dutch Guilders
- 1/4-inch cartridge tape (QIC-11 format): 190 Dutch Guilders

If you require DHL this is possible and will be billed through. Note that if you are not a member of EUUG, then there is an additional handling fee of 300 Dutch Guilders (please enclose a copy of your membership or contribution payment form when ordering). Do not send money, cheques, tapes or envelopes, you will be invoiced.

7. AUSTRALIA and NEW ZEALAND

For mailings in AUSTRALIA and NEW ZEALAND, send a cheque for 250 dollars Australian to:

CSIRO DIT
Attn: Andrew Waugh (ISODE DISTRIBUTION)
55 Barry St
Carlton, 3053
Australia

For information only:

Telephone: +61-3-347-8644
Fax: +61-3-347-8987
Internet: ajw@ditmela.oz.au

Please specify the media you desire: (a) 1/2-inch tape at 1600bpi, 3200bpi, or 6250bpi; or (b) Sun 1/4-inch cartridge tape in either QIC-11 or QIC-24 format. The tape will be written in tar format and

returned with a documentation set. Do not send tapes or envelopes. Documentation only is the same price.

8. FTAM on the JANET or PSS

The sources are available by FTAM at the UCL over X.25 using JANET (00000511160013) or PSS (23421920030013) with TSEL "259" (ascii encoding). Use the "anon" user-identity and retrieve the file <SRC>isode-6.tar. This is a 14MB tar image. The file <SRC>isode-6.tar.Z is the tar image after being run through the compress program (4.5MB).

9. FTAM on the Internet

The sources are available by FTAM over the Internet at host osi.nyser.net [192.33.4.20] (TCP port 102 selects the OSI transport service) with TSEL 259 (numeric encoding). Use the "anon" user-identity, supply any password, and retrieve the file isode-6.tar.Z from the pub/isode/ directory. This file is the tar image after being run through the compress program and is approximately 4.5MB in size.

For distributions via FTAM, the file service is provided by the FTAM implementation in ISODE 5.0 or later (IS FTAM).

For distributions via either FTAM or FTP, there is an additional file available for retrieval, called isode-ps.tar.Z which is a compressed tar image (7MB) containing the entire documentation set in PostScript format.

REFERENCES

- [1] Moore, C., Book review of The Open Book: A Practical Perspective on OSI, ConneXions, The Interoperability Report, Vol. 3, No. 11, November 1989.
- [2] Rose, M.T., ISODE Internals, INTEROP Tutorial, October 8-9, 1990.
- [3] Rose, M.T., The Open Book: A Practical Perspective on OSI, Prentice-Hall, 1989.
- [4] Rose, M.T., ISODE User's Manuals, 5 volumes.

APPENDIX A

ISODE DIRECTORY DESCRIPTIONS

ISODE DIRECTORY	DIRECTORY DESCRIPTION
acsap/	ACSE and ALS Library
bin/	Initiator Programs
callback/	ISODE callback Demo Program
comments/	Copy of a Presentation Given as a Follow-on to ISODE3/
common/	
compat/	Compatibility Library
config/	Configuration Templates
cookbook/	Copy of a Tutorial Given at the TCP/IP ISO/OSI Tutorials in April, 1988
cn-isdn/	" ISO Transport Services on Top of the TCP", Reprint of an Article Appearing in the Computer Networks and ISDN Systems Journal, Vol. 12, No.3 1986
dirent/	System Directory Access Compatibility (GETDENTS)
doc/	Document Set
ds/	Copy of a Presentation on OSI Directory Services given Dec., 1988
dsap/	X.500 (Directory Service) Library
duug/	Copy of a Paper on Directory Navigation in the QUIPU X.500 System
etc/	Administrator Programs and Responder Programs
ftam/	File Transfer Access and Management (FTAM) Library
ftam-ftp/	FTAM to FTP Gateway
ftam2/	FTAM User/Filestore Programs
ftp-ftam/	FTP to FTAM Gateway
h/	include files
idist/	ISODE idist Utility

ISODE DIRECTORY	DIRECTORY DESCRIPTION
ifip88-quipu/	Copy of a Talk Given on QUIPU Directory Services
image/	ISODE image Utility, described in User's Manual, Volume 3
imisc/	Example Applications
interim/	Copy of Two Papers Written by S.E. Kille Which Defines the Addressing Formats Used by the ISODE
include/	Include Files
islands/	Copy of a Presentation on Communications -- An Enabling Technology: Bridging the Islands of Automation
iso-migrate/	Copy of a Presentation on Recent Developements in ISO Migration given August, 1986
isode/	ISODE Source Code
isode1/	Copy of a Presentation Given at the IBM T.J. Watson Research Center
isode2/	Copy of a Presentation Given at Silicon Graphics
isode3/	Copy of a Birds-of-a-Feather Given at the Third TCP/IP Interoperability Conference in September, 1988
isode4/	Copy of a Presentation Given to HP
isode5/	Slightly Updated Presentation on ISODE 5.0
isode6/	Standard Presentation on ISODE 6.0
isode-6.0/	ISODE Version 6.0
isode-snmp/	
issues/	Copy of a Tutorial on Issues in Transition and Coexistence for TCP/IP to OSI Given in November, 1988
lib/	Object Libraries and Lint Libraries
lint/	ISODE Libraries

ISODE DIRECTORY	DIRECTORY DESCRIPTION
listen/	ISODE listen Demo Program
local/	User Programs
lookup/	ISODE password lookup Demo Program
manual/	User's Manuals
map-migrate/	Copy of a Presentation on Migrating to MAP/TOP Given in September, 1986
max/	Directory Browser (X-based)
mhs/	Message Handling System (MHS)
monterey/	Copy of a Session on Protocol Advances 3: ISODE Given in March, 1987
mosy/	Managed Object Syntax-Compiler (yacc-based)
ncf/	Copy of the Presentation Focus on OSI for Networked Applications Given in November, 1987
net/	
nordunet/	Copy of Some Presentations Given at NORDUNET '89
osilookup/	ISODE/SunLink OSI Conversion
osi-posix/	Copy of a Presentation on OSI Protocols Within an Openly Available, POSIX Conformant, Berkeley UNIX Environment Given in December, 1987
osi-update/	Copy of a Presentation on Recent Developments with OSI Implementations
others/	"Unsupported" Additions and Example Applications
pepy/	ASN.1 Compilers
pingpong/	Asynchronous Connection Establishment

ISODE DIRECTORY	DIRECTORY DESCRIPTION
practical/	Copy of a Tutorial on Practical Perspectives on OSI Given in October, 1989
psap/	Presentation Abstractions
psap2/	Presentation Library
psap2-lpp/	Light Weight Presentation Protocol (RFC1085) Library
quipu/ (doc/quipu/)	Describes the Design of the QUIPU Directory
quipu/	Directory System Agent (DSA)/Directory User Agent (DUA) Programs
rdbm/	ISODE Remote DBM Demo Program
reality/	Copy of a Presentation on TCP/IP, MAP/TOP and the Reality of the ISO Protocols Given in January, 1987
rfcs/	Request For Comments
rosap/	Remote Operations Service Element (ROSE) Library
rosy/	RO-Compiler
rtf/	RT-File Transfer Utility
rtsap/	Relible Transfer Service Element (RTSE) Library
snmp/	Simple Network Management Protocol (SNMP) Library and Programs
ssap/	Session Library
support/	Deamons, Databases, etc.
template/	Program Header Templates
tmp/	Log Files
tp0bridge/	ISODE TCP/X.25 TP0 Bridge
trek/	Copy of a Presentation Made at the IRM Conference
tsap/	Transport Library

ISODE DIRECTORY	DIRECTORY DESCRIPTION
tsaptalk/	Transcript of the Presentation Given at the MAP/TOP User's Meeting in May, 1986
tsbridge/	ISODE TS-Bridge
util/	Manual Pages
vt/	Virtual Terminal (VT) Programs
whitepages/	Documents for the White Pages Pilot Project

APPENDIX B

ISODE PORTS

*** MAIL 1 ***

Date: Mon, 23 Jul 90 13:50:10 NZS
From: P.Kay@massey.ac.nz
Message-Id: <9007230150.AA12227@sis-a>
To: ISODE@nic.ddn.mil
Subject: VMS port of ISODE

Hi,

I have recently finished some work that I started a couple of years ago. - Porting ISODE to VMS.

Current state is:

ISODE-6.0 - all the libraries up to rosy (including pepy, posy and rosy)

imisc, isoc, ftam, dish clients
tsapd (single shot - but in a loop) firing off
imiscd, isod (tsap) and ftamd

Very little effort has gone into VMS'ing any of the programs - they still have their UNIX flavour of what file systems look like.

The port was done by writing a UNIX socket/syst library to glue ISODE to VMS's system routines. Very few changes were made to the ISODE code itself (mostly cosmetic to get it through the compiler).

The UNIX library only backs on to CMU/TEKS tcp/ip package (v6.4), and perhaps PSI (it certainly did work - but I've lost my PSI connection so can't test it anymore). I hope a backend to Multinet can be written soon.

The extra software to manage the port is freely available to anyone. The socket library and ISODE files (extra to the 6.0 distribution) amount to approx 600Kbytes of a compressed tar file. Building the code takes about 11 hours elapsed time of a (single user) 3300, and requires a fair amount of disk space.

We are connected to the Internet, but I doubt the rest of NZ would be happy if everyone grabbed the files from Massey, neither do I have an anonymous account I can use. If you would like the software, then either I can post it to you on an at floppy disk, or if someone would volunteer to put it up on a server thats more central I'd be grateful.

Peter Kay

*** MAIL 2 ***

Date: 27 Jul 90 06:35:35 GMT

F r o m :
sdd.hp.com!uakari.primate.wisc.edu!samsung!munnari.oz.au!mel
.dit.csiro.au!s
mart@ucsd.edu (Robert Smart)
Organization: CSIRO DIT (Melb.)
Subject: Re: VMS port of ISODE
Message-Id: <1990Jul27.063535.1732@mel.dit.csiro.au>
References: <9007230150.AA12227@sis-a>
Sender: isode-relay@nic.ddn.mil
To: isode@nic.ddn.mil

In article <9007230150.AA12227@sis-a> P.Kay@MASSEY.AC.NZ
writes:

>
> I have recently finished some work that I started a
couple of years
>ago. - Porting ISODE to VMS.
>
>Current state is:
>
>ISODE-6.0 - all the libraries up to rosy (including pepy,
posy and rosy)
> imisc, isoc, ftam, dish clients
> tsapd (single shot - but in a loop) firing off
imiscd, isod (tsap)
> and ftamd
>
>
>We are connected to the Internet, but I doubt the rest of NZ
would be happy if everyone grabbed the files from Massey,
neither do I have an anonymous account I can use. If you would
like the software, then either I can post it to you on an at
floppy disk, or if someone would volunteer to put it up
>on a server thats more central I'd be grateful.
>
> Peter Kay

This is available for anonymous ftp (and fetchfile) from
128.250.90.81 (ditmela.mel.dit.csiro.au). It is the file
isode/vmsisode60_tar_z.v1.
If you don't have a unix machine you might find the VMS
versions of compress and tar helpful.

This is for Australian sites and for people overseas who have
arranged with Peter Kay to make this available in their part
of the world.

Bob Smart

*** MAIL 3 ***

From: rosenqui@tonyl.skl.dnd.ca (Eric Rosenquist)
Message-Id: <9007271926.AA13806@tonyl.skl.dnd.ca>
Subject: ISODE on 386/ix
To: isode@nic.ddn.mil (ISODE List)
Date: Fri, 27 Jul 90 15:26:02 EDT
Reply-To: rosenqui@crc.skl.dnd.ca
X-Mailer: ELM [version 2.2 PL14]

In the near future we will be faced with the task of getting
ISODE
running under INTERACTIVE's 386/ix Unix. We don't expect too
much
trouble, but it would be nice to know if anyone else has tried
this
already, and if so, how things went. Comments anyone??

Thanks in advance,
Eric

--

Eric Rosenquist
rosenqui@crc.skl.dnd.ca
Software Kinetics Limited
65 Iber Road, Stittsville, Ontario
Canada - K2S 1E7 Phone (613) 831-0888

isode-RE NIC 7/28/90
Eric Rosenquist ISODE List
386/ix

7/27/90 ISODE on

*** MAIL 4 ***

Date: 10 Aug 90 13:36:17 GMT
From: mcsun!unido!uni-koeln!a0098@uunet.uu.net (Peter
heiligers)
Organization: Regional Computing Center, University of
Cologne, F. R. Germany
Subject: ISODE under AIX370
Message-Id: <1990Aug10.133617.19014@rrz.uni-koeln.de>
Sender: isode-relay@nic.ddn.mil
To: isode@nic.ddn.mil

We would like to run ISODE under AIX370. It would be nice to
know if someone
has already tried this, and how things went.

Thanks in advance

Peter Heiligers, Regional Computing Center, University of
Cologne

Robert-Kochstr. 10, D-5000 Koeln 41, West Germany
Internet: a0098@rrz.uni-koeln.DE

Bitnet: a0098@dk0rrzk1

Phone: +49 (221) 478 5578

isode-RE NIC 8/10/90
Peter heiligers isode@nic.ddn.mil 8/10/90 ISODE under
AIX370

*** MAIL 5 ***

Date: Fri, 10 Aug 90 22:15:05 EDT
From: Valdis Kletnieks <VALDIS@VTVM1.CC.VT.EDU>
Organization: Virginia Polytechnic Institute
Subject: Re: ISODE under AIX370
To: Peter heiligers
<mcsun!unido!uni-koeln!a0098@uunet.uu.net>,
isode@nic.ddn.mil
Message-Id: <900810.221505.EDT.VALDIS@vtvm1.cc.vt.edu>
In-Reply-To: <1990Aug10.133617.19014@rrz.uni-koeln.de>

On 10 Aug 90 13:36:17 GMT you said:
>We would like to run ISODE under AIX370. It would be nice to
know if someone
>has
> already tried this, and how things went.
>
(Second try - I managed to trash out my MUA the first time.)
I've ported ISODE 6.0 to the 3090 side of an AIX/370 cluster.
I'm told my config file works fine on an AIX PS/2 1.1 system
as well, so
it shouldn't be THAT hard to fix it to use hidden directories
to support
both flavors at once.

Feel free to contact me for details - although as I am leaving
for SHARE
in New Orleans, I won't be able to reply before August 20 or
so.

Engineer

Valdis Kletnieks
Computer Systems

Virginia Tech
valdis@vtvm1.cc.vt.edu

isode-RE NIC 8/11/90
Valdis Kletnieks Peter heiligers 8/10/90*ISODE under
AIX370

*** MAIL 6 ***

To: mcsun!unido!uni-koeln!a0098@uunet.uu.net (Peter heiligers)
Cc: isode@nic.ddn.mil
Subject: Re: ISODE under AIX370
In-Reply-To: Your message of "10 Aug 90 13:36:17 GMT."
<1990Aug10.133617.19014@rrz.uni-koeln.de>
Date: Fri, 10 Aug 90 23:15:34 -0400
From: Tim.Howes@terminator.cc.umich.edu

> We would like to run ISODE under AIX370. It would be nice to
know if
someone h
- as already tried this, and how things went.

We brought up ISODE and Quipu under aix/370. It went pretty
smoothly. Just
use the makefiles for a system V-ish system. Do not make the
mistake of
trying to use -lbsd to get bsd compatibility. That way
madness lies ...

The only trouble I had was in the BYTE definitions in pepy, I
think it
was. That wasn't hard to solve, though.

ISODE seems to work fine. Quipu also works fine, with the
exception that
it can't make outbound connections for some reason. Since the
DSA we have
running is a second line dsa it only gets referrals from other
dsa's. So,
this hasn't been a big enough problem for me to solve yet...

-- Tim

isode-RE NIC 8/11/90

Tim.Howes@terminato Peter heiligers
AIX370

8/10/90*ISODE under

*** MAIL 7 ***

Date: 2 Oct 90 15:57
F r o m : B e r n a r d D e t r e m b l e u r
<bernard.detrembleur@ts.info.fundp.rtt.be>
To: <pp-people@cs.ucl.uk.ac.gold-400.gb>
Cc: isode@NIC.DDN.MIL
Message-Id: <752:bernard.detrembleur@ts.info.fundp.rtt.be>
Subject: ISODE 6.0 and PP 5.0 on HP 9000/375

Has somebody already ported ISODE 6.0 and PP 5.0 on HP-UX 7.0
with an X25 interface HP 36941A DIO and the X25 Programmatic
interface.

If yes, is there a patch somewhere to install ISODE 6.0 and
PP 5.0 on top of X25 for the interface (which is not CAMTEC,
UBC X25, SUNLINK X25...)

Thanks in advance for your help.

All the bests,

Bernard Detrembleur
Belgian WEP Manager

+++++F.U.N.D.P.+++++
+++++
Bernard Detrembleur
Institut d'Informatique
21, rue Grandgagnage
B-5000 NAMUR
bdt@info.fundp.ac.be (UUCP)
detrembleur@ts.info.fundp.rtt.be (X400)

*** MAIL 8 ***

To : B e r n a r d D e t r e m b l e u r
<bernard.detrembleur@ts.info.fundp.rtt.be>
Cc: pp-people@cs.ucl.uk.ac.gold-400.gb, isode@NIC.DDN.MIL
Subject: Re: ISODE 6.0 and PP 5.0 on HP 9000/375
In-Reply-To: Bernard Detrembleur's message of 2 Oct 90 15:57
.
<752:bernard.detrembleur@ts.info.fundp.rtt.be>
Date: Wed, 03 Oct 90 12:11:05 +0100
Message-Id: <8314.654952265@cli53an>
From: Sylvain Langlois <sylvain@cli53an.edf.fr>

> Has somebody already ported ISODE 6.0 and PP 5.0 on HP-UX
7.0
> with an X25 interface HP 36941A DIO and the X25 Programmatic
> interface.

It's on my work plan. I hope being able to do it by
mid-November. If
you have no other answer, I will be happy to send you my
resulting
work. However, if you get any positive acknowledgement to your
query, I
would be interested in getting them to . Thanks...

Sylvain

Sylvain Langlois "Dogmatic attachement to the
supposed merits
(sylvain@cli53an.edf.fr) of a particular structure hinders
the search of an appropriate structure" (Robert
Fripp)

*** MAIL 9 ***

Date: Thu, 11 Oct 90 20:06:29 PDT
Message-Id: <9010120306.AA26633@rufus.nosc.mil>
From: Kevin.F.Spalding%code413.nosc.mil@nosc.mil
(spalding@code413.nosc.mil)
To: isode@nic.ddn.mil
Subj: ISODE 6.0 on IBM RS6000 Under AIX

I am looking to know if anyone has ported isode 6.0
to AIX running on the IBM RS6000. How much if any real
effort is required to get it up and running.

Thanks in advance.

Kevin F. Spalding
email: spalding@code413.nosc.mil

isode-RE NIC 10/12/90
spalding@code413.no isode@nic.ddn.mil 10/11/90 No subject

*** MAIL 10 ***

Date: Sat, 27 Oct 90 18:30:01 MET DST

Cc: isode@nic.ddn.mil

In-Reply-To: <9007271926.AA13806@tonyl.skl.dnd&ca>; from "Eric Rosenquist" at Jul 28, 90 7:23 am

Organisation: CSD IMMD IV, University of Erlangen (FAU), W - Germany

X-Mailer: ELM [version 2.2 PL13]

> In the near future we will be faced with the task of getting ISODE

> running under INTERACTIVE's 386/ix Unix. We don't expect too much

> trouble, but it would be nice to know if anyone else has tried this

> already, and if so, how things went. Comments anyone??

The task of porting isode to 386/ix and SCO is financed by the DFN and will be done by netCS. Contact baumgarten@zpl.dfn.dbp.de for more info.

Toerless Eckert

isode-RE NIC 10/27/90

Toerless Eckert

rosenqui@crc.skl.dn 10/27/90*ISODE on 386/ix

APPENDIX C

ISODE TEST PROGRAM AND OUTPUT

```

: run this script through /bin/sh

# isode-6.0/isode-test
# A program to test out the isode services etc.
#
# Pretty simple minded - but gives some indications.
#
# Julian Onions <jpo@cs.nott.ac.uk> 15/1/86

# usage : isode-test [ hostname ]
if [ ! -f support/xisoc ]; then
    PATH=$PATH:/usr/local/bin export PATH
    P1= P2=
else
    P1=support/x P2=imisc/x
fi

error=0 fast=0
if [ "x$1" = "x-lpp" ]; then
    fast=1 S=-lpp
    shift
else
    S=
fi
IMISC=${P2}imisc${S} ISOC=${P1}isoc

if [ $# -gt 0 ]; then
    host="$1"
    echo "Hostname set to $host"
elif host=""`hostname`; then
    echo "Hostname set to $host"
elif host=""`uname`; then
    echo "Hostname set to $host"
else
    echo -n "I give up, what is your host name? "
    read host
fi

for i in utctime gentime time users chargen qotd finger pdggen
do
    echo "$i:"
    $IMISC "$host" $i || error=`expr $error + 1`
done

for i in ping sink echo
do
    echo "$i:"
    $IMISC -c 100 -l 1020 "$host" $i || error=`expr $error + 1`
done

if [ $fast = 1 ]; then

```



```

        echo "Test done, Errors: $error"
        exit $error
    fi

    for i in echo isode/echo
    do
        echo "rosap $i:"
        $ISOC "$host" rosap $i < /etc/passwd || error=`expr $error + 1`
    done

    for i in echo ros_echo "isode/rtse echo" isode/ros_echo
    do
        echo "rtsap $i:"
        $ISOC "$host" rtsap "$i" < /etc/passwd || error=`expr $error + 1`
    done

    for i in echo isode/echo
    do
        echo "psap $i:"
        $ISOC "$host" psap $i < /etc/passwd || error=`expr $error + 1`
    done

    for i in ssap tsap
    do
        echo "$i echo:"
        $ISOC "$host" $i echo < /etc/passwd || error=`expr $error + 1`
    done

    echo "Test done, Errors: $error"
    exit $error

```

* Output generated by Isode-test program *

Hostname set to ibis

utctime:

910102211527Z

gentime:

19910102211528.810000Z

time:

Wed Jan 2 16:15:30 1991

users:

zhang	console	Jan	2	10:50	
zhang	ttyp0	Jan	2	11:24	
shen	ttyp1	Jan	2	16:15	
root	ttyp2	Jan	2	14:32	(falcon)
zhang	ttyp3	Jan	2	11:24	
zhang	ttyp4	Jan	2	11:24	
zhang	ttyp5	Jan	2	11:24	
nee	ttyp6	Jan	2	15:39	(condor)
shen	ttyp7	Jan	2	15:54	(heron)

chargen:

```
! "#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\
]^_`abcdefg
! "#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\
]^_`abcdefgh
! "#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\
]^_`abcdefghi
! "#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\
]^_`abcdefghij
! "#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\
]^_`abcdefghijkl
! "#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\
]^_`abcdefghijklm
! "#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\
]^_`abcdefghijklm
```

gotd:

Oh don't the days seem lank and long
 When all goes right and none goes wrong,
 And isn't your life extremely flat
 With nothing whatever to grumble at!

finger:

Login	Name	TTY	Idle	When	Where
zhang	Qian Zhang	co	14	Wed 10:50	
root	Operator	p2	5	Wed 14:32	falcon
nee	Hai-Lin Nee	p6	23	Wed 15:39	condor
shen	David T. Shen	p7		Wed 15:54	heron

```

pwdgen:
IBPIRIBO
HUGOIDOV
AWETNATE
ETKEMACE
EPDIDOCO
ATADOWAR
ping:
100 operations in 3.80 seconds (26.32 ops/s); 2 bytes/op for
0.05 Kbytes/s
sink:
100 operations in 3.17 seconds (31.55 ops/s); 1024 bytes/op
for 31.55 Kbytes/s
echo:
100 operations in 3.64 seconds (27.47 ops/s); 1024 bytes/op
for 27.47 Kbytes/s
rosap echo:
ibis... connected
xisoc: 462 bytes echoed in 0.01 seconds (90.23 Kbytes/s)
xisoc: 462 bytes echoed in 0.01 seconds (47.49 Kbytes/s)
xisoc: 462 bytes echoed in 0.00 seconds (100.26 Kbytes/s)
xisoc: 462 bytes echoed in 0.01 seconds (47.49 Kbytes/s)
xisoc: 462 bytes echoed in 0.01 seconds (90.23 Kbytes/s)
xisoc: 462 bytes echoed in 0.00 seconds (100.26 Kbytes/s)
xisoc: 462 bytes echoed in 0.01 seconds (47.49 Kbytes/s)
xisoc: 462 bytes echoed in 0.00 seconds (100.26 Kbytes/s)
xisoc: 462 bytes echoed in 0.01 seconds (47.49 Kbytes/s)
xisoc: 462 bytes echoed in 0.00 seconds (100.26 Kbytes/s)
rosap isode/echo:
ibis... connected
xisoc: 462 bytes echoed in 0.02 seconds (45.12 Kbytes/s)
xisoc: 462 bytes echoed in 0.01 seconds (47.49 Kbytes/s)
xisoc: 462 bytes echoed in 0.01 seconds (47.49 Kbytes/s)
xisoc: 462 bytes echoed in 0.01 seconds (47.49 Kbytes/s)
xisoc: 462 bytes echoed in 0.00 seconds (100.26 Kbytes/s)
xisoc: 462 bytes echoed in 0.03 seconds (30.08 Kbytes/s)
xisoc: 462 bytes echoed in 0.01 seconds (47.49 Kbytes/s)
xisoc: 462 bytes echoed in 0.01 seconds (47.49 Kbytes/s)
xisoc: 462 bytes echoed in 0.01 seconds (47.49 Kbytes/s)
xisoc: 462 bytes echoed in 0.01 seconds (47.49 Kbytes/s)
rtsap echo:
ibis... connected
xisoc: 462 bytes echoed in 0.59 seconds (1.51 Kbytes/s)
xisoc: 462 bytes echoed in 0.77 seconds (1.16 Kbytes/s)
xisoc: 462 bytes echoed in 0.78 seconds (1.14 Kbytes/s)
xisoc: 462 bytes echoed in 0.78 seconds (1.14 Kbytes/s)
xisoc: 462 bytes echoed in 0.78 seconds (1.14 Kbytes/s)
xisoc: 462 bytes echoed in 0.78 seconds (1.14 Kbytes/s)
xisoc: 462 bytes echoed in 0.78 seconds (1.14 Kbytes/s)
xisoc: 462 bytes echoed in 0.78 seconds (1.14 Kbytes/s)
xisoc: 462 bytes echoed in 0.78 seconds (1.14 Kbytes/s)
xisoc: 462 bytes echoed in 0.78 seconds (1.14 Kbytes/s)

```



```

xisoc: 462 bytes echoed in 0.78 seconds (1.14 Kbytes/s)
rtsap ros_echo:
ibis... connected
xisoc: 462 bytes echoed in 0.58 seconds (1.56 Kbytes/s)
xisoc: 462 bytes echoed in 0.77 seconds (1.16 Kbytes/s)
xisoc: 462 bytes echoed in 0.78 seconds (1.14 Kbytes/s)
xisoc: 462 bytes echoed in 0.78 seconds (1.14 Kbytes/s)
xisoc: 462 bytes echoed in 0.78 seconds (1.16 Kbytes/s)
xisoc: 462 bytes echoed in 0.78 seconds (1.14 Kbytes/s)
xisoc: 462 bytes echoed in 0.78 seconds (1.14 Kbytes/s)
xisoc: 462 bytes echoed in 0.78 seconds (1.14 Kbytes/s)
xisoc: 462 bytes echoed in 0.78 seconds (1.14 Kbytes/s)
rtsap isode/rtse_echo:
ibis... connected
xisoc: 462 bytes echoed in 0.48 seconds (1.85 Kbytes/s)
xisoc: 462 bytes echoed in 0.77 seconds (1.16 Kbytes/s)
xisoc: 462 bytes echoed in 0.78 seconds (1.14 Kbytes/s)
xisoc: 462 bytes echoed in 0.78 seconds (1.14 Kbytes/s)
xisoc: 462 bytes echoed in 0.78 seconds (1.14 Kbytes/s)
xisoc: 462 bytes echoed in 0.78 seconds (1.14 Kbytes/s)
xisoc: 462 bytes echoed in 0.77 seconds (1.16 Kbytes/s)
xisoc: 462 bytes echoed in 0.78 seconds (1.14 Kbytes/s)
xisoc: 462 bytes echoed in 0.78 seconds (1.14 Kbytes/s)
xisoc: 462 bytes echoed in 0.78 seconds (1.14 Kbytes/s)
rtsap isode/ros_echo:
ibis... connected
xisoc: 462 bytes echoed in 0.49 seconds (1.84 Kbytes/s)
xisoc: 462 bytes echoed in 0.76 seconds (1.17 Kbytes/s)
xisoc: 462 bytes echoed in 0.78 seconds (1.14 Kbytes/s)
xisoc: 462 bytes echoed in 0.78 seconds (1.14 Kbytes/s)
xisoc: 462 bytes echoed in 0.78 seconds (1.14 Kbytes/s)
xisoc: 462 bytes echoed in 0.78 seconds (1.14 Kbytes/s)
xisoc: 462 bytes echoed in 0.78 seconds (1.14 Kbytes/s)
xisoc: 462 bytes echoed in 0.78 seconds (1.14 Kbytes/s)
xisoc: 462 bytes echoed in 0.77 seconds (1.16 Kbytes/s)
xisoc: 462 bytes echoed in 0.78 seconds (1.14 Kbytes/s)
psap_echo:
ibis... ..connected
xisoc: 462 bytes echoed in 0.17 seconds (5.31 Kbytes/s)
xisoc: 462 bytes echoed in 0.04 seconds (18.42 Kbytes/s)
xisoc: 462 bytes echoed in 0.03 seconds (23.14 Kbytes/s)
xisoc: 462 bytes echoed in 0.03 seconds (23.14 Kbytes/s)
xisoc: 462 bytes echoed in 0.03 seconds (23.14 Kbytes/s)
xisoc: 462 bytes echoed in 0.03 seconds (23.14 Kbytes/s)
xisoc: 462 bytes echoed in 0.03 seconds (23.14 Kbytes/s)
xisoc: 462 bytes echoed in 0.03 seconds (23.14 Kbytes/s)
xisoc: 462 bytes echoed in 0.03 seconds (23.14 Kbytes/s)
xisoc: 462 bytes echoed in 0.03 seconds (23.14 Kbytes/s)
psap isode/echo:
ibis... ..connected

```


ssap echo:

ibis... ..connected

xisoc:	462 bytes	echoed in	0.10 seconds	(9.02 Kbytes/s)
xisoc:	462 bytes	echoed in	0.02 seconds	(31.12 Kbytes/s)
xisoc:	462 bytes	echoed in	0.02 seconds	(31.12 Kbytes/s)
xisoc:	462 bytes	echoed in	0.01 seconds	(47.49 Kbytes/s)
xisoc:	462 bytes	echoed in	0.01 seconds	(47.49 Kbytes/s)
xisoc:	462 bytes	echoed in	0.02 seconds	(31.12 Kbytes/s)
xisoc:	462 bytes	echoed in	0.01 seconds	(47.49 Kbytes/s)
xisoc:	462 bytes	echoed in	0.02 seconds	(31.12 Kbytes/s)
xisoc:	462 bytes	echoed in	0.01 seconds	(47.49 Kbytes/s)
xisoc:	462 bytes	echoed in	0.02 seconds	(31.12 Kbytes/s)

tsap echo:

ibis... ..connected

xisoc:	462 bytes	echoed in	0.00 seconds	(100.26 Kbytes/s)
xisoc:	462 bytes	echoed in	0.00 seconds	(100.26 Kbytes/s)
xisoc:	462 bytes	echoed in	0.00 seconds	(902.34 Kbytes/s)
xisoc:	462 bytes	echoed in	0.00 seconds	(100.26 Kbytes/s)
xisoc:	462 bytes	echoed in	0.00 seconds	(100.26 Kbytes/s)
xisoc:	462 bytes	echoed in	0.00 seconds	(902.34 Kbytes/s)
xisoc:	462 bytes	echoed in	0.01 seconds	(47.49 Kbytes/s)
xisoc:	462 bytes	echoed in	0.00 seconds	(100.26 Kbytes/s)
xisoc:	462 bytes	echoed in	0.00 seconds	(100.26 Kbytes/s)
xisoc:	462 bytes	echoed in	0.00 seconds	(100.26 Kbytes/s)

Test done, Errors: 0

APPENDIX D

ISODE BUG INFORMATION

LIST OF ISODE FILES WHICH NEEDED LIBRARY "INTERNET.H"

acsap:	dse.py
rosap:	ro2ssexec.c ro2ssinitiat.c ro2ssrespond.c
rtsap:	rt2ssexec.c rt2ssinitiat.c rt2ssrespond.c
psap2:	psapinitiate.c
ssap:	ssapinitiate.c tsdu2spkt.c
tsap:	fd2tpkt.c text2tpkt.c tsapinitiate.c
compat:	general.c tailor.c servbyport.c

*** MAIL 1 ***

Date: 10 Aug 90 13:32:09 GMT
From: mcsun!ub4b!lln-cs!sunbim!old@uunet.uu.net (Olivier Dubois)
Organization: BIM Everberg Belgium
Subject: FTAM presentation user data size
Message-Id: <1532@sunbim.UUCP>
Sender: isode-relay@nic.ddn.mil
To: isode@nic.ddn.mil

Some weeks ago I sent an article on incompatibility between the isode FTAM implementation and M1/T1/A/111 profile compliant implementation. The isode responder as well as the initiator can generate too big presentation data value for other implementation to decode. I added a test in the code to limit the size of PDV. The files ftam-put.c and ftamd-trans.c in the ftam2 directory are to be modified I used the isode-5.9 code.

in ftam-put.c, in the put() function, after the lines:

```
.
.
#ifdef BRIDGE
    bsize = BUFSIZ << 2;
    size = magic >= bsize ? magic : bsize;
    if (size > bsize)
        size -= size % bsize;
#else
#ifdef MAXBSIZE
    bsize = BUFSIZ;
#else
    bsize = st.st_blksize > 0 ? st.st_blksize : BUFSIZ;
#endif
    size = (1024 <= magic && magic < bsize) ? magic : bsize;
#endif
```

```
add :
/*      90/08/10
      added by Olivier Dubois, BIM, Everberg, Belgium to
conform
      to ftam profile T1 or A/111 and derivatives where the
FTAM user data
      is limited to 7K. Conformant implementation can crash
receiving to big
      Presentation data value. This prevent from putting files
bigger than 7K
      to their filestore. This is a simple hack to allow
interoperability
with
```


conformant implementations. The limitation could be done fragmenting presentation element in the FDataRequest library function. It is the file

sender which decide the P-Data bound in the transfer, it is safe only to

modify the sender functions.

the same modification has to be made to the responder function uxfget in

ftamd-trans.c..

the bridge case has to be checked against this problem.

*/

```
#define FTAM_MAX_USER_DATA 4096 /* I choose 4K as 7k is the maximum decoded
```

```
few user data size, you have to keep a
```

```
from headroom bytes for the receiver ( 100
```

```
power experience ). So why not stay with a
```

```
2 for faster buffering)
```

```
*/
```

```
if ( size > FTAM_MAX_USER_DATA )
```

```
switch (vf - vfs) {
```

```
case VFS_UTF:
```

```
size = FTAM_MAX_USER_DATA + MAGIC_OCTET1;
```

```
break;
```

```
case VFS_UBF:
```

```
default:
```

```
size = FTAM_MAX_USER_DATA + MAGIC_SINGLE;
```

```
break;
```

```
}
```

```
/* end addition */
```

and apply the some modification in the file ftamd-trans.c in the function

ufxget().

I hope someone can find this patch usefull and I always interessted to hear

your feelings about that damned file transfer protocol.

Olivier Dubois,
System engineer,
BIM sa/nv, Belgium.

isode-RE NIC 8/11/90

*** MAIL 2 ***

To: old@sunbim.be
Cc: isode@nic.ddn.mil
Reply-To: Bug-ISODE@nisc.nyser.net
Subject: Re: FTAM presentation user data size
In-Reply-To: Your message of 10 Aug 90 14:01:59 +0000.
<1534@sunbim.UUCP>
Date: Sat, 11 Aug 90 18:15:02 -0700
Message-Id: <5675.650423702@cheetah.nyser.net>
From: Marshall Rose <mrose@cheetah.nyser.net>

You really should use the bug-isode address.

This is the first time I have seen your message.

A fix will be made.

/mtr

isode-RE NIC 8/12/90

Marshall Rose

presentation user data

old@sunbim.be

8/11/90*FTAM

*** MAIL 3 ***

Date: Mon, 13 Aug 90 21:41:52 GMT
From: Andy Sykes <asmec!andy@relay.EU.net>
Message-Id: <26c71ea0.asmec@asmec.uucp>
X-Mailer: Mush 6.5.6 (PC R6.2 22-Sep-89)
To: isode@nic.ddn.mil
Subject: Pepy and ACSE
Cc: bug-isode@nisc.nyser.net

First, some background;

ASMEC is a design consultancy specialising in COMMS/OSI related topics within the Industrial and Scientific fields of the European Computer Industry. We are currently engaged in the implementation of a full function OSI stack on Stratus hardware for ICL Network Systems Ltd (part of ICL/Fujitsu) to be used as the backbone for an X.400 message handling system.

Having previously ported ISODE onto other ICL UNIX based kit, we are grafting the Presentation and Application layers onto the Stratus OSI session layer server.

Now the questions; The following is little lengthy, so I apologise to anyone who is good natured enough to wade through it. In brief they concern some questions we have with regard to ACSE and PEPY. Being new to this conferance I'm not sure whether they should be aired here or on bug-ISODE, so I have copied this to both for good measure. However we would be very interested and gratefull for any help and/or comments anyone would care to make.

Association Control User Data
=====

Are there any further details or documentation on the use of the

User Data fields in the AcAssocRequest and AcAssocResponse calls?

>From the code this seems to be an array of PEs, but there is no details as to what form these should take. Moreover, the AcAssocRequest code specifically rejects PEs with the pe_context field set to PE_DFLT_CTX as in the following fragment from AcAsynAssocRequest.c:-

```
if (data) {
    register int i;
    register PE *pep;

    for (pep = data, i = ndata; i > 0; pep++, i--)
        if ((*pep) -> pe_context == PE_DFLT_CTX)
            return acsaplose (aci, ACS_PARAMETER, NULLCP,
                "default context not allowed for user-data at
slot %d",
                pep - data);
}
```

PE_DFLT_CTX, however, is the only value defined for this field in the psap.h header. We are confused. Has anyone actually used User Data during association control? The method of calculation of the slot value in the error message would suggest that this message is rarely if ever seen.

We have our own ideas on implementing ACSE User Data but don't wish to lock ourselves out of future releases of the code.

PEPY Problem No. 1
=====

How does one use the pepy MODULE include feature mentioned all too briefly in the documentation?

PEPY Problem No. 2
=====

It would appear that PEPY has a problem with ASN.1 definitions

which define a CHOICE construct as an OPTIONAL member of another construct. The example below illustrates this with respect to the ContType data element, being part of the X.411 MTS:-

```
ContType ::= CHOICE {
    built-in BuiltInContType,
    external ExternalContType }

RepDlvryArg ::= SET {
    subj-subm-id SubjSubmId,
    cont-id ContId OPTIONAL,
    cont-type ContType OPTIONAL,
    origin-eits OriginEITS OPTIONAL,
    per-recv-flds SEQUENCE OF PerRecvRepDlvryFlds
(1..ub-recv),
    returned-cont [0] Cont OPTIONAL }
```

The PEPY produced code for the RepDlvryArg construct is as follows.
Please ignore the asn1_ prefix and _f suffix to the function calls.
These are present to aid the Stratus build system (Oh for make under VOS!!).

```
int encode_P3_RepDlvryArg_f (PE *pe , int explicit , integer
len , char
*buffer
, struct type_P3_RepDlvryArg *parm )
{
    struct choice_P3_0 *choice_P3_0;
    struct element_P3_0 *element_P3_0;

    PE p10_z = NULLPE;
    register PE *p10 = &p10_z;

    if ( ( (*pe) = asn1_pe_alloc_f(PE_CLASS_UNIV,
PE_FORM_CONS,
PE_CONS_SET))
== NULLPE) {
        asn1_advise_f(NULLCP, "RepDlvryArg: %s",
PEPY_ERR_NOMEM);
        return NOTOK;
    }
    {
        choice_P3_0 = parm -> member_P3_0;
```

```

        element_P3_0 = parm -> per__recip__flds;
    }

    ***** Intervening code removed for brevity *****

    (*p10) = NULLPE;

    *****
    The following section encodes the ContType data
    element and
    should be dependant upon the presence of the
    choice_P3_0
    pointer.
    *****

    {
        int      p11;

        switch (p11 = (choice_P3_0 -> offset)) {
            case 1:      /* built-in */
                {
                    if (encode_P3_BuiltInContType_f(p10, 0,
NULLINT, NULLCP,
choice_P3_0 -> un.built__in ) == NOTOK)
                        return NOTOK;

#ifdef DEBUG
                                (void)      asn1_testdebug_f(      (*p10),
"built-in");
#endif

                }
                break;
            case 2:      /* external */
                {
                    if (encode_P3_ExternalContType_f(p10, 0,
NULLINT,
NULLCP,
choice_P3_0 -> un.external ) == NOTOK)
                        return NOTOK;

#ifdef DEBUG
                                (void)      asn1_testdebug_f(      (*p10),
"external");
#endif

                }
        }
    }

```

```

        break;

        default:
            asn1_advise_f(NULLCP,      "member      %s%d",
PEPY_ERR_INVALID_CHOICE,

                                p11);
            return NOTOK;
    }

#ifdef DEBUG
        (void) asn1_testdebug_f( (*p10), "member");
#endif

    }
    if ( (*p10) != NULLPE)
        if (asn1_set_add_f( (*pe),  (*p10)) == NOTOK) {
            asn1_advise_f(NULLCP,      "RepDlvryArg      %s%s",
PEPY_ERR_BAD_SET,

                                asn1_pe_error_f( (*pe) -> pe_errno));
            return NOTOK;
        }
    (*p10) = NULLPE;

    ***** Intervening code removed for brevity *****

    return OK;
}

```

It will be seen that as things stand the section of code which encodes the ContType element will be processed regardless. What should be present is an if statement as follows:-

```

    if (choice_P3_0)
    {
        int      p11;

        switch (p11 = (choice_P3_0 -> offset)) {

```

which will provide the desired result.

PEPY Problem No. 3

=====

Another problem from the following ASN.1 definition for the MTS:-


```

PerRecipientReportDeliveryFields ::= SET{
    actual-recipient-name [0] ActualRecipientName,
    report-type [1] ReportType,
    converted-encoded-information-types
                                ConvertedEncodedInformationTypes
OPTIONAL,
    originally-intended-recipient-name [2]
                                OriginallyIntendedRecipientName
OPTIONAL,
    supplementary-information [3]
                                SupplementaryInformation OPTIONAL,
    extensions [4] EXTENSIONS CHOSEN FROM {
        redirection-history,
        physical-fowarding-address,
        recipient-certificate,
        proof-of-delivery} DEFAULT {}
}

ReportType ::= CHOICE{
    delivery [0] DeliveryReport,
    non-delivery [1] NonDeliveryReport
}

DeliveryReport ::= SET{
    message-delivery-time [0] MessageDeliveryTime,
    type-of-MTS-user [1] TypeOfMTSUser DEFAULT public
}

NonDeliveryReport ::= SET{
    non-delivery-reason-code [0] NonDeliveryReasonCode,
    non-delivery-diagnostic-code [1]
NonDeliveryDiagnosticCode OPTIONAL
}

```

One the things PEPY does when dealing with CHOICE constructs which are members a higher level constructs is to 'pull up' the type of the CHOICE to the type which references them, in this case a SET. In this case, therefore, the data element structure should be as follows:-

```

SET (PerRecipientReportDeliveryFields)
+--> [0] actual-recipient-name ActualRecipientName
      |
      +--> [1] report-type SET (ReportType)
            |
            +--> either [0] SET DeliveryReport
                    or [1] SET NonDeliveryReport
            |
            Other members.

```


Actually, being a SET, the members can appear in the encoded data in any order.

The decode function produced by PEPY, having found the report-type member, correctly expects to see a SET constructor (ReportType) followed by another SET constructor representing the member of the CHOICE construct. If this is not the case there is a clash of the context specific tags.

Unfortunately the corresponding PEPY encode function does not produce the first SET constructor for the report-type element, possibly because both of the potential members of report-type are themselves of the SET type, though we have done no investigation to confirm this.

So, the questions are;

1. Has anyone encountered the same or similar problems and, if so, are there any solutions, particularly with the PEPY problems.
2. Are there any other related problems with PEPY we should be aware of?
3. Is there any later release of PEPY (> Rel 6.0) available?

Thank you for your perseverance if you have got this far. We look forward to any comments with interest.

isode-RE NIC 8/13/90

Andy Sykes

isode@nic.ddn.mil

8/13/90 Pepy and ACSE

*** MAIL 4 ***

Date: Tue, 14 Aug 90 16:35:17 PDT
From: guse@CCC.NERSC.GOV
Message-Id: <900814163517.20217e36@CCC.NERSC.GOV>
Subject: ASN.1 REAL encodings
To: isode@nic.ddn.mil
X-ST-Vmsmail-To: ST%"isode@nic.ddn.mil"

I am confused about the REAL encoding used in ISODE. It appears that the BER specifies that the mantissa of the real (in binary) is encoded as an unsigned integer. (10.5.5 -- ISO8825) However, pg. 299 of The Open Book, and apparently the ISODE implementation encodes the mantissa as a twos-complement integer. I am interested in clarification/explanation of this inconsistency.

isode-RE NIC 8/14/90
guse@CCC.NERSC.GOV isode@nic.ddn.mil 8/14/90 ASN.1 REAL
encodings

*** MAIL 5 ***

Date: Mon, 20 Aug 90 17:45:58 GMT
From: Andy Sykes <asmec!andy@relay.EU.net>
Message-Id: <26d021d6.asmec@asmec.uucp>
X-Mailer: Mush 6.5.6 (PC R6.2 22-Sep-89)
To: Bug-ISODE@nisc.nyser.net
Subject: Bug in info2ppdu
Cc: isode@nic.ddn.mil

We believe that we have uncovered a bug in the Presentation Layer function info2ppdu when dealing with simply encoded User data.

The User data is passed into this function as an array of PEs.

Depending upon the type in the pdu->offset field after checking, the User data is converted into a qbuf list using the info2qb function. For simply encoded data this function is called once for each PE such that the data therein is concatenated into a single qbuf. Fully encoded data is built into a list of qbufs.

The bug relates to the fact the the qbuf is not set up correctly for the simple encoded case. The qb_len field is left with a value of zero, and the qb_data pointer points to the end of the data held in the qb_base character array due to the manner in which info2qb operates. The qb_len field in the header structure is correct, however.

The upshot of all this is that simply encoded User Data is not included in the encoded PPDU, as qb2prim specifically tests the qb_len field.

A proposed solution is to insert the following just after the

```
for (d = data, i = 0; i < ndata; i++)
```

processing loop which calls info2qb:-

```
if (pdu->offset == type_PS_User__data_simple)
```



```

    {
        qb->qb_len = simple->qb_len;
        qb->qb_data = qb->qb_base;
    }

    return pdu;

```

This has worked successfully in our tests.

Best Regards
Andy Sykes

Andrew Sykes Micro-Electronics Consultants Ltd.
21 Willow Green, Needingworth, Cambridgeshire, PE17 3SW, UK
Tel/FAX: +44 (0) 480 69771 E-Mail:
andy@asmec.uucp

or

..mcsun!ukc!asmec!andy
isode-RE NIC 8/20/90

Andy Sykes
info2ppdu

Bug-ISODE@nisc.nyser 8/20/90 Bug in

*** MAIL 6 ***

To: andy@asmec.UUCP
Cc: Bug-ISODE@nisc.nyser.net, isode@nic.ddn.mil
Reply-To: Bug-ISODE@nisc.nyser.net
Subject: Re: Bug in info2ppdu
In-Reply-To: Your message of Mon, 20 Aug 90 17:45:58 +0000.
<26d021d6.asmec@asmec.uucp>
Date: Mon, 20 Aug 90 14:09:34 -0700
Message-Id: <5099.651186574@cheetah.nyser.net>
From: Marshall Rose <mrose@cheetah.nyser.net>

Thanks. Could you

1. Use just the bug-isode list in the future; and,
2. Send a context diff. From your report, I can not deduce what you suggest as the actual fix.

/mtr

isode-RE NIC 8/20/90

Marshall Rose
info2ppdu

andy@asmec.UUCP

8/20/90*Bug in

*** MAIL 7 ***

Date: 30 Sep 90 00:22:43 GMT
From: agate!linus!linus!mbunix.mitre.org@apple.com (Trei)
Organization: The MITRE Corp., Bedford, MA
Subject: Need some help with NIST ISODE.
Message-Id: <121729@linus.mitre.org>
Sender: isode-relay@nic.ddn.mil
To: isode@nic.ddn.mil

Help!

We seem to have come up against a serious shortcoming in the ISODE 6.0 tsaplisten facility. This is what is happening:

We have two or more processes that want to listen at the same transport address (at mutually exclusive times). But, the first process performs its TNetListen and TNetAccept just fine, but then when it does a TNetClose to stop listening on the transport address, it turns out that the address is still bound, and can't be listened on by another process until the first process exits. This is somewhat of a dilemma for us.

If anyone knows how to work around this problem, please mail to wheeler@mbunix.mitre.org. BTW, we are running ISODE-6.0 on a Sun 3 with SunLink 6.0 (using TP4, not X.25).

Thanks in advance.
isode-RE NIC 10/06/90

Trei
help with NIST
ISODE

isode@nic.ddn.mil

9/30/90 Need some

*** MAIL 8 ***

Date: Thu, 11 Oct 90 11:34:00 EDT
To: isode@nic.ddn.mil
From: Peter (P.W.) Whittaker <PWW@BNR.CA>
Subject: ISODE Vol.5 doc error?
Sender: Peter (P.W.) Whittaker <PWW@BNR.CA>
Message-Id: <90Oct11.115420edt.57580@ugw.uts.utoronto.ca>

My apologies if this has already been pointed out, but I've only been reading this group for a few months....

According to the QUIPU manual (ISODE Vol. 5, Ver. 6.0) page 155, dsap_init will ignore all arguments except a "-c dsa_name" pair.

This isn't strictly the case: dsap_init will ignore arguments that occur after "-c dsa_name" IF they are not "-flag" style arguments (i.e. if they are "flag" style), and will reject (i.e. generate an error message and exit your application) any "-flag" style arguments where "-flag" is not one of the ones listed below.

In fact, dsap_init will ignore ANY argument that occurs after a "flag" style argument, including the "-c dsa_name" argument. This is due to a function (tai_args()) called by dsap_init; To be safe and sure, dsap_init should only be passed an argv-style argument of the form:

your_appl_name :-c dsa_name: :-t dsaptailor_style_tailor_file:
 :-T oidtable_format_file:

OR

two NULL pointers (in which case the only tailoring dsap_init will do is to read your \$HOME/.isode_tailor file, if it exists).

(as usual, the :...: indicate that these arguments are optional).

What will dsap_init do with your options?

If you do not pass dsap_init(NULL, NULL), then it will:

1) look in your home directory for .your_application_name_tailor, an isodetailor-style file, which it will read for various tailoring options (log level settings, ETCDIR redefinitions).

If the HOME environment variable is not set, it will try to find this file in the local directory.

If it can't find it, it ignores it.

2) set dsa_address to the dsa_name following the '-c' flag; subsequent binds will connect to that dsa.

3) read the dsaptailor style file specified after the '-t' flag, which it will read for dsa addresses and names.

4) read the oidtable specified after '-T' (instead of ETCDIR/oidtable).

In case of inaccuracies or omissions in the above, please e-mail the correct information to me, and I will repost this article.

Thanks,

Peter W.

pww@bnr.ca

isode-RE NIC 10/11/90

Peter (P.W.) Whitta isode@nic.ddn.mil 10/11/90 ISODE Vol.5
doc error?

*** MAIL 9 ***

Date: 13 Oct 90 04:00:23 GMT
From: van-bc!ubc-cs!kiwi!newshost!morse@ucbvax.Berkeley.EDU
(Daryl Morse)
Organization: MPR Teltech Ltd., Burnaby, BC, Canada.
Subject: HELP!! Segmentation faults:
Ultrix-4.0/ISODE-6.0/OSIMIS-2.1
Message-Id: <MORSE.90Oct12210023@quark.mpr.ca>
Sender: isode-relay@nic.ddn.mil
To: isode@nic.ddn.mil

We are trying to bring up (read: keep up) OSIMIS and ISODE on DECstation 5000s running Ultrix 4.0, but are encountering segmentation faults under load, as well as lots of noise due to unaligned accesses.

We turned off the unaligned access messages with the uac command, but the segmentation faults are a brick wall.

My questions are directed at users of OSIMIS and ISODE running

DECstations, as well as DEC software support. We are not sure whether

the segmentation faults are a result of the software or the operating system. The unaligned access messages are clearly caused by the code.

(The Sun4 compiler has a handy-dandy "-misaligned" flag, which seems

to have averted problems on that platform.) Ultrix supposedly fixes-up

problems due to misalignment on the fly, as the numerous error

messages used to indicate before we shut them off. However, we already

are running a patched version of vm_drum.o, which was necessary to

enable us to run a well-known OO development environment without

crashing the system on a regular basis.

1. Is the runtime fix-up known to be robust under all circumstances, or will it break under pathological conditions? Will we know if it breaks?

2. Has anyone been able to get OSIMIS and ISODE to run reliably on a

DECstation (with any version of Ultrix)? If so, did you encounter similar problems?

3. Are there any known problems with Ultrix 4.0 that cause segmentation faults? Are any patches available? (We are desperate!)

Could the patch to vm_drum be causing other problems? (Sorry, we

received only a compressed binary, without any official documentation

(ie. no SPR or patch number). I do have the name and email address of

a DEC contact for it, however.)

I will be eternally grateful for any assistance offered. Please reply

via email. I will post the solution when I find it. Thanks.

--

Daryl Morse

MPR Teltech Ltd.

8999 Nelson Way, Burnaby, BC

Canada, V5A 4B5

quark.mpr.ca!morse@uunet.uu.net

isode-RE NIC 10/13/90

Daryl Morse

Segmentation faults:

U1

| Voice : (604) 293-5476

| Fax : (604) 293-5787

| E-Mail: morse@quark.mpr.ca

|

isode@nic.ddn.mil

10/13/90 HELP!!

*** MAIL 10 ***

Date: 19 Oct 90 17:59:50 GMT

F r o m :
van-bc!ubc-cs!news-server.csri.toronto.edu!utgpu!cunews!bnrg
ate!pww@ucbvax.

Berkeley.EDU (Peter Whittaker)

Organization: Bell-Northern Research, Ltd., Ottawa, Ontario,
CANADA

Subject: Re: Problems with str2aei()..

Message-Id: <1990Oct19.175950.19743@bnrgate.bnr.ca>

References: <559@.betaal.UUCP>

Sender: isode-relay@nic.ddn.mil

To: isode@nic.ddn.mil

In article <559@.betaal.UUCP> murthy@betaal.UUCP (C. V. R.
Murthy) writes:

>Hello!

>the following entry gives me a core dump:

```
> static LLog _pgm_log = {  
>   "/usr2/isode/isode-5.0/log/logfile", NULLCP, NULLCP,  
>   LLOG_FATAL | LLOGEXCEPTIONS | LLOG_NOTICE,  
>   -1,  
>   LLOGCLS | LLOGCRT | LLOGZER,  
>   NOTOK  
> };  
> LLog *pgm_log = &_pgm_log;  
>
```

>Thanks in advance

>& waiting eagerly for responses,

>

You need one more line (you took this structure straight from
the manual,
without checking the argument count - naughty, naughty :-)).

.....

```
LLOG_FATAL | LLOGEXCEPTIONS | LLOG_NOTICE,  
LLOG_NONE, /* no syslog logging */  
-1,
```

.....

That should do it.

ISODE is really quite neat, and fun to use (not to mention useful, functional, and free.... :@)) but the manuals run at about 75% accurate.

(For further news, join the QUIPU discussion group:
QUIPU-REQUEST@cs.ucl.ac.uk
It's not on the Internet at large (as far as I know), but is available as a mail-out to any interested party).

--
Peter W. (pww@bnr.ca)

*** MAIL 11 ***

Date: 18 Oct 90 10:47:20 GMT
From: shakti!betaal!murthy@uunet.uu.net (C. V. R. Murthy)
Organization: ERNet, Dept of CSE, IIT Bombay - 76
Subject: Problems with str2aei()..
Message-Id: <559@.betaal.UUCP>
Sender: isode-relay@nic.ddn.mil
To: isode@nic.ddn.mil

Hello!

This is our first posting onto this USENET newsgroups; I was looking for a newsgroup concerned with the ISODE. I really do not know if this the right group; appreciate if someone could place this article to a proper people/forum.

WE have recently installed the ISODE Rel 5.0 on our 386 Unix system V with TCP/IP. It works fine; at least quipu DSA and DUA can interact.

I wanted to experiment with the lookup daemon and see how the remote operations(ry) specs can be implemented using the rich tools available under ISODE.
I have defined a new service:

```
ernet passwdstore 1.17.4.1.7 "" "" #1040
```

in the TOPDIR/etc/isoentities.

I compiled the ros.lookup daemon (had some problems with some call to isodeserver() but I'll explain later) and inst-alled. When I ran ros.quipu daemon, I get the following error:

ernet-passwdstore: unknown application-entity

Then I wanted to confirm this with the acsapttest program; the acsapttest when it was run without any argument, it shows an

entry as follows:

ISO Entities Database:

...

Entity: ernet-passwdstore (1.17.4.1.7)

AE Info: <1.17.4.1.7,,,>


```

Address: '0410'H/
{
    tSelector '0410'H,
    nAddress{}
}

```

...

But when I run
% acsptest ernet passwdstore
then I have no luck!

(The ns_enable: field was set to on first then it called
str2aei_dse() and it was set to off in another try;
the net_result is same!).

Can anybody suggest me what is going wrong ?

2. Further, I do not really understand the isodeserver()
mechanism
provided in the ry modules; how should one go about writing
this?

Is there any sample program for the lookup daemon?

3. And lastly, I seem to be having problems with logging
messages:

the following entry gives me a core dump:

```

static LLog _pgm_log = {
    "/usr2/isode/isode-5.0/log/logfile", NULLCP, NULLCP,
    LLOG_FATAL | LLOGEXCEPTIONS | LLOG_NOTICE,
    -1,
    LLOGCLS | LLOGCRT | LLOGZER,
    NOTOK
};
LLog *pgm_log = &_pgm_log;

```

Thanks in advance
& waiting eagerly for responses,

---Murthy.

--

*** MAIL 12 ***

From: solomon@cs.wisc.edu (Marvin Solomon)
Message-Id: <9010251231.AA02465@gjetost.cs.wisc.edu>
Received: by gjetost.cs.wisc.edu; Thu, 25 Oct 90 07:31:02
-0500
Subject: problems with asynchronous lpp (fwd)
To: isode@nic.ddn.mil
Date: Thu, 25 Oct 90 7:31:00 CDT
X-Mailer: ELM [version 2.2 PL10]

I sent the following message to bug-isode@nisc.psi.net over a month ago and got no response whatsoever, not even an ack. I tried again a few days ago and still have no response. So I'm forwarding it to the whole list in the hopes that somebody can help me out. This message is partially a bug report about ISODE (at least I think it's a bug, but it might be a "feature"), partially a complaint about what appears to be a bug in the lpp spec, and partially a question about how the asynchronous association feature of ISODE was really meant to be used. If you have something to contribute about any of these issues, please respond to me (I'll summarize responses) or directly to the list, if you think it's of general interest. Thanks.

By the way, I also have a plain old bug report that sent to bug-isode with no ack. I'll forward that to the list in a separate message.

> From solomon Mon Sep 24 06:59:28 1990
> From: solomon (Marvin Solomon)
> Message-Id: <9009241159.AA18696@gjetost.cs.wisc.edu>
> Subject: problems with asynchronous lpp (fwd)
> To: bug-isode@nisc.psi.net
> Date: Mon, 24 Sep 90 6:59:18 CDT
> Cc: nhall (Nancy Hall), solomon (Marvin Solomon)
> X-Mailer: ELM [version 2.2 PL10]
>
> Hi someone (perhaps Marshall).
> I've been having some problems with the UDP variant of the 1085 code in
> ISODE 6.0 and hope you can help a bit. At first I thought

it was
 > simply a bug report, but now I'm not so sure.
 >
 > I want to write a client that probes several servers using
 udp/lpp.
 > Since a server may take a while to respond, I want to use
 > AcAsyncAssocRequest to "start the ball rolling" and come back
 later
 > with AcAsyncRetryRequest to complete the handshake. My first
 problem
 > was that AcAsyncAssocRequest was not returning immediately,
 even though
 > I was setting async=1. Upon plowing through the code, I
 discovered
 > that udpopen calls udpretry, which sends the connection
 request packet
 > and then does a select with a timeout of WAITRIES (30
 seconds!). I
 > kludged the code to change the timeout to OK (0) if udpopen
 is called
 > with async!=0. [Udpopen doesn't pass the async flag to
 udpretry and I
 > couldn't simply add an argument because of the way udpretry
 is called,
 > so I put the timeout in a global and modified it just before
 the call.
 > Ugly, but effective.]
 >
 > With that change in place, AcAsyncAssocRequest properly
 returns
 > CONNECTING_1. However, when I later call
 AcAsyncRetryRequest, the
 > server complains about unexpected PDU 1, and aborts the
 association.
 > It seems that AcAsyncRetryRequest also calls udpretry, which
 retransmits
 > the connection request pdu. RFC1085 says that a connection
 request pdu
 > in state WAIT2 or DATA causes the association to be aborted.
 I would
 > argue that that's a mistake; the responder should simply
 discard the
 > duplicates (easily recognized as such because of the
 connection
 > reference). Am I missing something obvious? Be that as it
 may, I
 > can't see how AcAsyncRetryRequest can possibly work correctly
 with the
 > current setup. Unless you can point out the error of my
 ways, I'm
 > planing to dig in and simply make udpretry discard the CR

pdu after

- > sending it once That kills the possibility of using retransmission to
- > recover from lost CR pdu's, but I think that's of marginal utility
- > anyhow, since the way 1085 is written, it can only recover from lost
- > CR's, not lost (or slow) CC's. Besides, as somebody once said "Low
- > quality means low quality!".
- >
- > On a related subject, I just can't seem to get the point of TSelectMask
- > and friends. The little tutorial on asynchronous requests in section
- > 4.2.3 of Vol II says in a couple of places "At some point in the future
- > call TSelectMask to get an argument for xselect". In the case of udp,
- > TSelectMask actually does selects of its own, so it appears to me that
- > you can't just call TSelectMask once immediately after the ConnRequest
- > and squirrel away the result for use later. But doesn't "polling"
- > TSelectMask for each of the relevant td's kind of defeat the purpose of
- > xselect? Suppose I want to fire off a bunch of connect requests and
- > then sit back and wait until one of them completes (or I get keyboard
- > input, or a message from the X server or an RPC message from another
- > process or any of a variety of other things). When should I call
- > TSelectMask? What I'm doing now is to manually build a big select mask
- > including all the file descriptors I care about that are unrelated to
- > ISODE, as well as all the td's (or sd's or whatever) I know about from
- > AcAsynConnRequest or AcInit. If one of the latter shows any activity,
- > I do AcAsynRetryRequest or RyWait, depending on my idea of the state of
- > the connection. I don't use ?SelectMask at all. It seems to
- > more-or-less work, but I still have the feeling I'm doing something
- > very naughty. A word or two of advice would be much appreciated.

>
> Sorry for the long-winded questions. I'll try not to bug
you too often
> (not more once every few months). Thanks for any light you
can shed on
> this matter.
> --Marvin
>
>
isode-RE NIC 10/25/90
Marvin Solomon isode@nic.ddn.mil 10/25/90 problems with
aynchronous
lpp

*** MAIL 13 ***

From: solomon@cs.wisc.edu (Marvin Solomon)
Message-Id: <9010251237.AA02485@gjetost.cs.wisc.edu>
Received: by gjetost.cs.wisc.edu; Thu, 25 Oct 90 07:37:07
-0500
Subject: uninitialized variable (fwd)
To: isode@nic.ddn.mil
Date: Thu, 25 Oct 90 7:37:05 CDT
X-Mailer: ELM [version 2.2 PL10]

The following is a bug in ISODE 6.0. Marshall, don't tell me
to RTFM and
and send it to bug-isode. I did and received no response.

Forwarded message:

> From solomon Wed Sep 26 05:31:46 1990
> From: solomon (Marvin Solomon)
> Message-Id: <9009261031.AA23208@gjetost.cs.wisc.edu>
> Subject: uninitialized variable
> To: bug-isode@nisc.psi.net
> Date: Wed, 26 Sep 90 5:31:38 CDT
> Cc: solomon (Marvin Solomon)
> X-Mailer: ELM [version 2.2 PL10]
>
> Reference: isode 6.0, compat/dgram read_dgram_socket
> Description: The variable mask is used but never initialized

> Fix: about line 394, replace
> FD_ZERO (&ifds);
> by
> FD_ZERO (&mask);
>

isode-RE NIC 10/25/90
Marvin Solomon isode@nic.ddn.mil 10/25/90 uninitialized
variable (fwd)

*** MAIL 14 ***

Date: Mon, 29 Oct 90 09:57:00 EST
To: isode@nic.ddn.mil, quipu-request@cs.ucl.ac.uk
From: Peter (P.W.) Whittaker <PWW@BNR.CA>
Subject: Memory leaks in ISODE stack.
Sender: Peter (P.W.) Whittaker <PWW@BNR.CA>
Message-Id: <90Oct29.110651est.57692@ugw.utcs.utoronto.ca>

Hello, we're running ISODE 6.1+ patches on SUNs (4.0.3) and HP-UX 6.5,
and we seem to be having memory leakage problems.

We are already aware of certain memory leaks in the QUIPU-specific code (entry.c, oper_act.c, update.c) but these problems seem to be more at the Transport Layer of ISODE.

The leaks were discovered by a test case that attempts several thousand bind/unbind operations. On a SUN with 65 Meg, it takes between 10- and 20 000 of these operations for the DSA to fail, while a DUA on an H/P with 8 - 16 M fails after about 6000 operations. Furthermore, when the DUA starts its binding, it manages 3 or 4 bind/unbind pairs a second. After about an hour, it's down to 1 every 3 or 4 seconds (i.e. the DUA grows and grows, becoming more and more of a load on machine resources, esp. pager and swapper).

The transport layer is suspected for two reasons:

i) there's a synching problem with the ISODE stack as well:

if three DUAs on three separate machines all attempt the bind/unbind

tests simultaneoulsy, one will invariably "win out", and complete

its bind/unbinds in 'normal' time. The other two will 'lock' their

TCP connections, and not 'hang' until they, or the DSA, are killed.

(This 'winning out' usually manifests itself after about 60 minutes

- roughly 500 binds per DUA).

(All data in the TCP exchange is good (no bad packets, dropped packets,

or collisions): this thanks to a LanAnalyzer.)

The DSA writes in its stats.log that it received a ds_unbind-request, and that the connection was dropped, but the connection persists.

So, we suspect that somewhere in the application stack a routine

'forgets' to either set or clear some data object.

ii) Because the bind/unbind operations do not involve any other DUA/DSA operations, there is a minimal amount of ASN.1/BER manipulation going on, so the presentation and session layers aren't suspected: ergo, transport layer.

(a third circumstantial reason is that the first error reported when the DUAs fail is A-ASSOCIATE.REQUEST out of memory, accompanied by

TAsynConnRequest: Congestion at TSAP :out of memory:
SAsynConnRequest(pseudo): Congestion at SSAP
ppktlose :Temporary congestion:
PAsynConnRequest(pseudo): Temporary congestion

The 'circumstantial evidence' is that the transport function is the first that reports an error.)

Has anyone else had similar problems with ISODE?

Thanks,

Peter W.

isode-RE NIC 10/29/90
Peter (P.W.) Whitta isode@nic.ddn.mil 10/29/90 Memory leaks in ISODE stack.

*** MAIL 15 ***

To: PWW@bnr.ca (Peter Whittaker, P.W.)
Cc: quipu-support@cs.ucl.ac.uk
Reply-To: quipu-support@cs.ucl.ac.uk
Cc: isode@nic.ddn.mil
Subject: Re: Memory leaks in ISODE stack.
In-Reply-To: Your message of Mon, 29 Oct 90 09:57:00 -0500.

<90Oct29.110651est.57692@ugw.utcs.utoronto.ca>

Date: Mon, 29 Oct 90 14:18:34 -0800
Message-Id: <25201.657238714@cheetah.ca.psi.com>
From: Marshall Rose <mrose@cheetah.ca.psi.com>

There were a few memory leaks in 6.0+. These are believed to
be fixed
in the current beta, 6.6

/mtr
isode-RE NIC 10/30/90
Marshall Rose Peter Whittaker, P. 10/29/90*Memory leaks
in ISODE stack.

*** MAIL 16 ***

Date: Wed, 31 Oct 90 08:54:10 -0500
From: sean@dsl.pitt.edu (Sean McLinden)
Message-Id: <9010311354.AA02173@cadre.dsl.pitt.edu>
To: PWW@BNR.ca, isode@NIC.DDN.mil, quipu-request@cs.ucl.ac.uk

Subject: Re: Memory leaks in ISODE stack.

I, too, have experienced memory leaks on DEC3100 (MIPS) running Ultrix 3.1, 3.2, and 4.0. At first I thought that it was due to problems with the MIPS C compiler, and was going to try compiling with GNU, but with warnings enabled I got nearly 200 pages of possible problems and gave up trying to find out where they all were. Most of our problems stem from running QUIPU. It would run about 18 hours and then die. Restarting got to be a problem because the startup DSA gets overwritten once the data is fetched from the master, but what is fetched cannot be used to restart the DSA. If there are patches to the QUIPU code, that might help. I had always assumed that the problem was there because we could run the rest of the ISODE for days without problems (but then, nothing banged on it as much as QUIPU).

Sean McLinden
Decision Systems Laboratory
University of Pittsburgh
isode-RE NIC 10/31/90
Sean McLinden PWW@BNR.ca
in ISODE stack.

10/31/90*Memory leaks

0000109