INFORMATION RETRIEVAL (IR)
KERNEL - TARGET
INSTALLATION MANUAL

INFORMATION RETRIEVAL (IR) KERNEL - TARGET INSTALLATION MANUAL

7 June 1993

Prepared for:

National Library of Canada
Hull, Quebec
K1A 0N4

Prepared by:

Software Kinetics Limited
900 Windmill Road
Suite 107
Dartmouth, Nova Scotia
B3B 1P7

Software Kinetics Document # H4900-01-47.01

DOCUMENT APPROVAL SHEET

FOR THE

INFORMATION RETRIEVAL KERNEL TARGET

Document Number: H4900-01-47.01

Document Name: Installation Manual

Approvals	Signature	Date
Project Engineer	____________________ B. Yen	________
Project Manager	____________________ G. Coughtry	________
General Manager, Atlantic	____________________ R. Kerr	________
Quality Assurance	____________________ T. Bushell	________

Document Revision History

Revision	Reason For Changes	Origin Date
01	New Document Issued, Version 1.0	7 June 1993

TABLE OF CONTENTS

1.0 SCOPE

1.1 Identification

This Installation Manual describes the implementation of both the ISO Search and Retrieve (SR) and ANSI Z39.50 application protocols, and the procedures required to install the Generic IR Target Software on a development system. The ANSI Z39.50 application protocol is a superset of the ISO Search and Retrieve standard service and protocol. The use of the IR term within this document is used to reference both the ISO SR and ANSI Z39.50 application protocols.

1.2 System Overview

The IR standards define separate roles for the Origin and Target systems. Using the Origin software, a user queries a remote database for specific information. The type of database queried is irrelevant to the user. The Target system accepts connections from an Origin system and processes Init, Search, Present, Delete and Resource-Report requests. With the exception of the Init request, the remaining requests are passed by the Target system to a database application which processes the request.

The Target system design (Software Kinetics Document #H4900-01-40.01, Detailed Design Document) describes a set of well defined interfaces (Software Kinetics Document #4900-01-37.01, Interface Specification) among system components to facilitate the integration of all or limited parts of the Target into a wide range of environments. The design of the Target system incorporates the notion that the Target system will be ported to multiple operating platforms. The Target system, as a whole, has been designed to allow for the provision of executing portions of the Target system on different platforms. This implementation is designed primarily to support bibliographic information retrieval; however, effort has been made not to preclude non-bibliographic usage. In addition, this implementation is designed to use TCP communications, but consideration has been given to future implementations making use of a full OSI protocol stack.

1.3 Document Overview

This document contains the following sections:

- SCOPE - the purpose and intent of the document;
- REFERENCES - list of documents that have been consulted during the preparation of this manual;
- VERSION DESCRIPTION - detail information about the current software version of the Generic IR Target Kernel;
- DEVELOPMENT ENVIRONMENT - defines the development environment required to port the Generic Target Kernel;
- INSTALLATION - describes the installation procedures required to install the Generic IR Target Kernel; and
- NOTES - contains a listing of abbreviations and acronyms referenced in this manual.

2.0 REFERENCED DOCUMENTS

The following documents have be referenced in this manual or consulted during its preparation:

- H4900-01-13.03 IR Target Kernel Software Requirement Specification, 06 January 1993;

- H4900-01-37.01 IR Target System Interface Requirement Specification, 07 June 1993; and

- H4900-01-40.01 IR Target System Detail Design Document, 07 June 1993.

3.0 VERSION DESCRIPTION

3.1 Software Licensing

The IR software is provided by the National Library of Canada as freeware. If any problems with this software occur during testing, the National Library requests that all trouble reports be submitted to Ms. Fay Turner (Internet address: fst@its.nlc-bnc.ca) National Library of Canada, Hull, Quebec, K1A 0N4. The National Library also requests that the user register with Ms. Turner for trouble reports and recommended fixes.

3.2 Inventory Of Materials

The IR Kernel software is available in two formats:

- 5.25 inch diskettes formatted for an IBM compatible PC running DOS; and
- Unix tar format available through the Internet via anonymous ftp.

The address of the host machine or the DOS installation diskettes can be obtained from the National Library.

Irrespective of the format, the following information is supplied:

- a README file of information on last minute changes;
- a source file to confirm the presence of required files;
- source files to build the IR kernel executable 'irtarget';
- source files to build the background Init RPC task executable 'tpstartu';
- script files to build the target process black box executable 'tpbbox';
- a make file 'Makefile' for producing the above executables;
- ASN.1 output files: encdec.c and apdu.h which were produced from the ASN.1 definition file ir.asn;

- GNU SNACC ASN.1 header files;
- a copy of the GNU SNACC ASN.1 library files which were built for SUNOS version 4.1.1. using GNU C compiler version 2.2.2 with C option (not C++).

  Note: Rebuilding of the ASN library files is unnecessary if you have the matching environment. However, if you are running a different version of Operating System and GNU C compiler, the ASN library files must be rebuilt. Refer to section 4.1.3 of this document for information needed to retrieve the GNU SNACC ASN.1 compiler; and
- refer to section 2.1 for a list of related documentation files included on the installation diskettes.

3.3 Inventory Of Packing Contents

This section describes:

- the contents of the tar file that was obtained through anonymous ftp, and
- the contents of the floppy diskettes that were obtained from the National Library.

Uncompressing the tar file or copying (using XCOPY) the contents of the floppy disk will create a sub-directory named 'irtarget'. Within this sub-directory another directory will be created called 'irtarget/doc'.

The information contained on the installation media contains the following files:

- the make file for compiling and creating the executables: irtarget, tpstartu, and tpbbox;
- the default parameter file for the irtarget software;
- the 'C' language source files that comprise the irtarget software;
- the 'C' language header files for the irtarget software;

- the black box (test software) source, header and irtarget support software;
- the SNACC ASN.1 header files; and
- the SNACC ASN.1 Library files.

For a list of the source and header files, see Appendices A through E.

3.3.1 General Make File

The make file (**Makefile**) for compiling and creating the executables: irtarget, tpstartu and tpbbox, has the following options:

make all	compile and produce all executables
make clean	delete all the related object (*.o) files
make irtarget	compile and produce the executable irtarget
make tpstartu	compile and produce the executable tpstartu
make tpbbox	compile and produce the executable tpbbox

3.3.2 Default IR Target Parameter File

The default parameter file **'irtarget.def'** for the irtarget software has the following format:

Line 1:	Comment line
Line 2:	<Debug file name #1 for tracing the flow of the program> <space(s)> <comment>
Line 3:	<Debug file name #2 for displaying the content of a PDU or an APDU> <space(s)> <comment>
Line 4:	<Debug file name #3 for displaying the variables> <space(s)> <comment>
Line 5:	<Error log file name for logging the errors> <space(s)> <comment>
Line 6:	<Init response timeout in minutes> <space(s)> <comment>
Line 7:	<Search response timeout in minutes> <space(s)> <comment>
Line 8:	<Present response timeout in minutes> <space(s)> <comment>
Line 9:	<Delete response timeout in minutes> <space(s)> <comment>
Line 10:	<Trigger resource control response timeout in minutes> <space(s)> <comment>

Line 11:	<Access control response timeout in minutes> <space(s)> <comment>
Line 12:	<Resource control response timeout in minutes> <space(s)> <comment>
Line 13:	<Idle/No activities timeout in minutes> <space(s)> <comment>
Line 14:	<Maximum message size in bytes> <space(s)> <comment>
Line 15:	<Maximum number of error records>
Line 16:	<Debug file name #1 for tracing the flow of the program (for tpbbox)> <space(s)> <comment>
Line 17:	<Debug file name #2 for displaying the content of a PDU or an APDU (for tpbbox)> <space(s)> <comment>
Line 18:	<Debug file name #3 for displaying the variables (for tpbbox)> <space(s)> <comment>

4.0 DEVELOPMENT ENVIRONMENT

4.1 Software Environment Required

The Generic IR Target Kernel was developed on a SPARC station using SunOS version 4.1.1 UNIX environment supporting TCP/IP. If the installation platform meets this configuration, no changes are required to the IR Target software. For other platforms, the following sections outline issues to be considered before attempting the port.

4.1.1 POSIX/UNIX Compliance

The operating system services that the Target Kernel uses are POSIX compliant and will facilitate porting the kernel to other platforms with operating systems that support the IEEE POSIX 1003.1 standard. The interface to UNIX is provided through the ANSI C standard as supported by the GNU C compiler version 2.0. The ANSI C standard is supported by numerous hardware platforms that will ensure portability of the Target Kernel across different platforms.

4.1.2 TCP/IP

The communication transport interface development package is provided by SUN Microsystems. This development environment provides the runtime libraries required to support the Berkeley Socket Application Program Interface.

4.1.2.1 RPC/XDR

The Target System uses Sun Microsystems' Open Network Computing consisting of:

- Remote Procedure Calls; and
- eXternal Data Representation standard (XDR).

The Sun RPC and XDR typically utilize either UDP or TCP as the transport layer. The RPC/XDR utility is one method used by the Target Kernel to invoke IR services. RPC/XDR is a widely accepted approach for transferring control from one process to another. RPC/XDR is available on most platforms that support the TCP/IP protocol, and is often bundled with the Network File System (NFS) which itself is built upon RPC/XDR.

4.1.2 GNU C Compiler

The GNU C compiler version 2.0 was used to develop the Generic IR Target Kernel. This compiler is available from the Free Software Foundation and has been ported to a number of different hardware platforms. This compiler was chosen because it supports the ANSI standard within the SunOS development environment. The GNU C compiler is available via anonymous ftp from site prep.ai.mit.edu.

4.1.3 GNU SNACC ASN.1 Compiler

Information that is transferred from the Client to the Target (and vice versa) is first encoded according to Basic Encoding Rules (BER). Given an Abstract Syntax Notation (ASN.1) source file, the SNACC compiler produces reasonably efficient C or C++ routines and data structures to support BER encoding and decoding values of the given ASN.1 data structures.

The IR ASN.1 source file has been compiled and the resulting source code and SNACC BER encode/decode libraries have been included in this distribution. Should you require changes to the IR ASN.1 source file, it will be necessary to obtain and install the SNACC ASN.1 compiler. The SNACC ASN.1 compiler is available via an anonymous ftp at site cs.ubc.ca.

Please refer to Appendix F for details about GNU General Public License.

4.2 Interface Compatibility

Three interfaces currently exist between the IR Target Kernel and the Target Process:

- RPC/XDR Layer Interface;
- Application Layer Interface; and

- Presentation Layer Interface.

4.2.1 RPC/XDR Layer Interface

The RPC/XDR layer interface is used to initiate a Target Process. The IR Target Kernel will start a new Target Process using the remote procedure call mechanism. At this time, any relevant data are exchanged using XDR. Once a Target Process is started, the Application and Presentation layer interfaces are used to exchange Application Protocol Data Units (APDUs).

4.2.2 Application Layer Interface

The application layer interface consists of a number of routines that are used to communicate between the IR Target Kernel, the Target Process and the IR Client. The following table provides a summary of these routines:

IR Target

IR Kernel Routine Name	Description
ira_ALI_setup	Application Layer setup
ira_comm_event	Processes communication events
ira_send_request	Sends a Request APDU
ira_send_response	Sends a Response APDU
ira_service_request	Sends a service Request APDU to the Target Process
ir_service_response	Sends a service Response APDU to the Target Process
ira_invoke_target_process	Invokes a new target process using the remote procedure call
ira_link_to_init	Establishes a TCP link between the Kernel and the Target Process
ira_terminate	Called by the IR Target Kernel to terminate a Target Process

IR Kernel Routine Name	Description
ira_abort_request	Called by the IR Target Kernel to abort an IR client/server association.

Test Software (Black Box)

Target Process Routines	Description
tpa_ALI_setup	Target Process Application Layer setup
tpa_service_event	Target Process service event
tpa_send_APDU	Target Process send request/response APDU

4.2.3 Presentation Layer Interface

The Presentation Layer routines perform the Basic Encoding functions to provide a mechanism to exchange APDU's. The following table summarizes these APDU's:

Presentation Layer Interface Routine	Description
cs_PLI_setup	Presentation Layer Interface setup
cs_Pconnect_request	Called by the IR Target Kernel to accept an Origin connection request
cs_Pdata_request	Used to apply the BER encoding rules to an ASN.1 data structure
cs_Pdata_indication	Used to apply the BER decoding rules to an incoming APDU
cs_Prelease_request	Used by the IR Target Kernel to release a specified client/server association
cs_P_pabort_request	Used by the IR Target Kernel to abort a specified client/server association.

4.3 Operational Description

Three components are required to setup and test the system:

- an IR Client;
- the Generic IR Target; and
- a Target Process (a black box process has been provided).

With these three components, it is possible to issue queries from the client and receive pre-packaged responses from the black box Target Process.

The Target Process black box provides a working model of the program interface between the Generic IR Target Kernel and a Target Process. These components can be used to perform a series of tests that will facilitate the integration of a customized Target Process.

5.0 INSTALLATION

5.1 Installation Instructions

These installation instructions assume that all software listed in section 4.1 is installed. Installation instructions are provided by the vendors for each of the products required. If the IR kernel software was obtained from anonymous ftp, follow the steps outlined in section 5.1.1. If the IR kernel software was obtained on a diskette, follow the steps outlines in section 5.1.2.

5.1.1 Installation From Anonymous FTP

If the IR kernel software was obtained from anonymous ftp, the following steps must be followed.

1. Review the README file for last minute changes.

2. Move the 'irtarget.tar.gz' file to the directory in which the installation will be performed.

3. Uncompress the irtarget.tar.gz file by entering the following commands:

   gzip -d irtarget.tar.gz
   tar -xvf irtarget.tar

   This operation will create two sub-directories: **'irtarget'** and **'irtarget/doc'**.

   The sub-directory **irtarget** contains all the source and header files required to compile and link the IR target and black box software. A detailed description of the files contained in these directories is listed in Appendices A through F.

   Note: gzip utility is available via anonymous ftp at site prep.ai.mit.edu.

4. The IR target consists of the IR kernel and the target process. The IR kernel and target process software can be run in two paradigms. The first paradigm has both the IR kernel and target process software executing on

the same hardware platform. The second paradigm has the IR kernel and target process software each running on different hardware platforms.

a) If you wish to install the IR kernel and target process software on the same machine, change the current directory to 'irtarget' and compile the source files by entering:

cd irtarget
make all

These commands will produce three executables: irtarget, tpstartu, and tpbbox.

b) If you wish to install the IR kernel and target process software on separate hardware platforms, create the IR kernel executable with the following commands:

cd irtarget
make irtarget

Move to the second hardware platform and perform steps 2 and 3 outlined above. To create the target process executables, enter the following commands:

cd irtarget
make tpstartu
make tpbbox

5. Adjust the parameter file 'irtarget.def' to suit the specific hardware platform environment (refer to section 3.3 for a more detailed description);

Note: The IR kernel program 'irtarget' uses TCP port number 2007 as the default TCP port number. However, the internet assigned Z39.50 TCP port number is 210. To use port 210, the IR kernel program needs root privilege. To give the IR kernel root privilege, it is recommended that the Set-User-Id approach in Unix be used.

5.1.2 Installation From Diskette

If the IR kernel software was obtained from diskettes it is assumed that the proper environment to execute the software is not available and the IR target software was obtained to study the source and header files. Therefore, the only required step is to move all source and header files to a new directory called 'irtarget' and to move all documentation to a new directory 'irtarget/docs'.

5.2 Install Verification Testing

The following tests may be used to verify that the installation has been successful:

1. Run the shell script 'irchecki' to check the existence of all the source files, other required files, documentation files, and the executables. If the script file irchecki is not in the sub-directory irtarget, it indicates the installation has failed.

2. Turn all debug levels on by having the following statements included in the header file irconst.h and tpconst.h:

   #define DEBUG_LEVEL1 1
   #define DEBUG_LEVEL2 2
   #define DEBUG_LEVEL3 3

   and re-compile the IR kernel programs by entering the following command:

   make clean
   make all

   The debug levels are defined as follows:

   LEVEL1: Display the calling sequence on all the functions and the internal logic flow of some functions.

   LEVEL2: Display the contents of the APDUs and PDUs.

   LEVEL3: Display the internal debugging variables and logic statements.

3. Execute the IR kernel programs. Refer to section 5.4 for how to run it.

4. Execute the client software and try to connect to the IR kernel system by matching the host name and TCP port number. If the client system can

receive the init response from the IR kernel system, it means that the IR kernel has been installed properly. However, if problems occur receiving the init response, try to look at the debugging output messages which are quite comprehensive and easy to spot where the problem is.

5.3 Integrating A New Database Engine

The target process black box supports all necessary protocol options to communicate with the IR kernel software. In addition, the target process black box accepts request APDUs and provides response APDUs corresponding to the IR kernel ANSI Z39.50 protocol state machine. To integrate a real database management system with the target process, the following functions of the target process must be modified. These modifications are required to perform the query as defined by the client and to populate the response PDU with the real data instead of the 'canned' responses of the existing system.

- Build the search query response PDU function tp_fill_search_response(),
- Build the present result set response PDU function tp_fill_present_response(),
- Build the delete result set response PDU function tp_fill_delete_response(),
- Build the resource report response PDU function tp_fill_resourceReport_response(), and
- Build the resource control response PDU function tp_fill_resourceControl_response().

5.4 Executing Software

Once the installation procedure is complete, the IR kernel software can be executed as follows:

1. Modify the parameter file **irtarget.def** to suit the requirements of the hardware platform;
2. Submit the RPC program **tpstartu** as a background task to allow loading the

target process black box program on the local or remote machine. Type

tpstartu&

Ensure that the target process program 'tpbbox' is in the search path of the RPC program 'tpstartu'. In other words, one of the following scenarios must be satisfied:

- the target process program 'tpbbox' must be in the same directory of the RPC program 'tpstartu'; or

- the target process program 'tpbbox' is in the operating system search path.

3. Execute the IR kernel program by entering the following command:

irtarget host=<host name> port=<port number>&

For Example:

irtarget host=tempest port=2007&

5.5 Troubleshooting

This section describes some common problems that may be encountered during installation of the IR software:

PROBLEM	SOLUTION
Missing software packages: for examples, tar, gzip, GNU C compiler, TCP communication package, GNU SNACC ASN.1 compiler etc.	Refer to section 3.2 and 4.1 for more information on how to acquire these files.
Insufficient disk space.	A minimum of 30 MB free disk space is required before installing the IR kernel and the target process software.

PROBLEM	SOLUTION
Installation on a VMS machine.	Insert the following statement as the first statement in ir.h and tp.h: **#define VMS 1**
Cannot open a TCP port.	First, ensure that the TCP/IP communication package was installed properly; secondly, some ports require root privilege to open them. Use the Set-User-ID approach or increase the TCP port number above 1024.

Installation on a machine that does not have the following configuration:

> SunOS Unix version 4.1.1 (AT&T Unix 5.3.2 equivalent) and GNU C compiler version 2.2.2.

could result in the following problems:

PROBLEM	SOLUTION
Failed to build executables because of compiler errors or link errors.	Add proper compilation and linking switches to the compilation command to ensure you are using the ANSI C standard and required Operating System libraries are being linked.
Cannot find system headers or system library files.	There are different directory layouts for different versions of UNIX, so modify the path in user header files to correct the problem.
There are differences between UNIX AT&T System V and BSD 4.2 UNIX flavours that may cause the BSD TCP functions to fail.	Re-map the structures and system calls, as required.
Incompatible SNACC ASN.1 compiler library files.	Rebuild the library files.

6.0 NOTES

The following list contains all the acronyms and abbreviations used within this document and their associated meanings.

ACSE	Association Control Service Element
ALI	Application Layer Interface
ANSI	American National Standards Institute
APDU	Application Protocol Data Unit
API	Application Programming Interface
ASE	Application Service Element
ASN.1	Abstract Syntax Notation Version 1
BER	Basic Encoding Rules
CANMARC	Canadian MARC record format
CPU	Control Processing Unit
DOS	Disk Operating System (i.e. Microsoft DOS)
GNU	GNU is not UNIX, software that implements UNIX functionality
gzip	GNU zip, a compression utility that replace the UNIX compress utility. It is available for many hardware platforms
IEEE	Institute of Electrical and Electronic Engineers
IR	Information Retrieval
ISO	International Organization for Standardization
MARC	Machine Readable Cataloguing
NFS	Network File System
OSI	Open Systems Interconnection
PC	Personal Computer
PDU	Protocol Data Unit
PLI	Presentation Layer Interface
POSIX	Portable Operating System Interface
RPC	Remote Procedure Call
SNACC	Sample Neufeld ASN.1 to C Compiler
SR	Search and Retrieve
TCP	Transmission Control Protocol
TCP/IP	The Internet Suite of Network Protocols
UDP	User Datagram Protocol
XDR	eXternal Data Representation Standard as specified by Sun Microsystems Open Network Computing

Appendix A
IR Target Source Files

'C' Source File Name	Description
acse.c	The Association Control Service Element functions take care of association and disconnection between the client and IR kernel i.e. A_associate(), A_abort etc.
csconn.c	This function connects to target process and obtains a TCP socket file handle based on the passed parameters: host name and TCP port number.
cspabort.c	This function closes the client and target process ports and removes a selected association from the association list by calling A_abort.
cspconn.c	This function accepts a TCP connection to the target process and returns the new socket file handle.
cspdatai.c	This function decodes an APDU.
cspdatar.c	This function encodes an APDU.
cspliset.c	This function sets up the TCP/IP service socket.
cspoll.c	This function polls the TCP ports: 1. New client (default) port. 2. An association client port. 3. An association target process port. 4. Repeat step 1 to 3 for the next association. And maps the ASN.1 event code to the IR kernel internal event code.
csrecv.c	This function receives an APDU with a selected socket file handle.
cssend.c	This function sends an APDU with a selected socket file handle.

'C' Source File Name	Description
encdec.c	This file contains the SNACC ASN.1 functions for this particular ASN.1 data definition.
iraccctr.c	This file contains access control functions to process the request PDU and the response PDU.
iraliset.c	This function initializes the Application Layer Interface.
irassoc.c	This file contains the functions to maintain the global client association information list. It includes the functions to add, to delete, to view, to find an association, to create and to free the association list.
ircomevt.c	This function polls all associations and checks for responses and idle timeouts.
irdelete.c	This file contains the delete result set functions to process the result set delete request PDU and the result set delete response PDU.
irerror.c	This function displays and logs the errors to screen and an error log file.
irfwrite.c	This function is a replacement of the C function fwrite(). It displays the Hex numeric value for non-printable characters.
irg2apdu.c	This function moves the second APDU to the first APDU position and returns an APDU type (such as init, search).
irinit.c	This file contains the initialization functions to process the init request PDU and the init response PDU.
irpresnt.c	This file contains the present result set functions to process the present request PDU and the present response PDU.
irrsctrl.c	This file contains the resource control functions to process the resource control request PDU and the resource control response PDU.

'C' Source File Name	Description
irrsctrp.c	This file contains the resource control report functions to process the resource report request PDU and the resource report response PDU.
irrwatch.c	This file contains the functions to activate, deactivate and monitor the timeout for the response event and idle association.
irsearch.c	This file contains the search result set functions to process the search request PDU and the search response PDU.
irserreq.c	This function sends the service request APDU to the target process.
irserrsp.c	This function sends the service response APDU to the target process.
irsndreq.c	This function sends the target process request APDU to the client system.
irsndrsp.c	This function sends the target process response APDU to the client system.
irspawnp.c	This file contains functions to start the init target process and to connect between the IR kernel and the target process.
irstkser.c	This file contains the stack functions to track the response timeouts.
irstksta.c	This file contains the stack functions to track the program states during access control and resource control transition.
irtarget.c	This file contains the IR kernel main program which implements the IR Target Kernel OSI ANSI Z39.50 Protocol. The IR kernel consists of state transition function, control function (communication dispatcher), ACSE (Association Control Service Elements), and IR ASE (Application Service Element).
irterm.c	This file contains the termination functions.

'C' Source File Name	Description
irtgrsct.c	This file contains the trigger resource control functions to process the trigger resource control request PDU.
tpstartc.c	Source code generated by the RPCGEN utility for server tp_startup_1() for init request.
tpstarts.c	Source code generated by the RPCGEN utility for the client main() for init request.
tpstartu.c	Source code generated by the RPCGEN utility for tp_startup_1() for init request. It is a user modified function which executes the target process black box program 'tpbbox' and then reads the TCP port number via the name pipe from the program 'tpbbox'. Finally, it returns the TCP port number to the IR kernel init program.
tpstartx.c	Source code generated by the RPCGEN utility to support XDR, i.e. function xdr_port_no_struct() for init request.

Appendix B
IR Target Header Files

Header File Name	Description
acse.h	Header file for acse.c;
apdu.h	SNACC ASN.1 definition header file;
csconn.h	Header file for csconn.c;
cspabort.h	Header file for cspabort.c;
cspconn.h	Header file for cspconn.c;
cspdatai.h	Header file for cspdatai.c;
cspdatar.h	Header file for cspdatar.c;
cspliset.h	Header file for cspliset.c;
cspoll.h	Header file for cspoll.c;
csrecv.h	Header file for csrecv.c;
iraccctr.h	Header file for iraccctr.c;
iraliset.h	Header file for iraliset.c;
irassoc.h	Header file for irassoc.c;
ircomevt.h	Header file for ircomevt.c;
irconst.h	Constant definition header file for all the IR programs;
irdelete.h	Header file for irdelete.c;
irerror.h	Header file for irerror.c;
irinit.h	Header file for irinit.c;
irpresnt.h	Header file for irpresnt.c;
irrsctrl.h	Header file for irrsctrl.c;
irrsctrp.h	Header file for irrsctrp.c;

Header File Name	Description
irrwatch.h	Header file for irrwatch.c;
irsearch.h	Header file for irsearch.c;
irserreq.h	Header file for irserreq.c;
irserrsp.h	Header file for irserrsp.c;
irsndreq.h	Header file for irsndreq.c;
irsndrsp.h	Header file for irsndrsp.c;
irstatab.h	Header file for irstatab.c;
irstkser.h	Header file for irstkser.c;
irstksta.h	Header file for irstksta.c;
irterm.h	Header file for irterm.c;
irtgrsct.h	Header file for irtgrsct.c;
ir.h	Header file for all IR programs;
ir_oid.h	Object Identifer definition header file;
tpstartu.h	RPC header file for tpstartc.c, tpstarts.c, tpstartx.c and irspawnp.c;

Appendix C
Test Software (Black Box) Source Files

'C' Source File Name	Description
tpaccctr.c	This file contains the functions - to create an access control request PDU, - to encode it to an APDU, - to send an APDU to the IR kernel, and - to receive and verify the access control response from the IR kernel.
tpbbox.c	This file is the main program for the target process blackbox. It controls the traffic flow between the target process and IR kernel.
tpdelete.c	This file contains the following functions: - checks the delete request PDU, - performs the security checks, - performs the resource verification, - creates and sends the delete response APDU to the IR kernel.
tpgetpdu.c	This function receives an APDU from a TCP data stream and then decodes it to an IR PDU.
tpinit.c	This file contains the following functions: - checks the init request PDU, - performs the security checks, - performs the resource verification, - creates and sends the init response APDU to the IR kernel.

'C' Source File Name	Description
tppresnt.c	This file contains the following functions: - checks the present request PDU, - performs the security checks, - performs the resource verification, - creates and sends the present response APDU to the IR kernel.
tprsctrl.c	This file contains the functions - to create an resource control request PDU, - to encode it to APDU, - to send the APDU to the IR kernel, and - to receive and verify the resource control response from the IR kernel.
tprsrpt.c	This file contains the following functions: - checks the resource report request PDU, - performs the security checks, - performs the resource verification, - creates and sends the resource report response APDU to the IR kernel.
tpsearch.c	This file contains the following functions: - checks the search request PDU, - performs the security checks, - performs the resource verification, - creates and sends the search response APDU to the IR kernel.

’C’ Source File Name	Description
tptgrsct.c	This file contains the functions to check the trigger resource control request.

In addition to the above ’C’ source, unique to the black box software, the IR kernel programs **encdec.c, irerror.c** and **irfwrite.c** are required also. A CanMARC data file **canmarc** is included for filling the search and present response database records.

Appendix D
Test Software Header Files

The following list contains the IR target process 'C' header files.

Header File Name	Description
tp.h	Header file for all the target process 'C' programs.
tpbbox.h	Header file for all the target process 'C' programs.
tpconst.h	Constant header file for all the target process 'C' programs.

Appendix E
SNACC ASN.1 Header Files

The following list itemizes the SNACC ASN.1 header files.

asn_any.h	**asn_bits.h**	**asn_bool.h**
asn_conf.h	**asn_enum.h**	**asn_incl.h**
asn_int.h	**asn_len.h**	**asn_list.h**
asn_null.h	**asn_octs.h**	**asn_oid.h**
asn_real.h	**asn_tag.h**	**asn_usef.h**

exp_buf.h	- included in asn_conf.h
hash.h	- included in asn_any.h
sbuf.h	- included in asn_conf.h
min_buf.h	- included in asn_conf.h
nibble_alloc.h	- included in asn_config.h
print.h	- included in asn_conf.h

The SNACC ASN.1 library files are

- **libasn1csbuf.a**
- **libasn1cebuf.a** and
- **libasn1cmbuf.a**.

These files are necessary to compile and produce the executable irtarget, tpstartu, and tpbbox provided your environment matches the following criteria: UNIX AT&T version 5.3.2 or equivalent, and GNU C compiler version 2.2.2 or equivalent. If you have to rebuild the SNACC ASN.1 library files, you should retrieve the SNACC ASN.1 compiler, refer to Section 4.1.3 for information on how to obtain the ASN.1 compiler.

Appendix F
GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

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

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

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

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

   Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

   You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code.(This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

   If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

   It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

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

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

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND

PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

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

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

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

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

<one line to give the program's name and a brief idea of what it does.> Copyright (C) 19yy<name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.
This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

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

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) 19yy name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type 'show w'. This is free software, and you are welcome to redistribute it under certain conditions; type 'show c' for details.

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than 'show w' and 'show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

> Yoyodyne, Inc., hereby disclaims all copyright interest in the program 'Gnomovision' (which makes passes at compilers) written by James Hacker.
>
> <signature of Ty Coon>, 1 April 1989
> Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.