Interactive COBOL

Utilities

Manual

ICOBOL Revision 4.70

No. 011-00300-20

08/16/2012

LICENSE AGREEMENT

Carefully read the following terms and conditions. **Use of this product constitutes your acceptance of these terms and conditions and your agreement to abide by them.**

You, the purchaser, are granted a non-exclusive license to use this software under the terms stated in this agreement. The program and its documentation are copyrighted and may not be copied or reproduced in any part, in any form, for any purpose, except according to the terms stated in this agreement.

You may:

1. use the software for up to the number of active users for which the software was purchased.

2. use the software provided a valid license is installed for the required number of active users to be supported at any one time.

3. copy the software into any machine readable form for backup purposes.

4. transfer the software from one computer to another.

5. assign or transfer the software and license to another party if the other party agrees to all the terms and conditions of this agreement. Once the transfer is complete you must destroy any copies of the software not transferred.

6. rent, sublicense, or lease the software and license if the user agrees to all the terms and conditions of this agreement.

7. not alter, modify, or adapt the software itself, including, but not limited to, translating, decompiling, or disassembling.

8. copy or reproduce the documentation for purposes of using a valid license.

This license and your right to use the software automatically terminate if you fail to comply with any provision of this License Agreement. You agree upon such termination to destroy the software and license.

LIMITED WARRANTY

Envyr Corporation warrants that (a) the software will perform substantially in accordance with the accompanying written materials for a period of ninety (90) days from the date of receipt; and (b) any hardware accompanying the software will be free from defects in materials and construction under normal use and service for a period of one (1) year from the date of receipt. Any implied warranties on the software and hardware are limited to ninety (90) days and one (1) year respectively. Some states do not allow limitations on duration of an implied warranty, so the above limitation may not apply to you.

Envyr Corporation's entire liability and your exclusive remedy shall be, at Envyr Corporation's option, either (a) return the license fee or (b) repair or replacement of the software or hardware that does not meet the above Limited Warranty and which is returned to the original vendor with a copy of the receipt. This Limited Warranty is void if failure of the software or hardware has resulted from accident, abuse, or misapplication.

In no event shall Envyr Corporation or its suppliers be liable for any damages at all, including, but without limitation, damages for loss of business profits, business interruption, loss of business information, or other pecuniary loss, arising out of the use of or inability to use this software or hardware, even if Envyr Corporation has been advised of the possibility of such damages.

NOTICE

This manual has been prepared for use only with the Interactive COBOL product by prospective customers or valid licensees. The information in this manual is preliminary and subject to change without prior notice.

In no event shall the seller be liable for any incidental, indirect, special or consequential damages at all (including but not limited to lost profits) arising out of or related to this document or the information contained in it even if the writers have been advised, knew or should have known of the possibility of such damage.

Revision History:
Release	2.00	- March 1994
Release	2.20	- August 1996
Release	2.40	- June 1998
Release	3.00	- August 2000
Release	3.10	- April 2001
Release	3.20	- April 2002
Release	3.30	- February 2003
Release	3.40	- March 2004
Release	3.64	- January 2008
Release	4.00	- October 2008
Release	4.01	- December 2008
Release	4.04	- February 2009
Release	4.07	- June 2009
Release	4.10	- August 2009
Release	4.21	- December 2009
Release	4.40	- June 2010
Release	4.50	- April 2011
Release	4.53	- July 2011
Release	4.70	- August 2012

Effective with:
   Interactive COBOL Revision 4.70

TRADEMARKS

ICHOST is a trademark of Envyr Corporation.

AOS/VS, RDOS, and DG/UX are trademarks of Data General Corporation.
IBM and PC-DOS are registered trademarks of International Business Machines Corporation.
Pentium and Pentium-Pro are trademarks of Intel Corporation.
Intel is a registered trademark of Intel Corporation.
AIX, PC, RISC System 6000, 3101, 3151, and 3161 are trademarks of International Business Machines Corporation.
Microsoft, MS-DOS, and XENIX are registered trademarks of Microsoft Corporation.
Sun and Interactive UNIX are registered trademarks of Sun Microsystems, Inc.
U/FOS is a trademark of Transoft, Ltd.
UNIX is a registered trademark of Unix System Laboratories, Inc. (USL).

All other product names mentioned herein are trademarks of their respective owners.

TABLE OF CONTENTS

<div align="center">**APPENDICES**</div>

LIST OF SCREENS

LIST OF TABLES

PREFACE

Scope

This manual provides the information needed to use the utilities provided with the Interactive COBOL Runtime product on UNIX® and Windows®.

The complete documentation for Interactive COBOL includes the following manuals:

Installing and Configuring Interactive COBOL on UNIX® (011-00402)
Installing and Configuring Interactive COBOL on Windows® (011-00403)
    Each manual provides the appropriate sections necessary to properly install and configure Interactive COBOL in the given environment.

Interactive COBOL Utilities Manual (011-00300)
    Provides a simple guide to all the user visible utilities.

Interactive COBOL Language Reference & Developer's Guide (011-00100)
    The complete COBOL syntax supported by Interactive COBOL. Provides how to use the development tools including the compiler and IDE. It also explains how the COBOL runtime works including its internal system calls, builtins, and how to program across the multiple environments supported by Interactive COBOL.

 sp2 panel Editor
    Provides how to develop and use the sp2 User Interface Development System.

COBOL FormPrint
    How to use the FormPrinter Editor to setup printers.

Terms

This document will use several terms which it will define as generic names to describe several individual products.

Windows will be used to collectively refer to all supported versions of Windows which includes Windows 2000, Windows XP, Windows Server 2003, Windows Server 2008, and Windows Vista versions.

ICOBOL refers to all models of Interactive COBOL unless otherwise stated.

ICOBOL/Windows refers to **ICOBOL** on either Windows 2000, Windows XP, Windows Server 2003, Windows Server 2008, Windows Vista, Windows 7 unless otherwise stated.

PC refers to any of the various Intel 80386, 80486, Pentium™, Pentium-Pro™, etc. -based microcomputers that are compatible to the IBM AT™ line of products.

UNIX refers to all supported flavors of UNIX including Linux unless specifically stated.

DG refers to Data General Corporation.

ENHANCEMENTS (Utilites)

Interactive COBOL 4.70 added support for the following:

- New environment variable ICCONFIGDIR to allow for customized system files

Interactive COBOL 4.50 added support for the following:

- ICNETD ISQL Server support added

Interactive COBOL 4.10 added support for the following:

- Ability to generate .PDF files in Printer Control

Interactive COBOL 4.00 added support for the following:

- Enhanced logging facilities supported by ICLOG and the runtimes.

- Default messages are preloaded into executables.

- Ability to run **ICOBOL** 2 .cx files

Interactive COBOL 3.56 added support for the following:

- The characters "(" and ")" are now valid filename characters.

Interactive COBOL 3.50 added support for the following:

- New utility ICWEBMSG

Interactive COBOL 3.40 added support for the following:

- ICSORT updated with scripting capabilities and translation between uppercase/lowercase and ASCII/EBCDIC

- ICSMVIEW can display more information for each type of connection

Interactive COBOL 3.36 added support for the following:

- ICSMVIEW enhancements

Interactive COBOL 3.35 added support for the following:

- Terminal Control screen updated to use number of lines of the terminal to control display size

- ICSMVIEW enhancements

- ICINFO Unix enhancements

Interactive COBOL 3.34 added support for the following:

- ICSMVIEW enhancements

- ICINFO (UNIX) enhancements

Interactive COBOL 3.33 added support for the following:

- tell_sem (UNIX) enhancements

Interactive COBOL 3.32 added support for the following:

- ICINFO (UNIX) enhancements

Interactive COBOL 3.30 added support for the following:

- ICNETD ThinClient (char) support added

Interactive COBOL 3.22 added support for the following:

- ICCHECK shows the record number for any most record problems and can check read-only files

- ICFIXUP shows if any unreliable flags have been set and shows the record position for many record problems

- ICREORG shows the input record number for any duplicate key record error

- ICSTAT shows the Deleted record count even when not scanning the file for version 6 and 7 files

Interactive COBOL 3.20 added support for the following:

- Enhanced auditing support added

Interactive COBOL 3.13 added support for the following:

- ICNETD i/o server now called icios

- ICNETD (UNIX) ThinClient (gui) support added

- ICNETD (Windows) Domain support added

- ICNETD (Windows) new switch (-N b) for no logon as batch

Interactive COBOL 3.10 added support for the following:

- ICNETD (Windows) added ThinClient (gui) support for sp2 and FormPrint runtimes.

Interactive COBOL 3.02 added support for the following:

- ICREORG allows the output argument to be optional and defaults to standard out and line sequential.

I. INTRODUCTION

A. Overview

This manual describes the various utilities (except ICCONFIG/ICEDCFW and ICEXEC) that are provided with the Interactive COBOL Runtime System.

Two sets of utilities will be discussed in this manual.

The first set of utilities are standard executables and can be run from any shell or command-processor. This set is composed of ICCHECK, ICCREATE, ICFIXUP, ICINFO, ICLIB, ICLINK, ICLOG, ICMAKEMS, ICNETD, ICPACK, ICPQUTIL, ICREORG, ICREV, ICREVW, ICSMVIEW, ICSORT, ICSTAT, and ICWHOHAS and will be referred to as command-line utilities. These are discussed starting on page 27.

The second set of utilities are available from within a COBOL program using the appropriate mechanism as documented in the Interactive COBOL Language Reference & Developer's Guide. This set is composed of Abort Terminal, Kill Terminal, Message Sending, Printer Control, System Information, Terminal Control, and Terminal Status. These are discussed starting on page 123.

B. How to Read this Manual

This manual should be used as a general reference for the Interactive COBOL utilities. As such, after reading this first chapter, the individual chapters that describe each utility should be read as needed using the Table of Contents or the page headers to find the specific utility.

C. Operating Environment

C.1.   General Concepts

The Interactive COBOL system has been designed to provide an application operating environment that works as consistently as possible among several different operating system environments. This consistency is expressed in a few key concepts that have their roots in the UNIX operating system. If you have used UNIX, the concepts may already be familiar to you.

The first concept is that programs communicate with their operating environment through three input/output streams or files: standard input (stdin), standard output (stdout), and standard error (stderr). Programs can read data to be processed from stdin, process it in some way, and write the results to stdout. Programs report errors to stderr. By default, most systems connect stdin to the console keyboard and both stdout and stderr to the console display.

Many utilities, especially in the COBOL environment, must process complex data files that do not fit this simple model and so they do not often use stdin for the data to process. However, the stdout and stderr files are still very useful. They allow the utility to logically separate error reporting from reporting the results of processing. For example, the ICSTAT utility reports statistics about ICISAM files. It reports these statistics to stdout. If an error occurs, for example one of the command arguments does not exist, the error is reported to stderr.

The second concept is the ability to redirect i/o files from the default files to another file or device. The MS-DOS and UNIX systems provide a very simple way to redirect these standard files in the command processor by using the special characters `<' and `>'. When stdout is redirected to a file, it provides a simple mechanism to capture the output of a utility. See your operating system command processor documentation for more on this concept.

The third major concept is the ability to customize the operation of specific programs by setting information in items called Environment Variables. Environment variables have a name and a value like program variables or data items. The difference is that these variables are managed by the command processor. The utility programs can ask the operating system whether a particular environment variable is set or not, and what its value is. They are most often

17

used to set default operating options, or the locations of important files. For example, all Interactive COBOL command-line programs look for the environment variable ICROOT as the base directory for finding certain system files. ICCONFIRDIR is also used to look for customized system files. They also look for command-line options in an environment variable by their own name. UNIX and Windows both provide environment variables.

All the command-line utilities support their own environment variable as "upper-case(utility-name)".

Environment variables are maintained in the command processor (or shell). Environment variables are set as follows:

```
   ICROOT=/usr/cobol470                          (On UNIX)

   SET ICROOT=C:\program files\icobol            (On Windows)
```

C.2.   Directory Structure

On UNIX, the Interactive COBOL software is installed in a directory with the revision number in the name by default. For example, Interactive COBOL Revision 4.61 will be in a directory named "cobol461". This directory can be installed wherever is most appropriate or convenient for your system. On Windows, Interactive COBOL will be installed in a directory named "icobol" in the "program files" directory by default. You will normally want to include this directory in your PATH.

The main directory contains all of the command-line programs, the readme file(s), and supplied COBOL executable programs. Additional subdirectories are provided as noted below. The help subdirectory contains help (.hf) files for all the command-line programs defined as <command>.hf. Interactive programs may have their own subdirectories under the help directory. Descriptions that start with (Dev) are part of the development system.

```
        Main
        Directory       Sub-
        Directory       Directories     Description
        cobol<rev>                      Main executables and needed files

                        cgicobol        Cgiruntime, scripts, examples

                        examples        Various examples

                        help            Help files (.hf)

                        link_kit        (Dev) Interface to install user-written
                                        subroutines into runtime

                        messages        Sample English Message files

                        tcs             ThinClient server files

                        user_lib        (Dev) Interface to ICISAM files (ICAPI)for
                                        non-COBOL use
```

ICOBOL Directory Structure (UNIX)

Main Directory	Sub-Directories	Description
icobol		Main executables and needed files
	cgicobol	Cgiruntime, scripts, examples
	examples	Various examples
	help	Help files (.hf)
	link_kit	(Dev) Interface to install user-written subroutines into runtime
	messages	Sample English Message files
	qpr	(Dev) Gui-printer development (Formprint)(ICQPRW)
	sentinel	Rainbow sentinel device files
	sp2	(Dev) Gui-screen development (ICSP2)
	tcs	ThinClient server files
	user_lib	(Dev) Interface to ICISAM files (ICAPI)for non-COBOL use

ICOBOL Directory Structure (Windows)

Installs previous to 4.70 had a print sub-directory for printer translation (.pti) files and background .pdf files and a term sub-directory for terminal description (.tdi) files. The default versions of .pti and .tdi files are now builtin to the runtimes and any customized file(s) should be stored in a directory that is sought with the ICCONFIGDIR environment entry.

Command-line programs require the corresponding help file to be available in order to display their help text. If it is not available, an error message will be displayed that it could not find the help file. There are three methods for finding the help file: using the ICCONFIGDIR environment variable, using the ICROOT environment variable or by passing a partial pathname to the operating system.

C.3.   ICEXEC

On UNIX and Windows, the Interactive COBOL system uses a control program called ICEXEC to coordinate multi-user access to system resources. The runtimes require the ICEXEC program to be running in order to operate. However, the utility programs can operate with or without ICEXEC. Note that UNIX does not provide an exclusive open capability and so this is provided by ICEXEC when it is running.

On UNIX, when ICEXEC is NOT running, an exclusive open output is emulated by posting a write-lock on the whole file, and an exclusive open input is emulated by posting a read-lock on the whole file. All non-exclusive opens post a read-lock on byte 1 of the file. Thus an exclusive open output can detect if anyone else has the file open and all other opens can detect if an exclusive open output has the file open by using this method. Care should be used when starting ICEXEC if any programs using this method are running since new invocations of programs will use the ICEXEC-controlled scheme for sharing files which does not use operating system locks. More on ICEXEC can be found in the respective Installing manuals.

The following executables **require** the shared area that ICEXEC manages:

icrun	icios	(I/O server started by icnetd)
icruncgi	icrunrs	(ThinClient server started by icnetd)

icsmview   iclogs   (Logging server started by icnetd)
icwhohas

All other Interactive COBOL executables can operate with or without ICEXEC.

D. **Conventions**

Another aspect of providing a consistent system across multiple operating platforms, is in the command-line interface. The command-line programs use a common command-line syntax across all platforms, and they adhere to the following standard conventions:

> 1) all switches are composed of a single letter or digit preceded by a hyphen (-) (or optionally a forwardslash (/) on Windows);
> 2) the switches are order independent;
> 3) the switches ARE case sensitive;
> 4) lower-case switches imply an action or modification of an action, e.g., `-h' for help;
> 5) UPPER-CASE switches imply an action with a required argument that must follow with an intervening space, e.g., `-A audit.log' for setting up an auditfile called audit.log.
> 6) multiple lower-case switches can be combined with one hyphen, e.g., `-axz' for `-a -x -z'.

The following shows how the various conventions for defining syntax will be represented in the Interactive COBOL documentation:

[ ]   Brackets enclose optional portions of a format. One of the options contained within the brackets may be explicitly specified or that portion may be omitted.

{ }   Braces enclosing a portion of a format means that one of the options contained within the braces must be specified.

| Bar will be used to separate choices when multiple choices are allowed.

...   Ellipsis indicates that the previous item can be repeated one or more times.

italic-lower-case Indicates a generic term representing a value that is defined as indicated.

UNIX systems support case-sensitive filenames as opposed to Windows systems that are case-insensitive. All released Interactive COBOL on UNIX files are lower-case which is in keeping with most UNIX systems. By default, the Interactive COBOL on UNIX runtime will convert all COBOL filenames, including program names, to lower-case before looking up that file in UNIX. Although Interactive COBOL on UNIX can support UPPER-CASE only or mixed-case, we recommend using only one case for filenames to ease portability to case-insensitive environments.

With this in mind, this document will still use upper-case names in the text for specific programs but will always use lower-case in examples and when showing what needs to be entered from the keyboard to run a program.

On UNIX, all examples assume the Bourne shell is being run.

E. **Common Switches**

E.1.   Overall

There are several common switches that appear on all command-line programs except for ICINFO. These are described in detail in the following sections so that the descriptions for each program can be abbreviated. The command-line switch processor scans all the command-line switches, checking for errors. Any errors display an

abbreviated startup banner (the program name and revision) to stdout before displaying the error message to stderr and then exiting with a non-zero exit code. If there are no errors to terminate processing prematurely, the common switches are processed. First, if the Help switch is given, an abbreviated startup banner and help text are displayed to stdout after which the program exits normally (i.e., no other switches or arguments are processed). Next, if the Audit switch is given, auditing is enabled. Finally, the Quiet switch, if given, is processed. The program then begins its specific processing by emitting a startup banner, consisting of the program name, revision level, system, and the copyright notice. When it finishes processing, it will emit a trailer message indicating that it is done.

E.2.   Audit Switch

The Audit switch will be shown in the syntax as:

   -a[:*aflag*] | -A *file*|*dir*[:*aflag*]

Where *aflag* is a|b|d|p|t|u|da|db|pa|pb|ta|tb|ua|ub and modifies the behavior of the audit file selection.

a	Append	If the file exists, do Not truncate the file, just append. The Append flag can be used alone or with the Date, Pid, Time, or Username options.
b	Backup	If a previous log file (.lg) exists, rename it to *.lgb and then open a new .lg file. The Backup flag can be used alone or with the Date, Pid, Time, or Username options. <u>On UNIX</u>, this will break hard links.
d	Date	Add date in the form of _*YYYYMMDD* before the .lg extension.
p	PID	Add pid in the form of _*NNNN* before the .lg extension.
t	Time	Add time in the form of _*YYYYMMDDHHmmsshh* before the .lg extension. (YYYY-year, MM-month, DD-day of the month, HH-hour, mm-minute, ss-second, hh-hundredths of seconds.)
u	Username	Add username in the form _*name* before the .lg extension.

NOTE:
1) <u>On Windows</u>, the option "-A c:a" will be treated as open file "c" in append mode in the current directory. Previously this would have been open file "a" in the current directory of drive C:. To get the old behavior, enter

   "-A c:.\a"

The audit flags (a,b,d,p,t,u) instruct the Audit processing to take a different action then the default for the audit file. The default action is the same as usual, truncate the file to zero on startup.

Note that:

-a	Audit to the default file for this command.
-A *file*	Audit to the specified file.
-A *dir*	Audit to default file in the specified directory.

Audit files contain a copy of any output that was sent to either stdout or stderr, in the same order as it was emitted at execution time (i.e., it may be interspersed). The programs handle this internally, so stdout and stderr can still be redirected. The audit file can be specified to use the default name in the current directory (-a), a user specified name (-A *file*), or the default name in a specified directory (-A *dir*). An audit file is always created if it does not already exist. If it does exist, it is truncated to zero unless the Backup flag is set.

The default audit file name is <command>.lg.

E.3.   Quiet Switch

The Quiet switch will be shown in the syntax as:

   -q

The Quiet switch works by suppressing all output that is emitted to stdout. The most obvious effect is that it suppresses the usual banner and trailer messages that are emitted to stdout as the program starts and terminates. Because it is suppressing stdout, the Quiet switch may also suppress other parts of the usual output.

E.4.   Help Switch

The Help switch will be shown in the syntax as:

   -h|-?

The Help switch displays a summary of the command-line syntax, the switches and what they do, and the applicable environment variables.

F. **Filename Extensions**

Interactive COBOL requires that the extension for certain specific types of files to match those given in the following table except for those marked *defacto*. Those marked *defacto* are only the most common extensions used for these purposes and not required. All Interactive COBOL release files will conform to these *defacto* standards.

Those extensions marked as this sentence is marked are extensions in some older revision of Interactive COBOL or ICHOST but are handled in some special cases by current Interactive COBOL utilities.

Common extensions used by Interactive COBOL include:

.CD	old ICHOST COBOL program file
.CF	old Configuration file (pre-3.30)
.CFI	Configuration file (.ini format)
.CL	Library file
.CO	COBOL Source programs (card format) *(defacto)*
.COB	AOS/VS COBOL source text file
.CX	COBOL Program file
.ER	Error file *(defacto)*
.FA	File attribute file
.FP	Failsafe protection file
.HF	Interactive COBOL help file
.LG	Audit / Log file *(defacto)*
.LGB	Backup Audit / Log file (*defacto*)
.LK	Link file
.LS	List file *(defacto)*
.MS	Message file
.OD,.NT	Pair of files, ICPACK data and index temporary files
.PD,.DD	Pair of files, older revision COBOL program file (program and data)
.PQ	Printer control file
.PT	old Printer translation file (pre-3.30)
.PTI	Printer translation file (.ini format)
.RP	Remote protection file (MS-DOS only)
.SD	ICRUN Sort data file (temporary)
.SR	COBOL Source program (text format) *(defacto)*

.ST	ICRUN Sort tag file (temporary)
.SY	COBOL Symbol table file
.TD	old Terminal description file (pre-3.30)
.TDI	Terminal description file (.ini format)
.TMP	Temporary file *(defacto)*
.UDB	U/FOS database
.XD,.NX	Pair of files, COBOL ICISAM file (data and index portion)
.XDB	ODBC database definition file (.ini format)
.XDT	ODBC table definition file (.ini format)
.XL	Log file
.XLG	Generation log file (pre-4.00)

On UNIX, all Interactive COBOL utilities support mixed-case filenames. If a utility needs to add an extension, e.g., .PD/.DD, .XD/.NX, etc., it searches back from the end of the simple filename for the first alphabetic character. If it finds an upper-case alphabetic, it will use an upper-case extension, otherwise a lower-case extension is used. For example, "iccheck DATAbase1" and "iccheck 12345" would use the lower-case extensions `.xd' and `.nx' for the ICISAM file, while "iccheck dataBASE52" would use the upper-case extensions `.XD' and `.NX'.

G. **Exit Codes**

All command-line programs return exit codes that provide an indication of the success or failure of the program. These are returned through the appropriate OS-specific mechanism (e.g., ERRORLEVEL on Windows and the exit code on UNIX). In general, the following codes will be returned:

0	The program completed without errors.
1	The program ran, but some items it processed had errors.
2	The program was running, but was terminated by an operator interrupt or external abort.
3	The program was running, but was terminated by some fatal internal error. For example, the compiler was running but detected that its virtual memory manager had run out of memory unexpectedly.
4	There were command-line errors and so the program did not perform any of the requested function(s).
5	The user was not authorized to execute the program or perform a requested operation, so the program did not run.
6	The program experienced an error during its initialization phrase and could not execute. For example, it could not allocate sufficient memory to perform its function.
7	Help was requested.
8-9	Reserved for future `common' errors.
10-	These codes are specific to each program and will be documented with each program.

All of the programs support exit codes 0 through 9 with the meaning described above.

H. **Common Environment Variables**

H.1.   Overall

There are several common environment entries that most command-line programs use. These are described in detail in the following sections so as to not be duplicated under all program descriptions. Other environment variables that are more program specific will be described under each program.

All command-line utilities support an environment variable that is their own name, i.e., the environment variable ICCHECK is read only by iccheck, ICSTAT by icstat, etc.

H.2.   ICROOT

ICROOT specifies the Interactive COBOL root directory. ICROOT is used to find ceratin system directories like help, print, and term.

The syntax is:

    ICROOT=*dir*

Where
    dir

        Specifies the directory where to find the Interactive COBOL help, print, and term directories. Usually this should be set the current revision directory.

On UNIX, if ICROOT is not set, the current directory is used. On Windows, if ICROOT is not set, then the registry is queried for the initial installation directory.

H.3.   ICCONFIGDIR   (Added in 4.61)

ICCONFIGDIR specifies a directory for customized system files. ICCONFIGDIR is used to find customized system files like help, messages, print, and term entries that have been customized by the user.

The syntax is:

    ICCONFIGDIR=*dir*

Where
    dir
        Specifies the directory where to find the customized system files (help, messages, print, and term).

If set, a particular help, message, term, or print file will first be sought using ICCONFIGDIR. If the particular file is not found, then ICROOT will be used.

If ICCONFIGDIR is not set, then ICROOT is used.

H.4.   ICTMPDIR

ICTMPDIR specifies a directory to which programs may write any temporary files.

The syntax is:

    ICTMPDIR=*dir*

Where
    dir
        Specifies a valid pathname for the directory in which any needed temporary files are to be written.

If ICTMPDIR is not set, the current directory is used.

Some of the programs that look for the ICTMPDIR environment variable are ICOBOL, ICLIB, ICRUN, ICSORT, and ICREV.

H.5.   Executable Name

All command-line utilities support an environment variable of the same name as the utility in upper-case. For example, 'icccheck' will recognize the variable ICCHECK. The environment variable can contain command line options for the utility which will be processed prior to any options actually present on the command line. If such an environment variable is present, the utility will display the complete set of options at startup.

H.6.   TZ (On Windows)

TZ specifies the time zone and number of hours past Greenwich mean time (GMT) for this location.

The syntax is:

```
TZ=tttn[ttt]
```

Where
ttt
        Specifies a time zone of three letters. The second time zone should be given if daylight-saving time applies at this location.
n
        Specifies a positive (west) or negative (east) integer number of hours difference from Greenwich mean time (GMT). Up to two digits can be specified.

If no TZ is specified, Interactive COBOL assumes all times are Greenwich mean time (GMT). If the second time zone is specified, Interactive COBOL assumes that daylight-saving time starts and stops based on the same schedule as used in the USA.

An example for Raleigh, North Carolina, USA would be:

```
SET TZ=EST5EDT
```

TZ is used on Windows for command-line programs to accurately report date and time, and to accurately set date and time information in file headers. It sets the time zone and number of hours past Greenwich mean time (GMT) for this location.

PART 1 - EXECUTABLES

II. ICCHECK

A. Introduction

The ICCHECK utility provides the ability to verify the integrity of ICISAM indexed and relative files. This utility should be run after all abnormal system terminations in the directories that contain ICISAM files. ICCHECK exclusively opens each ICISAM file, preventing any other process from using the files while ICCHECK is scanning them.

B. Syntax

The standard syntax is:

```
iccheck [-a[:aflag]|-A file|dir[:aflag]] [-e|-E file|dir] [-h|-?] [-i]
    [-N {s|w}...] [-p] [-q] [-s] [-u] [argument]...
```

Where
   -a[:*aflag*]|-A *file|dir[:aflag]* (Audit)
      Enables auditing (default iccheck.lg). Where *aflag* is a|b|d|p|t|u|da|db|pa|pb|ta|tb|ua|ub, defined as a-append, b-backup, d-date, p-pid, t-time, and u-username.
   -e|-E *file|dir* (Error)
      If an ICISAM file being processed has errors, this switch directs ICCHECK to create an error file or append to an error file that already exists. When the -e or -E *dir* switch is selected, the name of the error file is formed by adding a `.er' extension to the name of the ICISAM file being checked, and for -E *dir*, locating that file in the specified directory. The -E *file* switch creates a single error file with the specified name in which errors are recorded.
   -h|-? (Help)
      Displays help text.
   -i (Info)
      Displays information messages.
   -N {s|w}... (No options)
      Specifies a NO option. Valid NO options are:
         s (No-scan)   do not Scan the remainder of the file if the ICISAM reliability flags are not set and the .XD/.NX headers are valid. Use of the No-scan switch allows ICCHECK of many files to finish faster if the files are good.
         w (No-warning)   do not generate Warning messages.
   -p (Progress)
      A progress report is displayed as a key path is scanned. Progress is reported every 1%. This switch will substantially slow the processing of smaller files.
   -q (Quiet)
      Enables quiet operation.
   -s (Sync Reporting)
      Only report out-of-sync logging
   -u (Unreliable)
      Simply report the names of the files that do not pass the reliability tests (i.e. have their reliability bits set or have invalid file headers). No other checking of the integrity of any of the files will occur, so this option cannot be specified with the No-scan (-N s) or Error switches.
   argument
      Specifies any filename or template to be checked. If no argument is specified, all ICISAM files in the current directory are processed.

Environment variables:

   ICCHECK   Command line options

C. **Description**

If neither the No-scan switch (-N s) or Unreliable switch (-u) are given, full checking will be performed on the given files.

The reliability flags in an ICISAM file are set any time the integrity of the file may be compromised. They are set by the ICISAM system whenever a modification of the file is made and are normally cleared when the file is closed or otherwise updated to disk.

ICCHECK will set the ICISAM reliability flags if it detects any errors in the logical or physical structure of the file. If ICCHECK detects no errors and the reliability flags are set, ICCHECK will clear the flags, thus making the file accessible to Interactive COBOL processes without getting a File Status 9F.

ICCHECK first checks the .XD/.NX headers to make sure the headers contain valid information. Next, the header information is used to check that the files are consistent with the header information. If there are any errors at this point that ICCHECK can fix, it will correct the problem(s) to make the file consistent. Finally, unless the No-scan switch (-N s) is set, the index structure (.NX file) and sequential record path (.XD file) are scanned to check for valid records, keys, and consistent link structures. This check is done for each key declared in the file.

If ICCHECK detects any errors while processing a particular file, an error file is created according to the Error switch specification. All the same error messages as displayed on the console for that particular file are written to the error file.

If ICCHECK detects no errors on a file that previously had a .er file associated with it, the .er file is deleted.

The following information about an ICISAM file is given unless the No-scan switch was given:

- A possible Warning message that the end-of-file of the .XD file is not on a 2048 byte boundary
- The ICISAM version of the file
- The default deletion type (logical or physical) is given along with the maximum file size (2GB or 4GB)
- The number of alternate keys for indexed files, record size, and number of records allocated
- The total number of record slots available for records before the .XD end-of-file
- For each key it reports:
  - Whether the key is the Primary or an Alternate along with seven possible attributes shown by the possible letters "dursaop" or "-------" if no attributes were given for the file. The possible attributes are shown below:
    - d duplicates are allowed,
    - u upper-case only,
    - r reverse (or DESCENDING) storage,
    - s suppress certain key values (the suppressed value is shown later),
    - a multiple scattered keys using ALSO clause,
    - o multiple tabular keys using OCCURS clause, and
    - p this key has suffixes using PLUS clause.
  - Finally the number of keys, records, and purged keys are shown.
  - If auditing, for each level in the index for that key the number of nodes, the number of keys in that node, and the average density of the nodes.
  - If auditing, the key length and offset in the record including suffixes (PLUS), OCCURS, scattered keys (ALSO), and any suppression value (SUPPRESS WHEN) shown as LOW-VALUE, HIGH-VALUE, SPACE, ZERO, or its octal value.

If ICCHECK detects only that the .NX file is corrupt, the ICFIXUP utility can be used to build a new .NX file. If problems are detected on the .XD file, either a backup copy should be installed, or the ICFIXUP utility should be run to rebuild both the .nx and .xd files.

An exit code of 10 will be returned if a corrupt file was encountered and no other error was detected.

III. ICCREATE

A. **Introduction**

The ICCREATE utility is used to create an empty ICISAM file from the command-line. It has switches to control all of the various parameters of an ICISAM file, including the main ICISAM version and special tuning items like logically or physically deleted records. It also will build a file attribute file which holds all the file information for a particular ICISAM file that can later be used to create another file with similar attributes or fix a corrupt file.

After creating file attribute file(s) you should mark them read-only.

B. **Syntax**

The standard syntax is:

Create File attribute file for an existing file:

```
iccreate [-a[:aflag]|-A file|dir[:aflag]] [-h|-?] [-q] {-f|-F dir}
    { argument }...
```

Create Like another file:

```
iccreate [-a[:aflag]|-A file|dir[:aflag]] [-h|-?] [-q] [-f|-F dir] -L file
    [-D cnt|-R cnt] [-I cnt] { argument }...
```

Create Relative:

```
iccreate [-a[:aflag]|-A file|dir[:aflag]] [-C attr:on|off] [-h|-?] [-q]
    [-f|-F dir] -T r -S min[:max]] [-V version]
    [-D cnt|-R cnt] [-I cnt] { argument }...
```

Create Indexed:

```
iccreate [-a[:aflag]|-A file|dir[:aflag]] [-h|-?] [-q] [-f|-F dir]
    -T i -S min[:max] {-K <keyspecifier>}... [-C b|p:on|off] [-V version]
    [-D cnt|-R cnt] [-I cnt] { argument }...
```

Where
   -a[:*aflag*]|-A *file*|*dir*[:*aflag*] (Audit)
      Enables auditing (default iccreate.lg). Where *aflag* is a|b|d|p|t|u|da|db|pa|pb|ta|tb|ua|ub, defined as a-append, b-backup, d-date, p-pid, t-time, and u-username.
   -C *attr*:on|off (Control attributes)
      Set (on) or clear (off) the indicated file attribute(s). Available attributes are `b' or `p'. Control attribute can be specified multiple times to set each attribute.
         b (Big File) Allows the maximum file size of the file to grow to 4GB. If not set, the maximum file size is 2GB. Only allowed for version 7 files.
         p (Delete-is-physical) Sets the default type of record deletion on this file when neither LOGICAL nor PHYSICAL were specified in the delete operation. For logical deletes, the record is simply marked deleted so it can be undeleted; for physical deletes, the record area space is made available for new records to be written and no undelete can be done as the record is gone. The default is for the delete-is-physical attribute to be off. Only allowed for version 7 files.
   -D *cnt* (Data padding)
      Pad the data files with at least *cnt* 2048 byte blocks of free space. This may cause the file to grow in size. This switch may not be used with the Record padding switch.

-f|-F *dir* (File attribute file)

Create a file attribute file that contains all of the file creation parameters for the file, using the default name *argument*.fa in the current directory or the specified directory. This file can be used by the ICFIXUP utility to recover a corrupted file without having to enter all the parameters from the command-line. This switch can also be used to create an information file for an existing file by specifying just this switch and the name of an existing file as the argument.

-h|-? (Help)

Displays help text.

-I *cnt* (Index padding)

Pad the index file with *cnt* 2048 byte empty blocks of free space.

-K *pos*:*len*[:d][:r][:u][:s[=*val*]][:p=*ppos*:*plen*]...[:o=*cnt*:*span*] (Key specification)

-K *pos*:*len*[:d][:r][:u][:s[=*val*]][:a=*apos*]... (Key specification)

Specifies the keys for indexed files. A key specification must be supplied for each key in the file. At least one key must be specified for an indexed file creation. The number of keys will be determined by the number of key specifications. The first key specification will be for the primary key, all subsequent key specifications will be treated as alternate keys. All the alternate keys are sorted like the COBOL compiler sorts alternate keys allowing them to be specified in any order. Up to 5 key specifiers may be listed for version 5 and 6 files; and up to 17 key specifiers may be listed for version 7 files.

> *pos* specifies a 1-based byte position in the record of the start of the key.
> *len* specifies the length of the key in bytes.
> :d specifies that the alternate key is to allow duplicate keys and is only allowed for alternate keys and is required for version 5 and 6 indexed files which always allow duplicates.

The following key specification options are only allowed for version 7 files.

> :r specifies that this key is to be stored in reverse (DESCENDING) order.
> :u specifies that this key is always stored and retrieved in upper-case-only.
> :s[=*val*] specifies the value to suppress from key insertion and is only allowed on an alternate key. If *val* is not specified, LOW-VALUE is used.
> :p=*ppos*:*plen* specifies suffixed key values (PLUS) at the given position (*ppos*) and length (*plen*).
> :o=*cnt*:*span* and :a=*apos* specify multiple key locations in the record for this key and is only allowed on an alternate key. The :o parameter (OCCURS) gives a tabular view with *cnt* times and how far apart each entry is in bytes (*span*). The :a parameter (ALSO) specifies scattered key values for this key at the indicated positions (*apos*).

-L *file* (Like file)

Create the current file to be like another indexed or relative file that already exists. That file is opened and all of the attributes are set to be like it. If this switch is used, the various padding switches can still be used to pad the empty file that is created, but the other specification switches (-K, -C, -S, -V, and -T) cannot be specified. The *file* argument can also specify a file attribute (.FA) file.

-q (Quiet)

Enables quiet operation.

-R *cnt* (Record padding)

Pad the data file with at least *cnt* records of free space, rounded up to a 2048 byte boundary. This switch may not be used with the Data padding switch.

-S *min*[:*max*] (Size)

Defines the record size, in bytes. Required if the Like switch is not specified. If only *min* is specified all records are of that size. If *max* is specified, records can be between *min* and *max*.

-T i|r (Type)

Sets the file type to indexed (`i') or relative (`r'). If not specified, the default is indexed.

-V *version* (Version)

Create a file of a specific version. The *version* specifier must be an integer with the value 5, 6, or 7. The default is 7.

argument

Specifies the filename to be created. If more than one filename is specified, all of the files will be created with the same parameters. The filename must not already exist, since this would conflict with the intention to create them. However, if the File attribute switch (-f|-F) is the only switch, ICCREATE will create a file attribute file from an existing file, so the argument ICISAM file must exist, but the .FA file may not.

Environment variables:

    ICCREATE   Command line options

C. **Description**

In a key specifier, the :o (OCCURS) and :a (ALSO) can not both be specified for a single key entry.

Alternate record keys are sorted based on the following criteria (just as in the compiler):
    a. ascending root segment position.
    b. ascending root segment length.
    c. absence of also keys and if present ascending number of also and ascending alsos position.
    d. absence of suffixes, and if present ascending number of suffixes, ascending suffix position, and ascending
        suffix length.
    e. absence of occurs, and if present ascending number of occurs and ascending occurs span.
    f. absence of duplicates allowed.
    g. absence of descending order.
    h. absence of uppercase conversion.
    i. absence of suppress when value, and if present ascending suppress when value.

D. **Examples**

The following will create an indexed file called test2 with 100 byte records and a 10 byte primary key starting at character 1 in the record. In addition, the file attribute file "test2.fa" will be created.

```
iccreate -T i -S 100 -f -K 1:10 test2
```

The following will create a version 7 indexed file called test7 with 100 byte records, a 10 byte primary key starting at character 1 in the record, a 10 byte alternate key starting at character 11 that is always stored in upper-case, and a 20 byte alternate starting at character position 21 that allows duplicates.

```
iccreate -T i -V 7 -S 100 -K 1:10 -K 11:10:u -K 21:20:d test7
```

IV. ICFIXUP

A. **Introduction**

The ICFIXUP utility is used to recover corrupted ICISAM indexed and relative files. This is done in-place, thus preserving certain operating system level information such as UNIX hard links. The purpose of this utility is to recover a corrupted file, not change it or optimize it in any way, thus there are no options for any type of reorganization operations during the recovery process.

B. **Syntax**

The standard syntax is:

```
icfixup [-a[:aflag]|-A file|dir[:aflag]] [-C b|p:on|off] [-f|-F dir]
    [-h|-?] [-K <keyspecifier>]... [-p] [-q] [-r] [-S min[:max]] [-T i|r]
    [-U d|i|di ] [-V version] { argument }...
```

Where
    -a[:*aflag*]|-A *file*|*dir*[:*aflag*] (Audit)
        Enables auditing (default icfixup.lg). Where *aflag* is a|b|d|p|t|u|da|db|pa|pb|ta|tb|ua|ub, defined as a-append, b-backup, d-date, p-pid, t-time, and u-username.
    -C *attr*:on|off (Control file attribute)
        Set (on) or clear (off) the indicated file attribute(s). Available attributes are `b' or `p'. Control attribute can be specified multiple times to set each attribute.
            b (Big File) Allows the maximum file size of the file to grow to 4GB. If not set, the maximum file size is 2GB. Only allowed for version 7 files.
            p (Delete-is-physical) Sets the default type of record deletion on this file when neither LOGICAL nor PHYSICAL were specified in the delete operation. For logical deletes, the record is simply marked deleted so it can be undeleted; for physical deletes, the record area space is made available for new records to be written and no undelete can be done as the record is gone. The default is for the delete-is-physical attribute to be off. Only allowed for version 7 files.
    -f|-F *dir* (File attribute file)
        Specifies the file attribute file. If -f, use the default file (*argument*.fa). If -F *dir*, use the default file in the specified directory.
    -h|-? (Help)
        Displays help text.
    -K *pos*:*len*[:d][:r][:u][:s[=*val*]][:p=*ppos*:*plen*]...[:o=*cnt*:*span*] (Key specification)
    -K *pos*:*len*[:d][:r][:u][:s[=*val*]][:a=*apos*]... (Key specification)
        Specifies the keys for indexed files. A key specification must be supplied for each key in the file. At least one key must be specified for an indexed file creation. The number of keys will be determined by the number of key specifications. The first key specification will be for the primary key, all subsequent key specifications will be treated as alternate keys. All the alternate keys are sorted like the COBOL compiler sorts alternate keys allowing them to be specified in any order. Up to 5 key specifiers may be listed for version 5 and 6 files; and up to 17 key specifiers may be listed for version 7 files.
        pos specifies a 1-based byte position in the record of the start of the key.
        len specifies the length of the key in bytes.
        :d specifies that the alternate key is to allow duplicate keys and is only allowed for alternate keys and is required for version 5 and 6 indexed files which always allow duplicates.
        The following key specification options are only allowed for version 7 files.
        :r specifies that this key is to be stored in reverse (DESCENDING) order.
        :u specifies that this key is always stored and retrieved in upper-case-only.
        :s[=*val*] specifies the value to suppress from key insertion and is only allowed on an alternate key. If *val* is not specified, LOW-VALUE is used.
        :p=*ppos*:*plen* specifies suffixed key values (PLUS) at the given position (*ppos*) and length (*plen*).

35

:o=*cnt*:*span* and :a=*apos* specify multiple key locations in the record for this key and is only allowed on an alternate key. The :o parameter (OCCURS) gives a tabular view with *cnt* times and how far apart each entry is in bytes (*span*). The :a parameter (ALSO) specifies scattered key values for this key at the indicated positions (*apos*).

-p (Progress)

There is an ongoing display of the processing, which typically runs much slower for smaller files. The reporting interval is for every 1% of the file processed.

-q (Quiet)

Enables quiet operation.

-r (Rebuild)

Unconditionally rebuild the file, even if ICFIXUP detects no corruption.

-S *min*[:*max*] (Record size)

Specify the record size, in bytes (not including any overhead added by the system). This switch is required if the headers are bad and the File attribute switch (-f|-F) is not used. If only *min* is specified all records are of that size. If *max* is specified, records can be between *min* and *max*.

-T i|r (Type)

Set the file type to indexed (`i') or relative (`r'). The default is indexed.

-U d|i|di (Unreliable portion)

Indicate that the specified portion(s) (d-data, i-index) of the file is corrupt and to ignore that portion of the file. Both d and i can be specified. For -d the utility will expect a specification for the type, version, record size, record count, and keys, unless it is a version 7 file and the .NX header is not also marked corrupt (version 7 files duplicate all the information in both headers). The file attribute switch could be used.

-V *version* (Version)

Specify the file version number. The *version* specifier must be an integer with the value 5, 6, or 7. The default is 7. This switch is required if the headers are bad and the File attribute switch (-f|-F) is not used.

argument

Specifies a filename or template. Using a template is only meaningful/useful if the corresponding file attribute files are used to specify the fixup parameters, otherwise the same set of explicitly specified parameters would be applied to all files.

Environment variables:

ICFIXUP	Command line options
ICTMPDIR	Temporary files

C. **Description**

The following information about an ICISAM file is given:

- A possible Warning message that the end-of-file of the .XD file is not on a 2048 byte boundary
- The ICISAM version of the file
- The default deletion type (logical or physical) is given along with the maximum file size
- The number of alternate keys for indexed files, record size, and number of records allocated
- The total number of record slots available for records before the .XD end-of-file
- For each key it reports:
    - Whether the key is the Primary or an Alternate along with seven possible attributes shown by the possible letters "dursaop" or "-------" if no attributes were given for the file. The possible attributes are shown below:
        - d duplicates are allowed,
        - u upper-case only,
        - r reverse (or DESCENDING) storage,
        - s suppress certain key values (the suppressed value is shown later),
        - a multiple scattered keys using ALSO clause,
        - o multiple tabular keys using OCCURS clause, and
        - p this key has suffixes using PLUS clause.

- Finally the number of keys, records, and purged keys are shown.
- For each level in the index, for that key the number of nodes, the number of keys in that node, and the average density of the nodes.
- The key length and offset in the record including suffixes (PLUS), OCCURS, scattered keys (ALSO), and any suppression value (SUPPRESS WHEN) shown as LOW-VALUE, HIGH-VALUE, SPACE, ZERO, or its octal value. Finally the maximum key entries per index node is given.
- The total number of indexed nodes.
- The number of logically deleted records in the file.

When rebuilding a file, purged records are removed from the file and records from the end of the file are moved to replace each purged record, thus logically shrinking the file.

If ICFIXUP detects an attempt to insert a duplicate key value into an index which does NOT allow duplicate key values, the offending record will be written to a separate collision file and eliminated from the ICISAM INDEXED or RELATIVE file. If the error is detected on the primary key, then the record space is collapsed from the file; otherwise, if the error is detected on an alternate key, then the record space is placed on the purged record list. The collision file is a simple fixed or varying length SEQUENTIAL file having the same name as the ICISAM file but with the ".fix" filename extension. ICREORG can be used to add these records back into the ISAM file as:

```
icreorg -I s:size filename.fix filename
```

Remember when using ICREORG in this fashion that Logically deleted records can cause a Duplicate Key error for alternate keys that do NOT allow duplicates. Remember, when using logical deletes with records with alternate keys that DO NOT ALLOW DUPLICATES, a Duplicate key error can be given for an alternate key that points to a deleted record. The record must be physically deleted to insert a new record with the same alternate key.

Alternate record keys are sorted based on the following criteria (just as in the compiler):

a. ascending root segment position.
b. ascending root segment length.
c. absence of also keys and, if present, ascending number of also and ascending alsos position.
d. absence of suffixes and, if present, ascending number of suffixes, ascending suffix position, and ascending suffix length.
e. absence of occurs and, if present, ascending number of occurs and ascending occurs span.
f. absence of duplicates allowed.
g. absence of descending order.
h. absence of uppercase conversion.
i. absence of suppress when value and, if present, ascending suppress when value.

D. **Recommendations**

It is recommended that file attribute files (.FA) be built for all ICISAM files, ICCREATE can be used to do this function. Then allow ICFIXUP to use the file attribute file (-f|-F) to insure valid creation information for the ICISAM file and to keep from having to re-enter information about the file.

V. **ICINFO**

A. **Introduction**

The ICINFO utility is a diagnostic tool to be used in detecting the particular state of a machine.

UNIX and Windows each have a different flavor of ICINFO; please consult the appropriate version in the following sections.

The ICINFO syntax does not follow any particular guidelines.

The output of the ICINFO program should be included with any problem report. Do an "icinfo -a" and send the contents of icinfo.lg.

B. **UNIX ICINFO**

The UNIX ICINFO utility is provided as a diagnostic tool.

ICINFO tests some of the problem areas more frequently encountered by users and displays a report that should be submitted for all support requests.

Some of the information is intuitive while some only makes sense to Interactive COBOL's internal developers.

The syntax is:

```
icinfo [-a|-A file] [-h] [-n] [-t]
```

Where
  -a|-A *file* (Audit)
      Enables auditing (default icinfo.lg).
  -h (Help)
      Displays help message
  -n (network only)
      Only do Network scan
  -t (Timing)
      Enables timing tests

SCREEN 1 is a sample listing of the UNIX ICINFO output.

```
icinfo Revision 4.61 (Linux for x86)
Started: ...
Current pid: 9868.
Parent pid: 26025.
Current directory: /home/ralph
ttyname(0):  >>/dev/ttyp0<<   ttyname(1):  >>/dev/ttyp0<<
Testing Ioctl.
                st_mode   st_ino    st_dev    st_rdev   st_nlink
fstat(0)        00002180  00001595  00000128  00001605  00000001
fstat(00000000) 00002180  00001595  00000128  00001605  00000001
stat(ttyname)   00002180  00001595  00000128  00001605  00000001
fstat(1)        00002180  00001595  00000128  00001605  00000001
fstat(00000001) 00002180  00001595  00000128  00001605  00000001
stat(.)         000041FF  00000D4E  012A      00000000  00000002
statfs of (/):f_bsize: 1024  f_blocks: 330378   f_bfree: 101722
statfs of (.):f_bsize: 1024  f_blocks: 1000000  f_bfree: 51492
Login-name: >>ralph<<(long)  >>ralph<<(short)  User-id: 201  Group-id: 50
  User-id: 1002     Group-id: 1002
  Effective: User-id: 1002 Group-id: 1002
  Home dir: /home/ralph
  Supplementary groups (3) found:
    3, 4, 5
Max children (sysconf): 999
Memory: Page size: 4096 byets, Total Pages: 514476, Avail Pages: 292895
        Total: 997 MB, Avail: 62 MB.
Max open files/process(sysconf): 1024  (ulimit): 1024.
Max file size: 1073741824 bytes.
System name: Linux, Node name: intel1,
  Release: 2.6.9-42.0.10.ELsmp,
  Version: #1 SMP Fri Feb 16 17:17:17 EST 2007
  Machine: i686.
Machinename: intel1.
From /proc/cpuinfo.
  # of cpus: 1.
  Model name: xxxxx
From /proc/meminfo.
  ....
Some /proc/sys entries.
  kernel.shmmax: 200000000.
  vm.swappiness: 60
  net.i[v4.tcp_keepalice_time: 7200 (seconds).
  ...
Network Interface scan.
  Interface: 'lo' IP address: 127.0.0.1
  Interface: 'eth0' IP address: 192.168.2.2
  Interface: `eth0'  MAC address: xx:xx:xx:xx:xx:xx
INFO: Created tmp$$01
Locking++++++++++++++++++
Locks currently available are: 197  with Out of Locks errno=46.
(Ftime) Time in seconds past epock: 824216551 millsec 230
Before: >>abcdefghijklmnopqrstuvwxyz<< After: >>abcdefghijabcdefgh
Nosmear forward
Before: >>ABCDEFGHIJKLMNOPQRSTUVWXYZ<< After: >>KLMNOPQRSTUVWXYZQR
Nosmear backward
Finished: ...
icinfo is finished.
```

SCREEN 1. UNIX ICINFO

If ICINFO was started with a pipe (i.e., icinfo | pg etc.) then an error will be generated showing ttyname1 because standard-out has been redirected and the ttyname call is not valid at that time. This does not indicate a problem.

The first line after the icinfo started message gives the current pid of the icinfo process.

Generally the lines fstat(0), fstat(00000000), stat(ttyname), fstat(1), and fstat(00000001) should show exactly the same values since standard-in and standard-out should be the same.

The last set of lines (from Login-name) provide much of the easy to understand information. These are:

The login-name line shows both a long and short version of the current username. If these two names are different then Interactive COBOL and other user utilities that use username could become confused. You

should probably shorten your username. Next comes the user-id associated with the current username along with the current group setting and any supplemental groups that the user belongs.

The next line shows the maximum number of processes available.

The next line shows the maximum number of operating system file handles per process allowed by this kernel configuration. If it is not enough, rebuild the kernel with a larger value.

The next line shows the maximum file size allowed on this system.

The next two lines show the machine name and particular operating system information, including release and version number and machine architecture.

On some machines information from the /proc filesystem will be shown.

On some machines a "Network Interface scan" section will be shown. Here any network interfaces that can be detected and their IP and MAC address, if present, is shown.

The next two lines show how many record locks are available by actually trying to lock until it gets a lock error and what the lock error is. If it is not enough, rebuild the kernel with a larger value. If an error is received on the open then for some reason icinfo cannot create a file to test locking. If icinfo hangs at this point then for some reason locks are not allowed for this file (and/or the filesystem) that this file resides on. DO NOT use icrun to try to open files on this filesystem as the runtime will HANG!

Next a check for ftime is made.

The next four lines detect if the memmove routines in libc smear.

If the Timing option was selected then:

> The next two lines insure that the timer is working correctly.

> Finally the CPUINDEX gives a very simplistic cpu benchmark of how this machine compares to others. This benchmark is only as accurate as the setting in which it was taken. To make this more accurate, insure that no other processes are running while this timing is in effect. The "Others" numbers are just a basis of comparison to other very popular machines.

C. **Windows ICINFO**

The Windows ICINFO utility is provided to give detailed information about a particular system. It displays a report that should be submitted for all support requests.

Running ICINFO will tell you if WinSock is not at the correct revision. This could be possible if you are running a Windows 95 original version with no updates. If you get a DLL load error saying the SNMPAPI.DLL could not be loaded then you need to rev up. The error will usually be in a PopUp box and say:

> "A required .DLL file, SNMPAPI.DLL was not found.".

Download the update from Microsoft, our Web site, or copy it from the **ICOBOL** install CD. It will be called ws2setup.exe or w95ws2setup.exe. Then run the executable and then re-boot.

On the Microsoft Web site search for "Windows Socket 2" for more information.

The syntax for icinfo is:

```
icinfo [-a|-A file] [-h|-?] [-G option...|-N option... ]
```

Where
    -a|-A *file* (Audit)
        Enables auditing. Default icinfo.lg.
    -h|-? (Help)
        Displays help text
    -G *option* (Only option)
        Specifies to only display that particular *option (s)*
    -N *option* (No option)
        Causes the specified *option* (s) to NOT be displayed by default

    Valid *options* are:
c	Displays the CPU type (8088/6, 80286, 80386, 80486, Pentium, Pentium-Pro, Celeron, Pentium II, Pentium III, Pentium 4), and other cpu-related info. For Pentium III type processors the serial number is also shown if available.
d	Drive information
f	File version information (not default)
h	Check for hung semaphores
i	Displays install.txt (requires "r")
k	Displays the keyboard type and other language information
l	Performs a lock test
m	Displays memory information
n	Displays network information including whether TCP/IP is loaded
p	Display printer information including the default printer and all installed printers
r	Displays registry information
s	Displays system name and startup information
t	Displays sleep and CPU timings (not default)
v	Displays verbose information, especially for CPU type
z	Display miscellaneous cpu flags. (not default)

SCREEN 2 is a sample listing of the Windows ICINFO output.

```
icinfo Revision 4.61 (Windows)
Copyright (C) 1987-2012, Envyr Corporation.  All rights reserved.
Started: Mar-04-2012 13:40:29.59
System is running Microsoft Windows XP Professional
    Version 5.1 Build 2600 Service Pack 2.00 (Service Pack 2)
    Suites Installed:
      SingleUserTS
Systemname: RALPHJ
    System was started with a normal boot
    Startup Info:
      Console title:   "icinfo -a"
      Station\desktop: WinSta0\Default
    WindowsDirectory: C:\WINDOWS
    SystemDirectory: C:\WINDOWS\system32
Drive information for the current drive. (C:\)
    Current directory: C:\test200\
    Filesystem:   NTFS
    Total space:      40,007,729,152 bytes.
    Free space:        7,364,612,096 bytes.
Intel detected
System (CPUID & brand-string) has an Intel(R) Pentium(R) 4 CPU 2.80GHz
  Hardware information (from os)
    Processor Architecture: Intel
    Number of processors: 2
    Page size: 4096
    Minimum application address: 10000
    Maximum application address: 7ffeffff
    Active processor mask: 3
System Memory (ex):
    Total Physical Memory:        1030888 Kbytes
    Available Physical Memory:     475872 Kbytes
    Total PageFile Size:          1697204 Kbytes
    Available Page Space:         1279796 Kbytes
Keyboard type: 4  [IBM Enhanced 101 or 102 keys]  12 Function keys
    Keyboard code: 00000409        Keyboard Layout: English (U.S.)
Network detected.
    Username: ralph
    TCP/IP Installed.
MAC address of Ethernet interface: 00-0C-F1-AE-45-7C
Default printer: HP LaserJet 5/5M PostScript
    Default Printer Status: 0x0000
List of printers:
    HP LaserJet 5M
    HP LaserJet 5/5M PostScript
ICOBOL 4 software found.
  ICOBOL Installed Revision:   4.00
    ICOBOL Install file:      C:\Program Files\ICOBOL\Install.txt
    ICOBOL Working Directory: C:\test200\
    ICOBOL Default ICROOT:    C:\Program Files\ICOBOL
    ICOBOL Default ICCODEPATH: C:\Program Files\ICOBOL
    ICOBOL Default ICPERMIT_MACHINE:
Windows NT Registry Entries.
  Heap: %SystemRoot%\system32\csrss.exe ObjectDirectory=\Windows
SharedSection=1024,3072,256... MaxRequestThreads=16
  LanmanWorkstation UseLockReadUnlock: 0.
  No LanmanWorkstation UseOpportunisticLocking found (defaults to 1).
  No LanmanWorkstation SessTimeout found (defaults to 45 sec).
  No LanmanWorkstation KeepConn found (defaults to 600 sec).
Warning:  LanmanWorkstation EnableSecuritySignature: 1.
  LanmanWorkstation RequireSecuritySignature: 0.
  LanmanServer EnableOplocks: 0.
Warning:  LanmanServer EnableSecuritySignature: 1.
  LanmanServer RequireSecuritySignature: 0.
  No LanmanServer EnableOplockForceClose (defaults to 0).
  Product Type: WinNT
  No TCPIP KeepAliveTime found (def. 7,200,000 milliseconds (120 minutes)).
No ICRUN environment entry found.
Semaphore check:
  OPEN/CLOSE    semaphore is OK.
  BUFFER        semaphore is OK.
  LOCK          semaphore is OK.
  LOGON/LOGOFF semaphore is OK.
LockFileEx Test Succeeded, LockFileEx is available.
Time is: Mar-04-2012 13:41:09.60
Listing of C:\Program Files\ICOBOL\Install,txt
-------------------------------------------
....  ( listing of Install.txt ) ....
-----------End Listing Install.txt-----------
icinfo is finished
```

SCREEN 2. WINDOWS ICINFO

From top to bottom:

The operating system, revision, and build information. (Always shown)

43

The *systemname* is the machine name known to the operating system, startup information, and the Windows directory and system directory. (Option s)

The type of machine architecture and serial number for Pentium III type processors. (Option c)

The next several lines show physical and virtual memory status in Kbytes. (Option m)

The keyboard type and code, language, and code page selections are shown. (Option k)

Whether a network was detected or not and if so, the current username, whether TCP/IP is running, and the MAC address(s) if found of the network card(s). (Option n)

The default printer is shown along with the current list of available printers on this machine. (Option p)

The registry is checked for any ICOBOL software and its information is reported along with several system specific settings important to **ICOBOL** users . (Option r)

If the timing option was selected, several timing tests are performed. (Option t)

Finally the listing of the **ICOBOL** Install.txt file is shown. (Option i)

ICINFO will detect and show SMB Signing information from Workstation and Server parameter lists in the registry. These show up as "EnableSecuritySignature" and "RequireSecuritySignature". A setting of 0 is disabled and a setting of 1 is enabled.

If a Warning is shown relating to these settings, especially for "RequireSecuritySignature" AND you are having some "Access denied" problems or "The specified network name is no longer available" connecting over the network or you are receiving some "delayed write failures" over the network then you need to look into the possibility of disabling these settings.

For more information on these settings you can go to the MicroSoft site and search for "SMB Signing", "EnableSecuritySignature", or "RequireSecuritySignature".

One other note is that having SMB signing enabled, generally has a network performance penalty of from 10 to 15%.

VI. ICLIB

A. **Introduction**

The ICLIB utility allows the user to create and/or modify a library file. ICLIB allows for additions, replacements, extractions, listings, statistics, and deletions of files in the library.

The library file can be used by the runtime system to find COBOL programs (.CX files) by placing the library filename in ICCODEPATH as a separate entry.

Although the library file is primarily used for program files (.CX), any file (even an ICISAM file) can be placed into the library file. Under the runtime, data files can be used from the library file by placing the library filename in ICDATAPATH as a separate entry. Data files found in the library can only be opened for read access. Files in a library can not be processed by any command-line utility except for those that support a library switch (ICREV and ICREVSET) unless the files are first exported from the library file.

The library file can also be used as a software release mechanism to place all of the files in a release into a single large file and then extract the needed portions at the customer site.

B. **Syntax**

The standard syntax is:

Help
```
    iclib -h|-?
```

List Contents
```
    iclib [-a[:aflag]|-A file|dir[:aflag]] [-q] -l libname [argument]...
```

Import
```
    iclib [-a[:aflag]|-A file|dir[:aflag]] [-q] [-v] -i [-r|-n] libname
        argument...
```

Export
```
    iclib [-a[:aflag]|-A file|dir[:aflag]] [-q] [-v] -x [-r|-n] libname
        argument...
```

Delete
```
    iclib [-a[:aflag]|-A file|dir[:aflag]] [-q] -d libname argument...
```

Statistics
```
    iclib [-a[:aflag]|-A file|dir[:aflag]] [-q] -s libname
```

Where
    -a[:*aflag*]|-A *file*|*dir*[:*aflag*] (Audit)
        Enables auditing (default iclib.lg). Where *aflag* is a|b|d|p|t|u|da|db|pa|pb|ta|tb|ua|ub, defined as a-append,
        b-backup, d-date, p-pid, t-time, and u-username.
    -h|-? (Help)
        Displays help text.
    -q (Quiet)
        Enables quiet operation.
    -s (Statistics)
        Displays the statistics for the library file including the number of files, the size of the library in KB, and the
        number of deleted entries.

-v (Verbose)

Enables verbose operation, causing messages to be displayed for every step it executes. By default, only errors and summary messages are emitted.

libname (Library filename)

Specifies the library filename to use and is required. Most of the utilities that look at a library file default to the name icobol.cl. If the `.cl' extension is not given, it is appended.

-l (List contents)

Displays a sorted list of files in the library including size and last modification. If no arguments are given, all files in the library are listed; otherwise, only the specified files are listed.

-i [-r|-n] (Import)

Works in combination with the Replace switch (-r) or Newer switch (-n) to import the files specified by the argument(s) into the library. Without the Replace switch or Newer switch, only files that are not already in the library are added. The Replace switch says to add new files and replace any files with the same name. The Newer switch says to add new files and replace ONLY IF the one in the library is older than the new file.

-x [-r|-n] (Export)

Works in combination with the Replace switch (-r) or Newer switch (-n) to export files specified by the argument(s) from the library. Templates for extraction can not have directory specifiers. Without the Replace switch or Newer switch, only files that do not already exist in the destination directory are exported from the library. The Replace switch says to extract files and to replace any existing files that have the same name. The Newer switch says to extract files and replace an existing file ONLY IF the one in the library is newer than the one in the destination directory.

-d (Delete)

Deletes the files specified by the given template(s) from the library file. At least one template or filename is required.

argument

Can be a filename or template. The Export (-x) and Delete (-d) switches require the names and templates to be simple names (i.e., they may not have a directory prefix).

Environment variables:

ICLIB	Command line options
ICTMPDIR	Temporary files

C. **Description**

The Statistics, List, Import, Delete, and Export library switches are mutually exclusive.

When using the Newer switch with Import and Export, the file system modification date is used for comparison of times.

ICLIB can be used to extract all the old files, delete the old library, and then create a new library file. From time to time, the statistics switch should be used to determine if there is a large amount of unused free space in the library. This can be caused by replacing files with ones that are slightly larger, causing ICLIB to allocate a new area for the file. When there is a lot of free space, the library can be compacted by extracting all the files, deleting the old library, constructing a new one, and then deleting the file. However, this will cause the "modified" time stamp to be changed for all of the files in the library.

ICLIB opens the library file with the exclusive option so it will fail if another Interactive COBOL process has the library open. Similarly, while ICLIB has the library file open no one else can use the library file.

If ICLIB deletes the last entry in a library, a warning is issued and the library is deleted.

On UNIX, when performing the Delete, List, or Export operations with a template as an argument, it may be necessary to quote the template to prevent the template from being expanded by the shell.

Notes:

1. On Windows, all files will be loaded into the library as lower-case filenames to insure that they will be extracted as lower-case files on UNIX.

2. On UNIX, files will be loaded into the library in the case that the filename is presented to the utility. If the files are loaded on UNIX and extracted on UNIX their cases will be preserved provided Note 3 below is adhered to. Generally we recommend only lower-case files be stored in the library.

3. All files in a library are identified in a case-insensitive fashion. Thus no two filenames can have the same name in different cases, i.e., "A" and "a" represent the same file.

4. The library stores at most a 255-character name in the library file. Any file with a simple name longer than 255 characters will generate an error and will not be stored in the library.

VII. ICLINK

A. **Introduction**

The ICLINK utility allows the user to specify an alternate set of filenames for the files named in the COBOL programs executing under Interactive COBOL. It allows for a particular filename used in an OPEN, CALL, or CALL PROGRAM statement to be remapped (i.e., linked) to a new filename without changing the program. The linking can apply to entire pathnames. The utility also allows for exporting the contents of a link file.

The filename linking facility was specifically implemented to enhance the portability of existing Interactive COBOL programs.

Another important use for ICLINK is to use files on an ICNETD server (client/server). From the current working directory (usually on the local machine), certain files must be redirected to the server machine running ICNETD. ICLINK can be used to accomplish this without having to change the COBOL program(s).

The template characters "*" and "?" can be used to facilitate linking similar files.

ICLINK does NOT support linking files with embedded spaces to new filenames since space is used to delimit old names from new names in the ICLINK input file. New names can have embedded spaces on UNIX and Windows.

B. **Syntax**

The standard syntax is:

Help
```
iclink -h|-?
```

Import
```
iclink [-a[:aflag]|-A file|dir[:aflag]] [-q] -i [-r] linkfile textfile
```

Export
```
iclink [-a[:aflag]|-A file|dir[:aflag]] [-q] -x [-r] linkfile textfile
```

Statistics
```
iclink [-a[:aflag]|-A file|dir[:aflag]] [-q] -s linkfile
```

Where
    -a[:*aflag*]|-A *file*|*dir*[:*aflag*] (Audit)
        Enables auditing (default iclink.lg). Where *aflag* is a|b|d|p|t|u|da|db|pa|pb|ta|tb|ua|ub, defined as a-append, b-backup, d-date, p-pid, t-time, and u-username.
    -h|-? (Help)
        Displays help text.
    -q (Quiet)
        Enables quiet operation.
    -s (Statistics)
        Displays statistics about the *linkfile*. These include the number of names, the size of the file (in KB), the average entry link, and the average lookup chain length.
    -i (Import)
        Read the line sequential file given by *textfile* and add the names to the specified *linkfile* based on whether the Replace switch was specified.
    -x (Export)
        Creates a line sequential file specified by *textfile* with the old-name, new-name pairs in it. This file is in a format suitable for importing into a new link file. If the file does not exist, it is created. If it does exist, and

the -r switch is set, it is erased and the export list is written to it. If the -r switch is not set, the export list is appended to the file.

-r (Replace)

Specifies that during an import, any old-filename that already exists in the link file will automatically be replaced by a duplicate entry from the import file. Without this switch, such names are flagged with a warning and the replacement does not occur. For an export operation, the contents of the export file are erased and replaced by the new list, otherwise the list is appended to the file.

linkfile

Specifies the name of the linkfile to be used. If the .lk extension is not given, it is appended.

textfile

Specifies the name of a line sequential file containing the old- and new-filenames.

Environment variables:

ICLINK   Command line options
ICTMPDIR   Temporary files

The syntax of the textfile is:

old-filename space... *new-filename* <line-terminator>

.

.

Where

old-filename

Can be any set of characters except space and not longer than 255 characters. The template characters "*" and "?" can be used to match many or a single-character(s) respectively.

new-filename

Any string of characters beginning with the first non-blank character and going until a line-sensitive terminator (CR or NL) with any trailing spaces stripped. If template characters ("*" and/or "?") were given in the *old-filename* then matching template characters must be given in *new-filename*.

Remember the template "*.*" does NOT match the file "joe000" since it contains no "." character.

C. **Description**

A single source line cannot be longer than 255 characters.

To use ICLINK, first create a standard text file with each line containing the old-filename and the new-filename separated by at least one space. Then run ICLINK using that file as its textfile-argument and a link filename to create a linkfile.

If the ICRUNLK environment variable is set, this file is read by the Interactive COBOL runtime at startup time and used dynamically to replace occurrences of old-filenames with corresponding new-filenames. This replacement is done before the runtime adds the standard extensions to the names (.XD/.NX for ICISAM files or .CX for COBOL program files).

All old-filenames are stored and looked-up in a case-insensitive manner. Thus, attempting to import the old-filenames "ABC" and "abc" would cause an error unless the Replace switch (-r) was used, in which case the last one will replace the previous.

ICLINK opens the link file it is creating with the exclusive option so that no other Interactive COBOL process can access it while it is being constructed.

Upon invocation of the Interactive COBOL runtime, if the ICRUNLK entry is specified then the specified link file is activated, otherwise no filename linking is performed. The link file is opened for read-only access, read into memory, and then closed.

Link files are interchangeable from one machine to another and from one operating system to another.

Exact matches will be used before template matches. Template matches will be used in the order they appear in the source file.

The following is allowed to link "$lpt" to "|>lp -ddest -ond"

```
    $lpt <space>... |>lp -ddest -ond   <line-terminator>
```

These new-filenames with embedded spaces and illegal characters are only useful <u>on UNIX</u> in a sequential OPEN.

The link facility can also be used to open files using the ICNETD facility (client/server) by specifying ICNETD names as the new-filename. The template characters provide for whole groups of files to be mapped in this fashion. For example,

```
    data\a*        @\\machine6\data\a*
```

would link all files starting with "data\a" (including "data\a") over to machine6 using ICNETD.

> **Note:**   <u>On UNIX</u>, it is possible when running in lower-case mode or upper-case mode to pass filenames of the other case through, by selecting a new-filename of the opposite case. For example, when running Interactive COBOL on UNIX in lower-case mode the mapping "joe   JOE" will allow the upper-case filename "JOE" to be opened.

VIII. ICLOG

A. **Introduction**

The log management utility (ICLOG) provides the functionality to externally control the logging and recovery from backup (i.e., 'rescue') of modifications performed on ICISAM files since the time the backup was taken. It will:

A. enable (and initialize) logging of individual files; (-O e)

B. provide generations of log files; (-O n)

C. disable (and remove) logging of individual files; (-O d)

D. report status information regarding logging of individual files; (-O l)

E. apply logged modification operations (i.e., roll forward) to backups of individual files; (-O a)

F. Check or Test logged files; (-O c, -O t)

G. Fixup corrupt network files; (-O f)

H. Resync local files from their remote log and/or mirror; (-O s, -O r)

Logging can be performed in "Local" mode or "Network" mode. Local mode implies that all the files, both ICISAM and log are on the same machine. Network mode implies that some portion of the file is on a remote server. In this mode either the log file can be remote (remote logging) or an entire mirrored set of files can be on a remote server (mirroring). Network mode requires an ICLOGS license for a logging server (iclogs) that runs under control of ICNETD on the remote server.

B. **Syntax**

The standard syntax is:

Local mode:

```
iclog [-a[:aflag]|-A file|dir[:aflag]] [-D A|S|cCikpPrtU] [-h|-?] [-L dir]
    [-o] [-O a|c|d|e|l|n|u] [-p] [-q] [-v] [argument]...
```

Network mode:

```
iclog [-a[:aflag]|-A file|dir[:aflag]] [-D A|S|cCikpPrtU] [-h|-?] [-L dir]
    [-M server] [-o] [-O a|c|d|e|f|l|n|p|r|s|u] [-p] [-q] [-R server] [-s]
    [-U usernam e] [-v] [argument]...
```

Where
  -a[:*aflag*]|-A *file*|*dir*[:*aflag*] (Audit)
    Enables auditing (default iclog.lg). Where *aflag* is a|b|d|p|t|u|da|db|pa|pb|ta|tb|ua|ub, defined as a-append, b-backup, d-date, p-pid, t-time, and u-username.
  -D A|S|cCikpPrtU (Details)   [requires List operation (-O l)]
    When listing (-O l) specify Details to list: A-all, S-Summary, or any combination of:
        c-console, C-Computername, i-info on process, k-key value, p-process id, P-Program name, r-record data, t-timestamp, U-Username
  -h|-? (Help)
    Displays help text.

-L *log-dir* (Log location)   [requires Enable operation (-O e)]
        Location of log file (*.xl) in the directory *log-dir*. Can be absolute or relative.
-M *server* (Mirror to server)   [requires Enable operation (-O e)]
        Specify server machine to mirror file. (icnetd+iclogs)
-o (Override)
        Override an error condition. Should be used with care,
-O acdeflnprstu (Operation)
        Specifies a single operation to perform:
                a (Apply operation),
                c (Check operation)
                d (Disable operation),
                e (Enable operation),
                f (Fixup operation),   (only for remote or mirror)
                l (List operation),
                n (Next generation operation),
                p (Promote slave operation),   (only for mirror, run on mirror server)
                r (Recover master operation),   (only for mirror)
                s (Synchronize operation),   (only for remote)
                t (Test operation),
                u (Upgrade operation(from 1 to 2)). (only for a rev 1 log file)
-p (Progress)
        Enable progress reporting.
-P *password* (Password)   [requires Remote or Mirror option]
        Specify the password for a remote operation.
-q (Quiet)
        Enables quiet operation.
-R *server* (Remote log to server)   [requires Enable operation (-O e)]
        Specify a server for a remote log (.xl) file (icnetd+iclogs)
-s   (Save)   [requires Remote or Mirror option]
        Update saved network login information
-U *username* (Username)   [requires Remote or Mirror option]
        Specify username for remote operation
-v (Verbose)
        Provide verbose messages

argument
        argument specifies an ICISAM file(s). If no argument is given, "*.xd" is used.

Environment variables:

    ICLOG   Command line options

Logging is handled by the underlying ICISAM/io system in that any operation that could cause a modification to an ICISAM file is written to a log file(.xl). Logging is invisible to the application itself. In the remote or mirror case when the network or remote/mirror server becomes inaccessible the main system will continue running with only a slight hesitation. A message will be written to the log file for the process that initially detects the inaccessibility.

For Local mode, the log file is only on the same machine as the main file and records are written to it with the Immediate option such that the record is written to disk directly. When enabled in this fashion, logging does cause slower update operations.

For Network mode (Remote or Mirror options), a local log file is kept in addition to a Remote log and/or Mirror fileset but the local log is not written immediately. Network write's to the remote and/or mirror are written at the same time as the local log to ensure operations are logged in the correct order.

Log filenames use the base ISAM name followed by the generation number and then the .xl extension. For example, the ISAM file "testa" which encompasses the files testa.xd and testa.nx will have an initial log filename of testa.1.xl.

Log generations are a way to keep track of various logs and to keep them at a reasonable size. The Next generation operation closes out a log generation and increments to the next generation.

In general the use of logs can be viewed as:

1. Enable logging for a file or set of files.
2. Take a backup of those files (.xd/.nx)
3. Start running
4. If a crash happens and the .xd/.nx are corrupt, the backup can be loaded and an Apply operation can be done using the log or logs.

By default, log files use the same path as the file being logged. The Location option can be used to specify a different directory location for a log file. Either a relative or absolute location can be specified. A relative location could be "-L logs" which says to create the log file(s) in the "logs" subdirectory of the directory where the actual file is located. An absolute location like "-L \mainlogs" would create all log files in the "\mainlogs" directory. For Windows machines, this will be on the drive where the file is located. For Unix machines, this will be from the root.

In general, ICLOG will open all files (logs and ICISAM) exclusively unless it is not necessary to do so for status reporting operations. In particular the Check and List operations do NOT required exclusive access to the log file. ICLOG will not operate on any file which is marked as unreliable. ICISAM Version 5 and 6 files are not supported, and Version 7 files will be upgraded to Version 7.20 as necessary.

All operations, except Enable, require that the file has logging previously enabled.

C. **Description**

For Local Mode

The *Enable Operation* (-O e) establishes the logging ability for a previously log-disabled file. It turns on the log enable flag and sets the generation number to one(1) in the .XD file header of each ICISAM file, and creates the corresponding empty log file with a generation number of one (1). Attempting to enable logging for a file which does not exist, or for which logging has already been enabled generates an appropriate error. ICISAM files will be upgraded to ICISAM Version 7.20 if necessary. After enabling logging, a backup should be done.

The *Disable Operation* (-O d) removes the logging ability from a previously log-enabled file. It turns off the log enable flag and clears the generation number to zero (0) in the .XD header of each ICISAM file, and deletes the corresponding log file. Attempting to disable logging for a file which does not exist, for which the log file is not empty, or for which logging has not been enabled generates an appropriate error. ICISAM files will be NOT be downgraded from ICISAM Version 7.20 even if it might be possible.

The *List Operation* (-O l) displays a list of all the recorded operations in a particular .xl file when given a .xl file argument or in all .xl files when given the ICISAM .xd file. The Detail (-D) option can be used to tailor the data that is displayed. For WRITE and REWRITE operations if the -D r option is given the record will be shown with unprintable characters being replaced with the octal "\nnn". Upto 100 characters of the record will be shown. For relative files the key will also be shown. For DELETE, UNDELETE, and PURGE operations the key can be requested with the -D k option. The Detail option can be used to specify what is listed. To only get a summary of all operations use the -D S option.

The *Next Operation* (-O n) increments the generation number in the .XD file header of each ICISAM file, and creates the corresponding next generation log file. Attempting to generate the log for a file which does not exist or for which logging has not been enabled generates an appropriate error. The older generation of the log file may correspond to the previous backup and can itself be copied to backup media - it would be essential to recovering from that previous backup should the latest backup prove to be unrecoverable.

The *Apply Operation* (-O a) applies a log file or a multi-generation set of log files to its corresponding parent file. The Apply operation will automatically determine which log file generation and where in that log file it must start reading to bring the current ICISAM file up to date. The Apply operation will apply ALL necessary operations from all generations to make the ICISAM file match the latest generation log file.

The *Check Operation* (-O c) opens the ISAM file and checks its log header status. This can be done while the file is in use.

The *Test Operation* (-O t) fully opens the ISAM file and its log file to ensure that the two match. This requires exclusive access.

While running with Local Logging, all modification operations are written immediately to the local .xl file. This is the slowest logging mode since all operations must be logged immediately to the log file.

For Network Mode

With a Mirror

An *Enable Operation* is done with the Mirror option (-M mirror-server) specified. The specified file is set to logging and a mirror copy of the data and log are created on the remote mirror-server specified by the Mirror (-M) option. To perform this operation the user will be logged on to the remote server via ICNETD to start the logging server (iclogs). The current file will be copied over to that machine and a matching log file created there. Subsequent operations to the main file will cause the mirror file and mirror log to be kept updated in addition to the local log file. If needed a username/password will be requested or can be specified on the command line.

The *Check Operation* (-O c) opens the ISAM file and checks its log header status. This can be done while the file is in use. In Mirror mode it checks that the mirror server is still connected to the master file.

The *Test Operation* (-O t) fully opens the ISAM file and its log file to ensure that the two match. This requires exclusive access. In Mirror mode this actually opens the mirror files across the network and ensures that all files are in sync.

The *Promote Operation* (-O p) is used to promote a mirror file to be a master temporarily. When promoted to a master, the remote mirror can be used as the primary file and the matching log file will be kept up to date with log information. When the main machine is again available, a *Recover operation* should be done on that machine to reset its file to be the main file and un-promote the slave. The promote operation must be executed on the remote mirror machine

The *Recover Operation* (-O r) is used to recover the master machine back to being the master after a promote operation was done on a remote mirror OR when the mirror file and its log are more current then the main file. The recover operation must be executed on the master machine when the remote machine is again available to be the remote mirror.

The *Fixup Operation* (-O f) is used to fix a corrupt or invalid mirror when required.

With a Remote Log

An *Enable Operation* is done with the Remote option (-R remote-server) specified. The specified file is set to logging and a remote log is created on the remote server specified by the Remote (-R) option. To perform this operation the user will be logged on to the remote server via ICNETD to start the logging server (iclogs). A matching log file will then be created there. Subsequent operations to the main file will cause the remote log to be kept updated in addition to the local log file. If needed a username/password will be requested or can be specified on the command line.

The *Check Operation* (-O c) opens the ICISAM file and checks its log header status. This can be done while the file is in use. In Remote mode it checks that the remote server is still connected to the master file and log operations are still being sent.

The *Test Operation* (-O t) fully opens the ISAM file and its log file to ensure that the two match. This requires exclusive access. In Remote mode this actually opens the remote log file across the network and ensures that it matches the local log file.

The *Synchronize Operation* (-O s) is used to re-synchronize the local and remote log files if they get out of date.

The *Fixup Operation* (-O f) is used to fix a corrupt or invalid remote log when required.

Running in Network mode is slower than non-logged files BUT much quicker than Local mode as the logged operations are written to the network first and then to the local log file but NO immediate operation is performed. So with this in mind the Remote mirroring mode is the recommended mode as it provides the best available of a "available file" since the mirror is always kept in sync with the master.

When using the Test operation:

A) An error like:

    Error: Connection refused (oserr=10061): Opening remote log on server xxxxx

or

    Error: Connection refused (oserr=10061): Opening mirror on server xxxxx

indicates that ICNETD is not running on the remote machine.

B) An error like:

    Error: Device timeout: Opening remote log on server xxxxx

or

    Error: Device timeout: Opening mirror on server xxxxx

indicates that a network connection to the remote machine is not available.

When either Remote logging and/or Mirroring has been disengaged i/o operations will continue with only simple logging in effect.

An ICSTAT, ICCHECK, or an ICLOG Check on the file will show that the remote is out of sync. To bring the remote log file and/or mirror back to use a -O f (fixup) must be done.

Upgrading

The *Upgrade Operation* (-O u) is provided to upgrade a revision 1 file (i.e., 7.11 ICISAM file) that has simple logging enabled to a revision 2 file (i.e., 7.20 ICISAM) file with simple logging. Please see the last section in this chapter for the differences between revision 1 logging and revision 2 logging.

Basics

The basic operation using logging can be shown in the following schematic:

```
enable-logging-for-needed-files                            (iclog -O e ...)

        ┌──────→backup-these-files (*.xd and *.nx)
        │
        │    run ──→ crash ──→ load-from-backup
        │      │                apply log(s)               (iclog -O a)
        │      │                ready to run again (in sync)
        │      │
        │      │ make backups as          increment generations (iclog -O n)
        │      │ needed                     as needed
        │      │
        │      │←─────────────────────┘
        │     OK
        │   finished running
        │      │
        │      │
        └──────┘
```
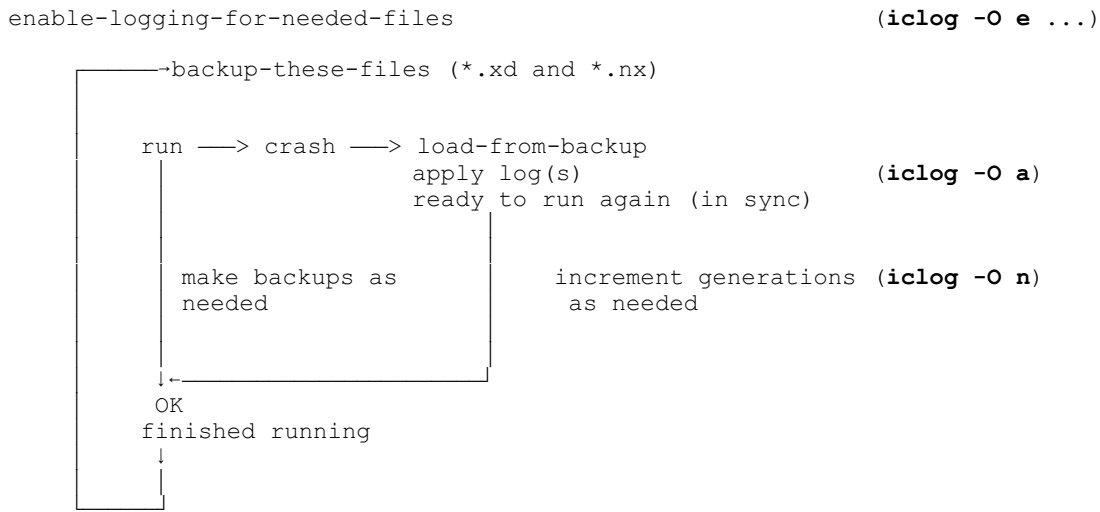
When using Network mode from Windows to UNIX, the Windows drive letter will be mapped from {letter}: to {letter}_. The drive letter will always be in lower-case. The rest of the filename will be in the case presented to the Windows system.

When using Network mode from Windows to Windows and the root directory is set, the colon (:) will be replaced with an underscore (_).

When first creating a remote log or a mirror remember that the user must be able to log on the specified remote machine. If an error such as below is given:

Error: The network path was not found: Creating remote log file <xxxxx> on server <yyyyy>

then the username/password was no available to be logged on the remote machine. The Password option (-P *password)* and/or the Username option (-U *username*) can be used to set the proper combinations and consider using the Save option (-s) to save that combination.

D. **Overview**

In ICISAM Version 7.20, ICISAM implements two modes of logging file modification operations. Both are built on a simple log file per ICISAM file model in which all modifications to the data file, as they are successfully completed, are recorded in the log file by appending records which comprise the relevant operation information. Management of the log files is controlled by a single utility program providing log initialization, information, roll-forward, and clearing functions.   All logging can be initialized and enabled (or disabled) as desired completely invisible to the functioning of the application on a per file basis.

Local Mode

The first mode of logging (a.k.a."Local mode or Local-logging") is that all files are local to a single machine. In the event of a system failure, each file can be restored from a backup copy and all modifications made to it since the backup can be applied.

Local mode is implied with an Enable operation (-O e) when neither a Mirror (-M) or Remote (-R) option are given. The Location (-L) option is available to specify that the log file should be in a separate directory. All components (.nx, .xd, .xl) of the file must be on the same machine.

Network Mode

The second mode of logging (a.k.a. "Network mode or Network logging") involves placing some or all files on a remote server. If only the log portion is placed on the remote server (a.k.a. "Remote logging") then a copy of the log file is written via a remote log server (iclogs) to a remote server. If both the ICISAM file and the log are placed on the remote server (a.k.a. "Mirroring" then a complete set of files are kept updated on a remote server. Both Remote Logging and Mirroring can be enabled at the same time provided two different servers are specified.

Network mode is specified by using either or both of the Mirror (-M) or Remote (-R) options when logging is enabled (-O e)are given. The Location (-L) option is available to place the log file in a separate directory.

The ICISAM File

A log enabled flag defined in the .XD file header is the means by which an otherwise normal open of the file by an application will cause the log file to be automatically opened and all modifications be logged to it.

The Log File

The log file (.xl) is a standard-header file maintained either in parallel with the index (.nx) and data (.xd) portions of an ICISAM file for which logging has been enabled. It is automatically and internally opened by ICISAM when an application program opens a logging-enabled ICISAM file. As each modification to the file is successfully completed, a log record containing all the information necessary to replicate the operation is appended (with unbuffered writes) to the log file. The log file is synchronized to make sure the disk copy is updated after each log record is written.

An empty log file contains no logged modification operation records, it is comprised of the standard file header and the log file (.xl) specific file header only.

All log files (.xl) contain a generation number in their file name corresponding to the generations from the initial logging.

A log record includes a header followed by operation-specific user and/or system data. For all typical modification operations logged, the record header specifically includes an operation code and options, the length of the primary key value associated with the operation, the length of the data record associated with the operation, and various other data such as user-id, computer-id, program-name, etc.

Write Operation

The log record for a write operation is comprised of the header, the primary key value if it is outside the record, and the user record data. The header contains the operation code for the write operation, options, the length of the primary key value (zero if not applicable), the length of the user record data.

Rewrite Operation

The log record for a rewrite operation is comprised of the header, the primary key value if it is outside the record, and the user record data. The header contains the operation code for the rewrite operation, options, the length of the primary key value (zero if not applicable), the length of the user record data.

Delete Operation

The log record for a delete operation is comprised of the header and the primary key value. The header contains the operation code for the delete operation, options, the length of the primary key value.

Undelete Operation

The log record for an undelete operation is comprised of the header and the primary key value. The header contains the operation code for the delete operation, options, the length of the primary key value.

E. **Automatic Logging**

The following ability is provided to allow for newly created ICISAM files to have logging automatically enabled.

ICRUNLOGOPTS is an environment variable sought by the various runtimes (icrun, icrunw, icthins, icrunrs, icruncgi) to allow isam files that are created under program control to have logging enabled. This is NOT supported for files created via ICNETD, i.e., @.....filename.)

ICRUNLOGOPTS can include any of the following:

```
-L log-path
-M remote-mirror-server
-O e                            (implied, if not specified)
-R remote-log-server
{-X exclude-path}...            (upto 16)
```

If a new isam file is created and it does NOT match the *exclude-path*, then it is created with the provided log options.

F. **Examples**

Example 1

To enable "Local-logging" for some files:

```
iclog -O e file1 file2 file3...
```

Where file1, file2, file3, ... are standard ICSAM files of at least version 7. The log files file1.1.xl, file2.1.xl, and file3.1.xl will be created in the current directory for these files.

As these files are opened and modified all the modifications will be written to the local log(s).

Example 2

To enable "Network-logging" in Mirror mode:

```
iclog  -O e -M redhat4  file22
```

G. **Differences with Revision 1 Logging** (Pre 4.00)

File-set logging is NOT supported in 4.00 and greater.

Generation numbers for the log file are kept as part of the file name. (I.e., joe.1.xl, joe.2.xl)

Logs can be applied at any time to a logged file. The apply operation will automatically determine what records must be written to the backup file. In addition, this allows for backups to be taken at any time.

Additional detail is available in the logged record itself that can be viewed via the listing operation.

Records written to the logged file itself are now compressed resulting in significant savings on disk.

Revision 1 logging files are not supported in 4.00. An Error "Revision 1 logging is not supported" will be given. Use ICLOG -O u to upgrade revision 1 logging-enabled files.

Previous versions of ICOBOL will detect 4.00 Revision 2 logging as an incompatible revision with an error of:

"File does not have the correct revision".

Revision 2 logging shows up as a 7.20 revision file.

H. **Additional Notes**

Optional Network-Logging capabilities

Requires a new license (ICLOGS) for network server log servers that run under ICNETD on the remote server. (iclogs) (Requires ICPERMIT 4.00+ and ICNETD 4.00+)

1. Provides for remote logging of the .xl file.

    (-R remote-log-server)

In Remote mode, both a local log file and a remote log file are written.

    iclog -O e -R redhat80 myfile1

(This enables logging both locally and remotely on machine redhat80 for file myfile1.)

2. Provides for the recovery of the main file from the remote log. (Requires exclusive access to the master file to recover.)

    iclog -O s

When recovering from the remote log, the progress switch can be given to allow the user to see that recovery is in progress as this may take a few moments for large files.

In this mode, the master isam file and its local log are synchonized from the remote log.

This can happen if the main server crashes with buffers in memory. The remote log is available to restore those records. Or if backups must be loaded on the main server the remote log will be used to apply the needed additional operations.

An additional use of this option is to copy the master file and its local log to another machine and occassionally run the Sync operation against the remote to keep an additional warm-backup machine.

The Sync operation works with the Remote log to bring a backup file up to date.

3. Provides for mirroring of the isam file along with its log on a remote machine.

    (-M mirror-server)

When enabled, the current ICISAM file and its log is copied to the mirror-server machine. On the mirror-server machine, the Mirror isam file can NOT be opened for updating but can be opened in read-only mode. For example, to do queries.

In Mirroring mode, the mirrored isam file (along with its log) is kept up-to-date.

    iclog -O e -M intel8 myfile1

(This enables logging locally and mirrors the isam files (xd/nx) and a log file on intel8 for file myfile1.)

4. Provides for the use of the mirror file as the primary because the main server has failed. Requires exclusive access to the slave mirror to enable.

    iclog -O p   (run on the mirror-server, mirror file)

Promote the slave file to a "promoted" master for the short-term. Applications can be restarted on the slave-server until the main server is restored to operation.

To return to using the main server, use the recover operation. Requires exclusive access to the master file to recover.

    iclog -O r   (run on the main server)

This will update the main file with updates from the promoted slave mirror and then restore the promoted slave mirror to act just as a slave again.

The Recover operation can also be used to bring a master back in sync with its mirror.

5. To check that a file is still being run in remote or mirror mode use the check Operation.

    iclog -O c

This scans the local file and, for any logged file that has remote or mirror set, it will check whether the Remote and/or Mirror status in the local header is still valid. If not, a message is displayed and a non-zero exit code is returned.

The status is not valid when the remote log and/or mirror is no longer in sync with the local file and an ICLOG Fixup operation is required.

6. To test that a remote log and/or mirror file is accessible use the Test operation.

    iclog -O t

Scans the local file and for any logged file that has remote or mirror set and checks that the remote log and/or mirror file can be opened and is valid. If not, an ICLOG Fixup operation is required.

If the file/files cannot be opened a non-zero exit code is returned.

7. To fix a remote log or mirror file when the status flag(s) are set. (See M. above.) Requires exclusive access.

    iclog -O f

Will fix a remote log and/or mirror file that is out of sync with the master file. If the remote log or mirror does not exist it will be created as long as the needed directory structure is present.

When using Network mode, either Remote logging and/or Mirroring there is an additional environment variable available for the runtime:

ICRUNLOGTIMEOUT

ICRUNLOGTIME provides the number of tenths of seconds a network request can be outstanding before the logging operation is suspended and remote logging/mirroring is disengaged. A message similar to:

    Error: Device timeout: processing mirror for <filename>

or

    Error: Device timeout: processing remote log for <filename>

will be written to the audit file but the program will continue to execute with no errors. The default is 150 or 15 seconds. It takes two cycles for this to take affect so a 15 second setting would end up taking 30 seconds before the operation will continue.

Both Remote logging and Mirroring can be done at the same time but the two servers must be different from each other.

    iclog -O e -R redhat80 -M intel8 myfile1

In both cases, if the remote connection goes down and the main server keeps running, then the local .xl file will continue to be up-dated. The remote log and/or mirror will need to be re-sync'ed when the connection is again available by using the ICLOG Fixup operation.

The ICSMVIEW utility can also be used to view this case by using the -O i and -v operation and looking for a re-sync flag.

If the main server goes down, then:

    In the remote logging case, the remote log gets applied to the backup file and applications can run on the remote server.

    In the remote mirroring case, no log needs to be applied, the application can run on the remote server after doing a "break mirror". (-O p) promote.

    Once the main server comes back on line the applications can be rerouted to it by using the -O r and -O s operations to re-mirror and re-sync the main files from the remote files. (-O r must be done for -M case, and for the -M and -R case both a -O r and -O s must be done).

For just a small performance penalty the mirror case is probably more useful than just the remote log case.

IX. ICMAKEMS

A. **Introduction**

The ICMAKEMS utility is used to convert a text version of system message files into the internal format used by all of the Interactive COBOL programs (command-line utilities, compilers, and runtime). These files, *.ms, will be used to replace the default internal messages. These files can be changed to accommodate languages other than English by editing the released sample English message files and building the respective .ms file.

Any customized message file should use the ICCONFIGDIR mechanism to find the file.

B. **Syntax**

The standard syntax is:

```
icmakems [-a[:aflag]|-A file|dir[:aflag]] [-h|-?] [-q] [-O dir]
    [argument]...
```

Where
  -a[:*aflag*]|-A *file*|*dir*[:*aflag*] (Audit)
    Enables auditing (default icmakems.lg). Where *aflag* is a|b|d|p|t|u|da|db|pa|pb|ta|tb|ua|ub, defined as
    a-append, b-backup, d-date, p-pid, t-time, and u-username.
  -h|-? (Help)
    Displays help text.
  -q (Quiet)
    Enables quiet operation.
  -O *dir* (Output)
    Place the output in the indicated directory *dir*. If the output file already exists, the new messages replace
    the old ones.
  argument
    Specifies a filename, a list of filenames, or a template.

Environment variables:

ICMAKEMS	Command line options
ICTMPDIR	Temporary files

C. **Description**

This utility is not required if the default set of messages are sufficient. The text files (coberrs-en.h, runerrs-en.h,and syserrs-en.h) in the messages directory show the default messages in English. Also found in the messages subdirectory is the file infostat.ms which is a special message file used by the IC_INFOS_STATUS_TEXT builtin. It is built from infostat.txt.

The content of the input files for ICMAKEMS are very simple .h files. Please see their descriptions within the samples provided.

syserrs	messages are the standard exception values provided by the runtime.
runerrs	messages are runtime and ICEXEC startup messages
coberrs	messages are compiler messages

The builtin IC_MSG_TEXT can be used to retrieve messages from within a COBOL program.

These files can be used to build a new message file (*.ms). For example:

```
icmakems syserrs-new.h
```

Will build a systerrs-new.ms file which can be renamed to syserrs.ms and it will be read at startup to replace the default entries.

Any customized message file should be placed in the directory specified by ICCONFIGDIR.

X. ICNETD

A. **Introduction**

ICNETD provides the TCP/IP network communication and security handling for the server side of the client/server operations. These include the file i/o server (icios), the ThinClient server (icrunrs), the remote logging server (iclogs), and the Remote ISQL server (icsqls) for **ICOBOL**. ICNETD is available on UNIX and Windows. Clients that use the i/o handler (icios) will generally be referred to as "ThickClients".

A TCP/IP network must be running between the client and server machines with the necessary daemons and drivers. The ping utility can be used from both the client and the server to check that each can access the other with the given machine name or IP address. For Windows clients, this means that TCP/IP networking and a WinSock 2.0 or greater compliant driver must be installed.

ICNETD I/O client/server (icios) is required in order to share files between UNIX machines and Windows machines and between two or more UNIX machines. (Sharing files via NFS mounted file systems is NOT supported or recommended.) The ICNETD I/O client/server (icios) is a performance enhancer when sharing files between two or more Windows machines.

The ICNETD I/O client access can be used over the Internet to access remote files. On the server machine, if a firewall is being used, the necessary port (default 7333) must be opened.

ICNETD is required on the server when ThinClient support is needed.

I/O Client (thickclient)

I/O Client support is offered by the UNIX and Windows runtime systems, the ODBC driver (ICODBCDR), and the user library. This is generally referred to as thickclient mode. The I/O client/server model differs from the traditional **ICOBOL** support for remote file access in that it acts at the COBOL operation level rather than the operating system operation level. In other words, it remotely reads and writes records rather than disk blocks. For complex files, like indexed files, this generally provides enhanced I/O performance in the network environment while reducing network traffic.

The I/O client (icios) requires a separate **ICOBOL** Network Server License and count in the license description file that ICPERMIT manages in order to service clients in i/o client/server mode, no runtime licenses are used in this mode. A license is required for each connection (icios process).

ThinClient

ThinClient client (icrunrc) support is offered on all machines and gui support (sp2/qpr) is available under Windows. In the thinclient cases, only a small part of the code is on the client machine. Just enough to display the provided information and provide keyboard input support.

When providing ThinClient (icrunrc) support, both an **ICOBOL** Runtime license and an **ICOBOL** Remote Runtime client license must be available to start the ThinClient server (icrunrs). In addition, if any gui is to be used (sp2 or qpr) then an **ICOBOL** SP2RUN license will be required. ThinClient is similar to telnet support but is done all in **ICOBOL** space via an encrypted interface with the additional support for the gui components of sp2 and qpr and an automatic reconnection ability.

On Unix, when using ThinClient client, ICRUNRC, and connecting to a UNIX machine (thru ICNETD), the basic environment variables normally set by UNIX logon are set by ICNETD. These are: HOME, LOGNAME, MAIL, and SHELL. The MAIL entry is set only if ICNETD is given the default MAIL path by using the ICNETD_MAIL environment variable to provide the path to the standard mail directory, to which ICNETD will add the username.

So if ICNETD_MAIL is given: /usr/spool/mail/ then when the user "joe" logs on via a ThinClient, then the MAIL environment variable will be set to "/usr/spool/mail/joe".

The SHELL entry will only be set if the shell value is provided by the passwd file.

ICLOGS client

The ICLOGS client is any remote process that opens a file for modification that has remote logging enabled   An ICLOGS license must be available on the server in order for the ICLOGS server process to start.
    .

ISQLClient

An ISQL client is a runtime using ISQL.

When providing ICSQL support, an **ICOBOL** Remote ISQL license must be available to start the ISQL server (icsqls). The ISQL server communicates with the ODBC Administrator and the ODBC Driver manager only on the remote machine.

B. **Syntax**

The standard syntax is:

```
icnetd [-a[:aflag]|-A file|dir[:aflag]] [-d] [-h|-?]
   [-M machine[:port]|:port] [-N bp] [-O a|b|c|h|m|p|s|t] [-q] [-R rootdir]
   [-s] [-S {a|t}:{on:off}] [-t]
```

Where
    -a[:*aflag*]|-A *file*|*dir*[:*aflag*] (Audit)
        Enables auditing (default icnetd.lg). Where *aflag* is a|b|d|p|t|u|da|db|pa|pb|ta|tb|ua|ub, defined as a-append, b-backup, d-date, p-pid, t-time, and u-username.
    -h|-? (Help)
        Displays help text.
    -d (Debug)
        Run in debug mode, no daemonization. (On UNIX only.)
    -M *machine*[:*port*]|[:*port*] (Machine)
        Specifies the remote machine and/or TCP service port address for ICNETD. *Machine* defaults to localhost if not specified. *Port* defaults to 7333 if not specified.
    -N bp (No options)
        Specifies NO options. Valid NO options are:
            b (No-logon-as-batch)   Allow logons to the icios server without the logon-as-batch privilege (On Windows only)
            p (No-password)   Allow logons without passing a password (On UNIX only)
    -O a|b|c|h|m|p|s|t (Operation)
        Specifies an operation to perform. Valid operations are:
            a (Amplify)   Amplify daemon tracing
            b (Boost)   Boost (amplify) server tracing
            c (Check)   Check to see ICNETD is already running
            h (Hush)   Hush (mute) server tracing
            m (Mute)   Mute daemon tracing
            p (Post)   Cause connection information to be written to the log file
            s (Start)   Start ICNETD   (On Windows only)
            t (Terminate)   Terminate ICNETD.

-q (Quiet)
>   Enables quiet operation.

-R *rootdir* (ROOT)
>   Specifies the effective root directory on the machine to which thickclient remote users have access. Default is "/" on UNIX and "current-drive:\" on Windows. Only used by the icios server.

-s (Service)
>   Service indicator. On Windows only, and is required when running as a service.

-S {a|t}:{on:off} (Server)
>   Server options:
>>   a-audit (per processing)
>>   t-tracing can be enabled and disabled.

-t (Trace)
>   Enables tracing to allow debugging.

Environment variables:

ICNETD	Command line options
ICTMPDIR	Temporary files
ICPERMIT_MACHINE	Remote machine for server licensing

On Windows, ICNETD can also be stopped and started manually by going to the Services selection under Control panel. New arguments can also be provided by specifying startup parameters. If given, the new startup parameters replace the stored parameters for this single invocation. On installation, ICNETD is installed as a service using the ICSVCMGR utility. More on ICSVCMGR can be found in the Installing and Configuring on Windows manual. ICSVCMGR can be used to change the default command line if needed.

On termination, ICNETD outputs a table of connection information to the log file.

C. **Description**

On UNIX, you must be super user to start the ICNETD server and to use the "amt" Operate options. On Windows, you must have administrator privilege to start/stop the ICNETD service and to use the "amst" Operate options.

ICNETD -O c (check) can be done by any client to see if an ICNETD daemon is running. The Machine switch can be used to check a remote machine.

From the operating system standpoint, a TCP/IP network must be running between the client and server machines with the necessary daemons. The *ping* utility can be used from both the client and the server to check that each can access the other with the given host-name.

ICNETD is provided as part of the Interactive COBOL media but requires separate licenses for the various services that it offers. **ICOBOL** I/O Server license and user count (on the server machine) in the license file that ICPERMIT manages in order to fork I/O servers (icios) for client/server file access, ICNETD does not use runtime licenses in this mode. When providing ThinClient (icrunrc) support, runtime and remote client licenses must be available to start the ThinClient server (icrunrs) and an sp2runtime license must be available if sp2 or qpr are to be used. When providing Network logging support, an ICLOGS license must be available. When providing remote ICSQL support, an ICSQLS license is required.

On UNIX, the startic and stopic scripts support starting and stopping ICNETD. On Windows, the installation program will install ICNETD as a service if requested, otherwise the ICSVCMGR can be used to control ICNETD.

On the server system, ICNETD must be running to provide the initial server connection to the client. When ICNETD starts, it registers itself with TCP/IP as a server listening for connections on a particular TCP/IP port. The default port is 7333, but this can be changed from the command line. When a client opens a TCP/IP connection on this port, ICNETD is notified of the connection and the type of request (either ThickClient, ThinClient, Logging

Client, or remote ICSQL client) being made. ThickClients are from remote runtimes, ICODBC drivers, or user library applications that need client/server file support while ThinClients are specialized front-ends on the clients that provide only a very small interface handler. ICNETD then tries to login the given username/password. On Windows, the provided username must have the "logon as batch" privilege for icios and the "logon locally" for icrunrs and icthins. On Windows 2000, the password cannot be blank. Up to three attempts will be made before an error is returned to the application. If no such user is found, an exception 309 "Network path was not found" is usually given. While making these attempts, username, password, and domain will be prompted for on the client machine. Once logged in, the appropriate server (icios for ThickClient support, icrunrs for ThinClient support, iclogs for remote logging or icsqls for remote ICSQL support) is started and the appropriate license(s) are requested (a Network Server license for ThickClients, both an **ICOBOL** runtime and Runtime Remote Client license for a ThinClient, and then possibly a sp2runtime license if sp2 or qpr are to be used, iclogs or icsqls license as appropriate. From this point on, the server process will handle all client requests. If ICNETD is not running when a connection is attempted, the client will usually receive an exception 315.

Each server process acts just like that user with all the same access controls and privileges. Icios servers will remain until the process that requested the service terminates. (I.E., even if all opened "remote" files are closed such that there are no open across the connection the server will remain.) This provides a performance boost for closing and then re-opening files in an application at the expense of keeping the process and license in use.

On UNIX, the No-password option (-N p) can be used to not require a password. On Windows, if the username has a password it must be provided. If a username/password is prompted for that username/password pair can be saved for future logons. Just select the appropriate box in the username/password dialog box. The username/password/domain will be saved in the registry for each ip address for Windows clients and in a file with the name .icnet.<ip-address> is written to the user's home directory (as given by the HOME environment entry) for UNIX clients if the Save option is selected on the username/password prompt screen. On Windows, the user profile must have the "logon as Batch" privilege enabled to use icios servers and the "logon locally" privilege must be enabled to use icthins and/or icrunrs servers.

The following ICNETD servers (icios, icrunrs, and iclogs) require ICEXEC to be running. The ICSQL server (icsqls) does NOT require ICEXEC to be running.

> NOTE: ICNETD servers use processes as well as file resources from ICEXEC, thus the process count and other system parameters in the system configuration must take this into account.

When ThickClient or ICLOGS clients access files through ICNETD, the file pathnames are relative to ICNETD's effective root directory. By default, this is the actual root of the server file system. It may be desirable for security reasons to limit remote users to a subset of the server file system. This can be done by using the -R parameter when starting ICNETD to change its effective root. It will prefix the filenames from the client with the subdirectory from the -R parameter before opening them. Thus, if -R is set to "/remote/files" and the client opens "/ar/customer", the server will open "/remote/files/ar/customer".

The Rootdir switch (-R *rootdir*) instructs ICNETD to always prepend the *rootdir* to any name passed to the ICNETD server for ThickClients. (Icios)

If no Rootdir is given, all filenames start at the root. (On Windows, the root of the current-drive.)

On Windows, to access multiple drives no Rootdir must be specified and the appropriate drive must be given by the COBOL program (i.e., "@//machine10/D:/test/file").

When server tracing is enabled, each server generates its own log file (icios_*<pid>*.lg, icthins_*<pid>*.lg, or icrunrs_*<pid>*.lg) in the **ICOBOL** working directory in addition to the log file generated by the server (ICNETD). *Pid* is the pid number for the server process.

For ThickClients, each ICNETD I/O server requires an ICNET license.

Another item to note, when opening files ASSIGN'ed to PRINTER, if the filename is an ICNETD remote file it is NOT placed in the local printer control file (.pq). If the ICNETD server has an ICEXEC running with PCQ's enabled, it will be placed in that printer control queue (.pq) file. The console number for the entry will be set to -1.

For ICLOGS clients, each server process requires an ICLOGS license. .

<u>On Windows</u>, ICNETD servers cannot accessed mapped drives since they were started from a service which never loaded that mapping.

D. **Use**

ThickClient

<u>Icios</u>

The ThickClient client accesses files on the server by using a special network filename. The syntax of this filename uses the special leadin character that is also used by logical device names followed by a standard Internet Uniform Resource Locator (URL). The syntax is as follows:

```
@[icnet:]//machine[:port-address]/path
```

Where
> *machine*
>> Is the remote machine name or IP address of the machine on which you wish to access files.
> *port-address*
>> Is the TCP service port on which ICNETD is listening on the remote machine instead of the default (7333).
> *path*
>> Is the filename, including any directory specifiers, to the file on the specific machine.

The *machine* is often a simple name on a local area network, e.g., "accounting". It can be a full internet name on a wide area network, such as "accounting.envyr.icobol.com", or an IP address, such as "166.82.100.101". The naming used will depend on how your network is configured.

As mentioned above, the *path* supplied will depend on whether ICNETD has been configured with an effective root or not. In order to access the file "/remote/files/ar/customer" on the "accounting" server, the client would specify the following:

> # ICNETD<enter>   (ICNETD started with the default root)

> @icnet://accounting/remote/files/ar/customer

> # ICNETD -R /remote/files<enter>   (ICNETD started with a new effective root)

> @icnet://accounting/ar/customer

In order to print a queued file on the server using @PCQ0, the client would specify:

> @icnet://accounting/@PCQ0

Nothing except TCP/IP is required on the client system from Interactive COBOL to connect to the server. Client exceptions that can be received trying to connect are:

252 "Program is not authorized to run"	An ICNET I/O Server license is not available on the specified machine.
306 "Network Request not supported"	A revision mismatch between the client and the ICNETD daemon.
307 "Remote Computer is not available"	There is no computer by the given machine name available on the network.
309 "Network path was not found"	The current username is not available (from /etc/passwd) on the remote machine. Access denied.
315 "Unexpected Network Error"	There is no ICNETD running on the remote machine.
323 "Network name not found"	Couldn't set group-id or user-id from /etc/passwd.

On the server system, the ICNETD daemon must be running to provide the initial server connection to the client. It then forks a server ICNETD process that "logs into the given user's account" on that machine. For this reason, all users who access files on a server must have accounts available on that server with matching user names. Each server process acts just like that user with all the same access controls and privileges. If no such user is found, an exception 309 is given. On Windows, ICNETD starts icios servers with the logon_batch option so all usernames on the ICNETD server machine must have the "Logon as a batch job" privilege when using thickclients. To add this user right do the following on the machine on which ICNET is running: Select the User manager. Select Policies and then User Rights. Check the Show Advanced User rights box. Now select the "Log on as a batch job" right and then add the needed groups and/or users.

If the username/password is invalid on the first open to the ICNETD server, the user will be prompted for a valid username/password with a pop-up box. Three(3) attempts will be allowed before an error is returned. ESC will cause the open to fail. Once a valid password is given, it will be remembered for all subsequent connections. If a new username was given, the new username/password pair is not remembered for a new connection, the original username/password pair will be used.

When communicating with a Windows based ICNETD, the following os-errors can be reported in the Logon Failure username/password pop-up dialog box. Use these error descriptions to help solve the problem.

1314	Privilege not held
1315	Invalid account name
1317	No such user
1326	Logon failure (unknown username or bad password)
1327	Account restriction
1328	Invalid logon hours
1329	Invalid workstation
1330	Password expired
1331	Account disabled
1385	Logon type not granted (Need a privilege)

From a client, to check that the ICNETD server is running on a remote machine the following should be done:

```
icnetd -O c -M machine
```

This will work on all clients except Windows 9x machines.

Icios can be set to detect loss of client connection by using the ICNETDUSESHEARTBEAT environment variable from the client side. All runtimes, the user library, and the ic odbc driver support this variable as of 4.40.

Setting ICNETDUSESHEARTBEAT=1 causes a heartbeat thread to be enabled on the client and server to continuously provide a heartbeat across the network. In this mode if icios detects a loss of the hearbeat it will shutdown the icios server process cleanly closing all files. Usually this will happen withing 60-120 seconds.

ThinClients

When icnetd starts a ThinClient server (icrunrs), it will pass the client's ip address in as an environment variable called ICREMOTEADDRESS and the client's host name as ICREMOTEHOST. (Basically a "-E ICREMOTEADDRESS=n.n.n.n -E ICREMOTEHOST=xxxx" on the command line.) These two entries can then be queried from COBOL by using the IC_GET_ENV builtin after determining that a ThinClient is running by doing an IC_TERM_STAT builtin and looking at the two ThinClient flags.

The sample logon program has been updated to show this information in the upper left corner of the main screen, if available.

The ThinClient server (icrunrs) is started by ICNETD and runs the logon program by default. On Windows, the ThinClient server is installed when ICNETD is selected. When the ThinClient server is invoked by ICNETD, it requests both a runtime license and a runtime remote client license from the license manager and then starts the COBOL program logon.cx by default. If sp2 or qpr calls are made by the COBOL program then a sp2runtime license will be acquired at that point. The ThinClient server uses consoles with device set to "machine-name" or ip-address first, then "icrunrs", and finally to (blank). The ICTERM setting is provided by the ThinClient client. Note that all users that attach to ICNETD via a thinclient must have the "Log on locally" privilege when the server is a Windows machine. Also note that on Windows 2000 the password cannot be empty.

On the server ensure that the following are accessible in the current directory or via PATH, ICCCODEPATH, ICDATAPATH, etc:

- cobol object code (.cx files)
- data files

Once the application is running, it will make user interface calls which are intercepted by the ThinClient server library. Some of these calls are processed on the server and some are sent to the client machine for processing. Normally character calls sent to the client will result in a response from the end user. Each ThinClient server (icrunrs) requires a runtime, remote client license, and possibly an SP2Runtime license.

The ThinClient client can only communicate with the ICNETD service 2.80 and up. To work in both screen and gui mode the ThinClient client requires an ICNETD service of 4.00 and up.

To debug ThinClient consider the following:

A. Make sure the program(s) run without ThinClient before moving to ThinClient.
B. With ThinClient
    B.1 On the client, set
    B.2 On the server, set
    B.3 On the server, turn on ICNETD server tracing (icnetd -O b). This will cause icrunrs_(pid).lg files to be created for each icrunrs started. Any **ICOBOL** errors will be logged to this log file.
    B.4 On the server, turn on ICNETD server tracing (icnetd -O a). Provides more logging information in the icnetd.lg file.

If gui support (sp2 and/or qpr) is also to be used then an additional SP2RUNTIME license is required on the server. To run totally in gui mode the logon program sp2logon can be invoked. Gui support is only provided when the ThinClient client is running on Windows.

On the server ensure that the following are accessible in the current directory or via PATH, ICCODEPATH, ICDATAPATH, ICCONFIGDIR, SP2DIR, SP2.CFG, etc:

- cobol object code (.cx files)
- data files
- panel files
- sp2 configuration file

- sp2tc.ini

Once the application is running, it will make SP2 (and FormPrint) user interface calls which are intercepted by the ThinClient (gui) server library. Some of these calls are processed on the server and some are sent to the client machine for processing. Normally sp2 calls sent to the client will result in a response from the end user. Each ThinClient server that uses SP2 or QPR requires an SP2 runtime license in addition to the standard runtime license and remote connection license when using gui calls.

To debug ThinClient with gui consider the following:

A. Make sure the program(s) run without ThinClient before moving to ThinClient.
B. With ThinClient
    B.1 On the client, set SP2DBG=2 to get an sp2dbg.xxx log file. (QPRLOG=1 for FormPrint)
    B.2 On the server, set SP2DBG=2 to get an sp2dbg.xxx log file. (QPRLOG=1 for FormPrint.)
    B.3 On the server, turn on ICNETD server tracing (icnetd -O b). This will cause icrunrs_(pid).lg files to be created for each serverd started. Any **ICOBOL** errors will be logged to this log file. Without this log file, all **ICOBOL** messages are lost.
    B.4 On the server, turn on ICNETD server tracing (icnetd -O a). Provides more logging information in the icnetd.lg file.

More on using ThinClient with gui support can be found in the readsp2.txt file.

E. **ThickClient Examples**

On UNIX, if you start ICNETD with no root directory (-R):

@//server1/usr/joe/data	would access /usr/joe/data on machine server1
@icnet://server1/usr/joe/data	would access /usr/joe/data on machine server1
@//server2/data	would access /data on machine server2
@//server2/@pcq6	would access @PCQ6 on server2

On Windows, if you start ICNETD with no root directory (-R):

@\\server1\user\joe\data	would access \user\joe\data on machine server1 drive C:
@icnet:\\server1\user\joe\data	would access \user\joe\data on machine server1 drive C:
@\\server2\D:\data	would access D:\data on machine server2
@\\server2\@PCQ6	would access @PCQ6 on server2

The ICLINK utility can be used to provide a mapping from filenames in the COBOL program to client/server type filenames. See the ICLINK Chapter for more information.

F. **ICNETD Auditing**

ICNETD provides various logging modes to facilitate debugging.

Initially, a default ICNETD log file will look like:

```
Audit log for icnetd 4.20 (Windows) created Dec-21-2009 09:34:13.00
icnetd Revision 4.20 (Windows)
Copyright (C) 1987-2009, Envyr Corporation.  All rights reserved.
Started without Startup Parameters specified
Options: -A C:\WINDOWS\:b -s -M :7333
Dec-21-2009 09:34:13.80 icnetd (868):          icnetd is ready, listening on: port=7333 on machine RALPHJ
```

When running with ThinClients that happen to disconnect for some reason the following will be added:

```
Dec-21-2009 09:51:49.26 icnetd (868):          Reconnect request from surrogate on pid 2064 for client on
machine ENVYRMOBILE pid 2320
Dec-21-2009 09:51:58.81 icnetd (868):          Reconnect request from surrogate on pid 1676 for client on
machine ENVYRMOBILE pid 2884
```

These could reconnect or not as:

```
Dec-21-2009 09:53:49.60 icnetd (868): Warning: Reconnecting surrogate terminated (Exit Code = 0): pid=2064 for
client on machine ENVYRMOBILE pid 2320
Dec-21-2009 10:58:41.11 icnetd (868):          Reconnecting client on machine ENVYRMOBILE pid 2884 to surrogate
on pid 1676
Dec-21-2009 10:58:41.29 icnetd (868): Error:   Reconnecting from 75.251.64.189:57346
Dec-21-2009 10:58:41.29 icnetd (868): Error:   The reconnection key does not match any connection: Processing
reconnect
```

In one case (surrogate 2064) you will see that the surrogate terminated, in this case ICREMOTETIMEOUT had been set and it timed out.

In the second case (surrogate 1676) no timeout was given and the remote client reconnected.

The two error lines are when the thinclient client tried to reconnect with the previously terminated surrogate (2064).

Additional lines in the log can be shown by making an ICNETD -O p (post) call. This will cause ICNETD to dump its current connection table as such:

```
Dec-21-2009 11:05:34.11 icnetd (868):          Connected to server (command request) 127.0.0.1:2933
Dec-21-2009 11:05:34.11 icnetd (868):          Post current connection data:
 Client........ ............. .... ........ ............. ....... .... Surrogate  Date and Time
 Computer Name  IP Address    Port     PID User Name     Program Rev      PID Connection Established  Recon
 -------------- ------------- ----- ------- ------------- ------- ---- -------- ---------------------- -----
  ENVYRMOBILE   75.251.64.189 57347    2828 Ralph Jordan  icrunrc 4.20     3652 Dec-21-2009 11:03:09.00   No
  ENVYRMOBILE   75.251.64.189 57345    2884 Ralph Jordan  icrunrc 4.20     1676 Dec-21-2009 09:50:23.00   No
  RALPHJ        192.168.2.105  2930     696 ralph         icrun   4.20     1800 Dec-21-2009 11:05:23.00   No
 -------------- ------------- ----- ------- ------------- ------- ----- -------- ---------------------- -----
```

Note the final column of Recon. This column will be set to "Yes S" if the surrogate has requested a reconnection or an "Yes C" if the client has requested to reconnect.

By adding additional auditing options this log can be used to show all the actual requests and even the command lines used to start the various surrogates. ICNETD -O a, ICNETD -O b, etc. All of these operations/options can be set with either the command line directly when starting ICNETD or as operations from a command ICNETD. These should only be used for trace and debugging purposes as they will create a much larger log file.

XI. ICPACK

A. **Introduction**

The ICPACK utility tailors the structure of the .NX portion of an ICISAM file to allow for optimized storage and access time. ICPACK works with the current .XD/.NX portions of a file to build a new .NX file to replace the current one.

ICPACK does not remove logically deleted records, use ICREORG. ICPACK does not reclaim space from physically deleted record slots, use ICREORG for that purpose.

B. **Syntax**

The standard syntax is:

```
icpack [-a[:aflag]|-A file|dir[:aflag]] [-C density] [-h|-?] [-I cnt]
    [-K key:density]... [-N i[:pct]] [-p] [-q] [-D cnt | -R cnt]
    [argument]...
```

Where
    -a[:*aflag*]|A *file*|*dir*[:*aflag*] (Audit)
        Enables auditing (default icpack.lg). Where *aflag* is a|b|d|p|t|u|da|db|pa|pb|ta|tb|ua|ub, defined as a-append, b-backup, d-date, p-pid, t-time, and u-username.
    -C *density* (Compaction Density)
        Sets the default packing density for any unspecified key. If not set, the default is 99%.
    -D *cnt* (Data padding)
        Pad the data files with at least *cnt* 2048 byte blocks of free space. This may cause the file to grow in size. This switch may not be used with the Record padding switch.
    -h|-? (Help)
        Displays help text.
    -I *cnt* (Index padding)
        Pad the index file with *cnt* 2048 byte empty blocks of free space. This may cause the file to grow in size.
    -K *key*:*density* (Individual key density)
        Provides the density percentage values for each of the file's specified keys to be packed. If more than one density is the same, you can use the Compaction Density switch (-C) to set a default and then specify only those that are different. The *density* must be a number between 50 and 99, inclusive. The *key* value can be p for the primary, a1 for alternate key 1, a2 for alternate key 2, . . . up to a16 for alternate key 16. If a particular file has fewer keys than specified in the density-switches, the extra values are ignored. If density is not specified for a particular key, it defaults to the value given by the Compaction Density switch (-C).
    -N i[:*pct*] (No-packing)
        Do not pack the file. If the optional *pct* is added, the No-packing switch is conditioned on the percentage of space that would be freed: No-packing if less than *pct* amount of space is freed up. This switch is implemented based on calculations, not scans.
    -p (Progress)
        There is an ongoing display of the processing, which typically runs much slower for smaller files. The reporting interval is for every 1% of the file processed.
    -q (Quiet)
        Enables quiet operation.
    -R *cnt* (Record padding)
        Pad the data file with at least *cnt* records of free space, rounded up to a 2048 byte boundary. This may cause the file to grow in size. This switch may not be used with the Data padding switch.
    argument
        Specifies the filename or template for the files to be packed. If not given, all ICISAM files in the current directory are processed.

Environment variables:

    ICPACK   Command line options
    ICTMPDIR   Temporary files

C. **Description**

ICPACK requires temporary disk storage in which to build the new .NX file and only at the very end is the current .NX file deleted and replaced with the new .NX file. These temporary files have the `.nt' and `.dt' extensions for index and data respectively. This insures that if the machine crashes while ICPACK is running, the original file is not damaged in any way. <u>On UNIX</u>, because ICPACK installs the new .NX file with a rename, all hard links to the .NX file will still point to the old version, i.e., when finished all hard links to the new .NX file will have been removed. Also the file attibutes of the .nx file could be changed from the .xd if the *umask* is different.

ICPACK will not run on a file if either reliability flag is set (i.e., the file is corrupt). Run the ICCHECK utility on the file to make sure it is corrupt. If corrupt, the ICFIXUP utility can be used to build a new ICISAM file or a backup should be used if the .XD file is badly corrupted.

ICPACK will pack a particular key index structure from 50% to 99% full. The more dense the packing (i.e., the larger the percentage) the less storage the key structure takes and the faster a keyed access is performed. On the other hand, adding many new keys to a highly packed file will cause index node splitting to occur and may be slower.

ICPACK removes purged alternate keys.

ICPACK can reduce the size of the .NX file by up to 50% for files that have been randomly written and have not been packed.

ICPACK opens the file(s) with the exclusive option.

The following information about an ICISAM file is given:

    - A possible Warning message that the end-of-file of the .XD file is not on a 2048 byte boundary
    - The ICISAM version of the file
    - The default deletion type (logical or physical) is given along with the maximum file size
    - The number of alternate keys for indexed files, record size, and number of records allocated
    - For each key it reports:
        - Whether the key is the Primary or an Alternate along with seven possible attributes shown by the possible
            letters "dursaop" or "-------" if no attributes were given for the file. The possible attributes are shown
            below:
                d - duplicates are allowed,
                u - upper-case only,
                r - reverse (or DESCENDING) storage,
                s - suppress certain key values (the suppressed value is shown later),
                a - multiple scattered keys using ALSO clause,
                o - multiple tabular keys using OCCURS clause, and
                p - this key has suffixes using PLUS clause.
        - Finally the number of keys and records are shown.
    - The amount of space freed from packing.

XII. ICPQUTIL

A. **Introduction**

The ICPQUTIL utility validates and optionally dumps printer control (.pq) files.

B. **Syntax**

The standard syntax is:

```
icpqutil [-a[:aflag]|-A file|dir[:aflag]] [-d] [-e] [-h|-?] [-N dhjz] [-q]
    [-u] argument
```

Where
-a[:*aflag*]|A *file*|*dir*[:*aflag*] (Audit)
Enables auditing (default icpqutil.lg). Where *aflag* is a|b|d|p|t|u|da|db|pa|pb|ta|tb|ua|ub, defined as a-append, b-backup, d-date, p-pid, t-time, and u-username.
-d (Dump)
Dump the contents of the .pq file.
-e (fix Errors)
Fix any fixable errors.
-h|-? (Help)
Displays help text.
-N dhjz (No options)
Do NOT dump the selected options: d-devices, h-header, j-jobs, z-items with zero status.
-q (Quiet)
Enables quiet operation.
-u (Update)
Update the .pq file if needed.
argument
Specifies the .pq filename to be viewed. If not given, system.pq is used.

The -e (fix Errors) and the -u (Update) switches require exclusive access to the .pq file such that the file can be updated.

When updating the file with -u, all jobs in the file are checked to insure that the file is still available. If the file is no longer available then the file is removed from the .pq file.

Environment variables:

ICPQUTIL   Command line options
ICTMPDIR   Temporary files

C. **Description**

ICPQUTIL can be used as a debugging tool to check and correct .pq files that go bad. The various dump options allow the file to be viewed in several ways differently than what is shown under the Printer Control Utility.

XIII. ICREORG

A. **Introduction**

The ICREORG utility is a general file reorganization utility. It can convert files from one format to another among the supported Interactive COBOL file formats. In addition, the output records have a limited formatting capability that can be used to create reports. If the output file does not exist, it will be created. If it exists and is a sequential file, the new data will be appended. If it exists and is an indexed or relative file, the data will be merged according to the Merge switch (-m).

ICREORG assumes all indexed and relative input files are valid. It does not process corrupt files. ICFIXUP should be used to fix a corrupt indexed or relative file.

ICREORG can be used to remove logically deleted records from ISAM files. ICREORG can be used to remove physically deleted record slots.

B. **Syntax**

The standard syntax is:

```
icreorg [-a[:aflag]|-A file|dir[:aflag]] [-B num] [-C b|p:on|off] [-e]
    [-F pos:len|-F str:cnt]... [-h|-?] [-I type[:min[:max]]] [-k]
    [-K <keyspecifier>]... [-L file] [-m] [-N cnt] [-O type[:min[:max]]]
    [-p] [-q] [-R num] [-S key] [-u] [-V version] inputfile [outputfile]
```

Where
> -a[:*aflag*]|-A *file*|*dir*[:*aflag*] (Audit)
>> Enables auditing (default icreorg.lg). Where *aflag* is a|b|d|p|t|u|da|db|pa|pb|ta|tb|ua|ub, defined as a-append, b-backup, d-date, p-pid, t-time, and u-username.
>
> -B *num* (Beginning)
>> Specifies the beginning record number to use from the input file. It can be used with the Number (-N) switch to process just a portion of a file. The default is to begin with the first record. The record number is determined by numbering (counting) the records in the order they would be read according to the key order specified. Thus, the Nth record by the primary key will probably be different from the Nth record following an alternate key. The record number will also depend on the setting of the Undelete switch.
>
> -C b|p:on|off (Control output file attribute)
>> Set (on) or clear (off) the indicated file attribute(s). Available attributes are `b' or `p'. Control attribute can be specified multiple times to set each attribute.
>>> b (Big File) Allows the maximum file size of the file to grow to 4GB. If not set, the maximum file size is 2GB. Only allowed for version 7 files.
>>> p (Delete-is-physical) Sets the default type of record deletion on this file when neither LOGICAL nor PHYSICAL were specified in the delete operation. For logical deletes, the record is simply marked deleted so it can be undeleted; for physical deletes, the record area space is made available for new records to be written and no undelete can be done as the record is gone. The default is for the delete-is-physical attribute to be off. Only allowed for version 7 files.
>> This is not allowed for Btrieve or C-ISAM files.
>
> -e (Exclusive)
>> Open the input file exclusively.

-F *pos*:*len*|-F *str*:*cnt* (Field specifier)

    Specifies the next field in the output record. The first format specifies that *len* characters starting at position *pos* in the input record are to be copied to the output record. The second format specifies that *cnt* instances of the character(s) *str* are to be copied to the output record. If the string starts with a quote it must be ended with the same quote, if not started with a quote it ends before the first ":". If the string starts with an integer, it must be enclosed in quotes. If the string includes a ":", it must be enclosed in quotes. The string may use the format *nnn* to specify an arbitrary character in octal format. In order to include the quote character inside the string, the octal form must be used. If one Field specifier is used, the whole output record must be defined using one or more Field specifiers. Up to 33 field specifiers may be supplied.

-h|-? (Help)

    Displays help text.

-I *type*[:*min*[:*max*]] (Input type)

    Used during file conversions to specify the type of the input file, and, optionally, the input file record length. *Type* is one of the following:

        b   (On Windows only) BTRIEVE Indexed file
        c   (On UNIX only) C-ISAM Indexed file
        i   Indexed file
        l   Line sequential file (delimited by <cr>, <nl>, <ff>, <nul>, or <cr><nl>), omit zero length records
        r   Relative file
        s   Fixed-length sequential file, length is required.
        v   Variable-length sequential file
        z   Line sequential file, keep zero length records.

    If not specified, *type* defaults to i-indexed. The *min* field must be specified for fixed-length sequential files. If it is specified for other file types, the records are truncated or padded with null (or space for line-sequential) to the specified length.

-k (Keep)

    Maintain logically deleted records from the input file as they are copied to the output file. By default, deleted records are ignored when reading the input file. This is not allowed for Btrieve or C-ISAM files.

-K *pos*:*len*[:d][:r][:u][:s[=*val*]][:p=*ppos*:*plen*]...[:o=*cnt*:*span*] (Key specification)

-K *pos*:*len*[:d][:r][:u][:s[=*val*]][:a=*apos*]... (Key specification)

    Specifies the keys for indexed files. A key specification must be supplied for each key in the file. At least one key must be specified for an indexed file creation. The number of keys will be determined by the number of key specifications. The first key specification will be for the primary key, all subsequent key specifications will be treated as alternate keys. All the alternate keys are sorted like the COBOL compiler sorts alternate keys allowing them to be specified in any order. Up to 5 key specifiers may be listed for version 5 and 6 files; and up to 17 key specifiers may be listed for version 7 files.

        pos specifies a 1-based byte position in the record of the start of the key.

        len specifies the length of the key in bytes.

        :d specifies that the alternate key is to allow duplicate keys and is only allowed for alternate keys and is required for version 5 and 6 indexed files which always allow duplicates.

    The following key specification options are only allowed for version 7 files.

        :r specifies that this key is to be stored in reverse order.

        :u specifies that this key is always stored and retrieved in upper-case-only.

        :s[=*val*] specifies the value to suppress from key insertion and is only allowed on an alternate key. If *val* is not specified, LOW-VALUE is used.

        :p=*ppos*:*plen* specifies suffixed key values (PLUS) at the given position (*ppos*) and length (*plen*).

        :o=*cnt*:*span* and :a=*apos* specify multiple key locations in the record for this key and is only allowed on an alternate key. The :o parameter (OCCURS) gives a tabular view with *cnt* times and how far apart each entry is in bytes (*span*). The :a parameter (ALSO) specifies scattered key values for this key at the indicated positions (*apos*).

-L *file* (Like)

    Used to create the file to be like an existing file when the output file is an indexed or relative file. If *file* does not include an extension (or if it is .NX or .XD), it is assumed to specify an indexed or relative file. If the .FA extension is used, the contents of the file attribute file are used.

-m (Merge)

    If the output file is an indexed or relative file, directs the utility to merge new keyed records with existing

records by deleting the existing record with this primary key and then writing the newer record. If this switch is not set, the existing record is retained, and the new record is ignored.

-N *cnt* (Number)

Process at most *cnt* records. It can be used to limit the processing to the first *cnt* records in a file. If this switch is not specified, the input file is processed until end of file is reached. It can be used with the -B switch to select a range of records.

-O *type*[:*min*[:*max*]] (Output type)

Used during file conversions to specify the type of the output file, and, optionally, the output file record length. *Type* is one of the following:

    b   (On Windows only) BTRIEVE Indexed file
    c   (On UNIX only) C-ISAM Indexed file
    i   ICISAM Indexed file
    l   Line sequential file (delimited by \<cr>\<nl> on Windows and by \<nl> on UNIX)
    r   ICISAM Relative file
    s   Fixed-length sequential file
    v   Variable-length sequential file

If not specified and *outputfile* is specified, *type* defaults to indexed. If not specified and *outputfile* is not specified, *type* defaults to line-sequential. If not specified, *min* defaults to the length used for the input file unless the relative or indexed output file already exists in which case its length is used, or a like file was specified.. If different from the input length, the records are truncated or padded with null (or space for line-sequential) to the specified length. If *max* is specified, then *min* and *max* represent the minimum and maximum record sizes allowed in the file.

-p (Progress)

There is an ongoing display of the processing, which typically runs much slower for smaller files. The reporting interval is for every 1% of the file processed.

-q (Quiet)

Enables quiet operation.

-R *num* (Relative start)

Used when the output file is a relative file, and it directs the utility to renumber the relative key values beginning with the value *num*. By default, reorganizing a relative file to a relative file leaves the numbering unchanged, and converting another file type to relative, numbers the records consecutively beginning with the value 1.

-S *key* (Sequence key)

Used when the input file is an indexed file with alternate keys to change the order in which the file is read. By default, the file is read in the order of the primary key. This changes it to be in the order of one of the alternate keys. The *key* field can have the value `a1' to `a4' for version 5 or 6 files, and `a1' to `a16' for version 7 files.

-u (Undelete)

Undelete logically deleted records from the input file as they are being copied to the output file. By default, deleted records are ignored. Not allowed for Btrieve or C-ISAM files.

-V *version* (Version)

Sets the output file version number during conversions to an ICISAM indexed or relative file. The *version* may be 5, 6, or 7 for an ICISAM file.

inputfile

Specifies the input file.

outputfile

Specifies the output file. If not specified, standard out is assumed and the Output type is set to line-sequential.

Environment variables:

    ICREORG   Command line options
    ICTMPDIR   Temporary files

C. **Description**

The default input type is indexed, with the key and record information derived from the file. If the outputfile name is given and the output file does not exist, the default type will be the same as the input file. If the outputfile is not given then standard out is used and the default output type is set to line-sequential.

The -k and -u switches are mutually exclusive.

In a key specifier, the :o (OCCURS) and :a (ALSO) can not both be specified for a single key entry.

On UNIX, when using the Field specifier (-F) switch you must be careful about entering characters with the *nnn* octal specification as the shell interprets a single "\\" to mean take the next character literally when not within quotes and as an octal representation when within quotes. Thus, to use the *nnn* specification you should enclose the string within quotes. When using quotes you must also understand that the shell strips the quotes after using them to delimit a single argument. I.E., a "12HH" will be passed to ICREORG as 12HH which ICREORG will attempt to treat as a *pos* argument since it begins with a digit. (12HH is not legal as a *pos* argument and an error will be generated.) You must enter it as \\"12HH\\" to have the shell pass a "12HH" to ICREORG.

On Windows, when using the Field specifier switch (-F) with the string option, the double-quote (") symbol must be escaped with a backslash (\\) in order for ICREORG to see the double-quote. The close-quote (') can also be used.

An item to note for revision 7 indexed files is that if an alternate key that does not allow duplicates is written to the file and there already is an alternate key with that value, then the record will be not be written, it will be ignored. ICREORG will show the input record number for any duplicate key records. Just remember, when using logical deletes with records with alternate keys that DO NOT ALLOW DUPLICATES, a Duplicate key error can be given for an alternate key that points to a deleted record. The record must be physically deleted to insert a new record with the same alternate key.

Alternate record keys are sorted based on the following criteria (just as in the compiler):
   a. ascending root segment position.
   b. ascending root segment length.
   c. absence of also keys and if present ascending number of also and ascending alsos position.
   d. absence of suffixes, and if present ascending number of suffixes, ascending suffix position, and ascending
       suffix length.
   e. absence of occurs, and if present ascending number of occurs and ascending occurs span.
   f. absence of duplicates allowed.
   g. absence of descending order.
   h. absence of uppercase conversion.
   i. absence of suppress when value, and if present ascending suppress when value.

D. **Examples**

The following syntax reads the indexed file test6 and builds a line sequential file report1, that has the first 10 bytes of the record, followed by two spaces, followed by bytes 20-29, and 50-69 in the output file.

```
icreorg -O l -F 1:10 -F "\040":2 -F 20:10 -F 50:20 test6 report1
```

The following syntax reads the line sequential file inputdata, starting at record 2, and builds the indexed file outputdata using a 100 byte record with a 20 byte primary key starting at character position 11, and a 10 byte alternate key with duplicates starting at position 26. The output record is composed of 10 bytes of spaces, followed by the first 20 characters from the input record, followed by 5 "-" characters, and then followed by the next 65 characters (characters 21 through 80) from the input record.

```
icreorg -I l -B 2 -O i:100 -K 11:20 -K 26:10:d -F "\040":10 -F 1:20
   -F "-":5 -F 21:65 inputdata outputdata
```

XIV. ICREV

A. **Introduction**

The ICREV utility displays the file revision information for various types of Interactive COBOL files. The revision information includes the file format version, file creation information, file modification information, and the programmer supplied revision string that was set using ICREVSET or ICOBOL. This utility can also show the revision information for .PD/.DD files, even though the system never uses them directly.

B. **Syntax**

The standard syntax is:

```
icrev [-a[:aflag]|-A file|dir[:aflag]] [-c] [-h|-?] [-L file] [-m] [-q]
    [-r] [-s] { argument }...
```

Where
    -a[:*aflag*]|-A *file*|*dir*[:*aflag*] (Audit)
        Enables auditing (default icrev.lg). Where *aflag* is a|b|d|p|t|u|da|db|pa|pb|ta|tb|ua|ub, defined as a-append, b-backup, d-date, p-pid, t-time, and u-username.
    -c (Creation info)
        Only the file creation information is displayed.
    -h|-? (Help)
        Displays help text.
    -L *file* (Library)
        Specifies a library file in which to find the specified file(s).
    -m (Modification info)
        Only the file modification information is displayed.
    -q (Quiet)
        Enables quiet operation.
    -r (Revision info)
        Only the file format revision information is displayed.
    -s (Programmer info)
        Only the programmer(supplier) revision string is displayed.
    argument
        Can be a filename or a template. The filenames or templates must specify an extension. Specifying a library file as an argument returns revision information about the library itself.

Environment variables:

    ICREV   Command line options
    ICTMPDIR   Temporary files

C. **Description**

The Library switch says use a library to look for the files specified by the argument list rather than looking in the host file system.

The creation and modification information includes the date and time, the utility responsible, and the system the utility ran on. For .PD/.DD files, and for files made from a .PD/.DD, the information will also include the 1.xx **ICOBOL** and optimizer (ICOPT) information.

On UNIX, when using the library switch, template arguments may need to be quoted to prevent them from being expanded by the shell.

XV. ICREVW (Only on Windows)

A. **Introduction**

The ICREVW utility displays the file revision information for various types of Interactive COBOL files. ICREVW is the Windows GUI version of ICREV and is only available on Windows. The revision information includes the file format version, file creation information, file modification information, and the programmer supplied revision string that was set using ICREVSET or ICOBOL. This utility can also show the revision information for .PD/.DD files, even though the system never uses them directly.

B. **Syntax**

The standard syntax is:

```
icrevw [-a[:aflag]|-A file|dir[:aflag]] [-h|-?] [-q] [-u] { argument }...
```

Where
    -a[:*aflag*]|-A *file*|*dir*[:*aflag*] (Audit)
        Enables auditing (default icrevw.lg). Where *aflag* is a|b|d|p|t|u|da|db|pa|pb|ta|tb|ua|ub, defined as a-append, b-backup, d-date, p-pid, t-time, and u-username.
    -h|-? (Help)
        Displays help text.
    -q (Quiet)
        Enables quiet operation.
    -u (Update)
        Allow the programmer(supplier) revision string is to be updated.
    argument
        Can only be a filename. The filenames must specify an extension. Specifying a library file as an argument returns revision information about the library itself.

Environment variables:

ICREVW	Command line options
ICTMPDIR	Temporary files

C. **Description**

The creation and modification information includes the date and time, the utility responsible, and the system the utility ran on. For .PD/.DD files, and for files made from a .PD/.DD, the information will also include the 1.xx **ICOBOL** and optimizer (ICOPT) information.

XVI. ICSHELLX (Only on Windows)

A. Introduction

ICSHELLX is a utility program which extends the capabilities of Windows Explorer on systems with **ICOBOL** installed. This extended information includes revision information on all standard icobol file types and enhanced descriptions of the structure of ICISAM files. During **ICOBOL** installation the shell extensions module (ICSHELLX.DLL) is registered with the Windows operating system. Thereafter it becomes available as one or more tabs in the Properties window of a file selected on Windows Explorer.

B. Use

To use the extensions, select an **ICOBOL** file from within Windows Explorer. A single click of the left mouse button will accomplish this task. Next a single click of the right mouse button will bring up a context menu. Move the mouse so that the "Properties" menu entry is highlighted and select it using a single left click. The properties entry is normally the last entry on the context menu.

At this point a tabbed Properties window will appear. In addition to the standard tabs labeled "**General**" and "**Security**", when ICSHELLX is available a "**Revision**" tab will be shown. Other tabs may be available as well depending upon the type of file initially selected.

Select the "**Revision**" tab by a single left clink. A screen similar to the following will be shown.
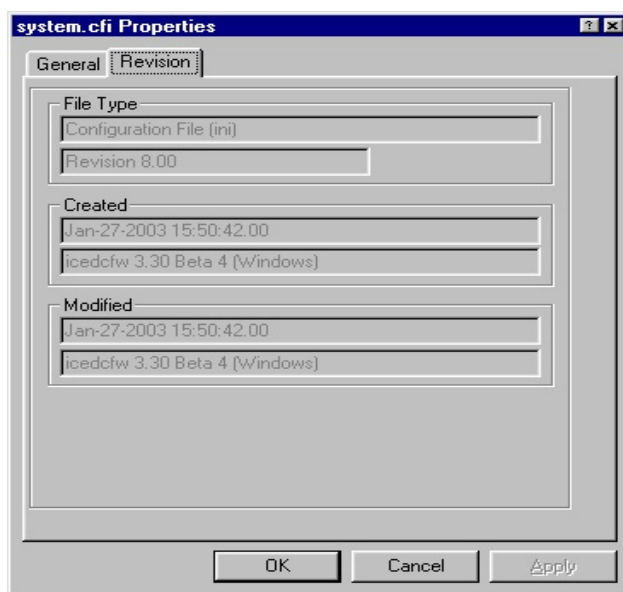
The "**Revision**" tab provides the following information:

- A description of the file including its file type, revision number and byte-ordering.

- The date and time that the file was created, and the process that created the file along with its revision number

- The date and time that the file was last modified, and the process that last modified the file along with its revision number

- The supplier (OEM) revision number and the process and its revision number that applied the supplier revision to the file, except for .ini type files (cfi, pti, and .tdi).

ICSHELLX provides a **Revision** tab for all standard **ICOBOL** files including ICSAM files (.nx, .xd), ICISAM file attribute files (.fa), program files (.cx), libraries (.cl), symbol table files (.sy), project files (.icp), printer control files (.pq), and link files (.lk). It will also provide information without the supplier (OEM) information for configuration files (.cfi), printer translation files (.pti), and terminal description files (.tdi).

If either the .xd or .nx of an ICISAM file is selected, ICSHELLX provides several additional tabs to the Properties window. These additional tabs are labeled as "**File**", "**Record**", and "**Node**" and may be selected with a single left click.

The "**File**" tab is shown below.
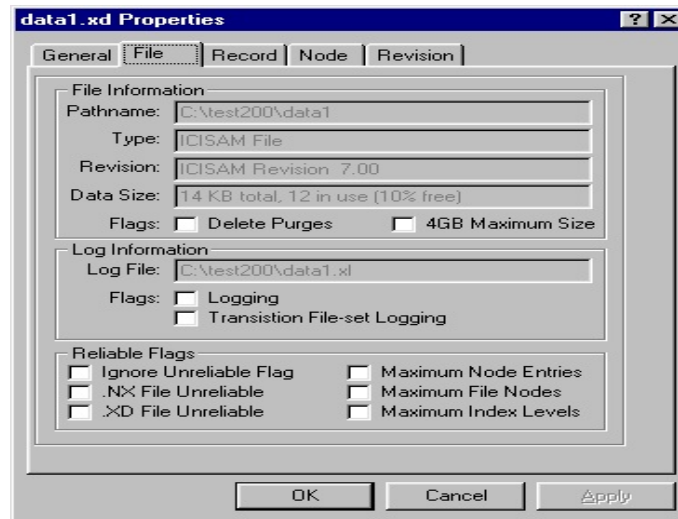
The "**File**" tab will show three sections.

- The "File Information" section includes the full pathname of the file, the file type and revision, the available data space in the file's .xd and the percentage of that space which is in use. Also shown are two check boxes which will be checked if physical deletes are the default (Delete Purges) and if large file support is enabled (4GB Maximum size).

- The "Log Information" section includes the full pathname to the ICISAM log file, and check boxes which when checked indicate if either logging or transition logging is enabled for the file.

- The "Reliable Flags" section contains a series of check boxes which when checked indicate the status of the file. The boxes include the state of the reliability bits for each portion of the file, the ignore reliability bits flag used internally, and the flags indicating that the maximum number node entries, nodes per file or index levels have been hit.

The "**Record**" tab is shown below.
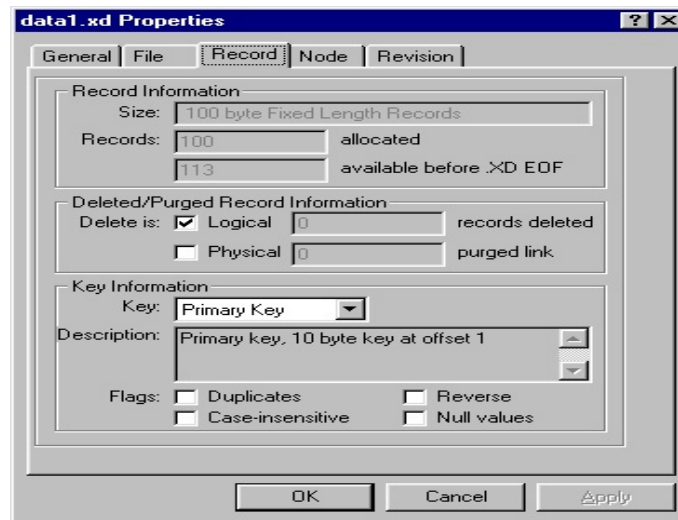
The "**Record**" tab will again show three sections.

- The "Record Information" section shows the size of the records, how many have been allocated and how many are available before the end of file.

- The "Deleted/Purged Record Information" shows whether the default delete type is logical or physical, the logically deleted record count and the purged record link.

- The "Key Information" section shows information about a files's keys. The key to be described may be selected via a pull-down list. The information shown includes key size, record offset and composition of each of the keys as well as check boxes for whether the key's description contains the ALLOWS DUPLICATES clause (Duplicates), the ORDER BY ALPHABETIC-UPPER clause (Case-insensitive), the VALUES ARE DESCENDING clause (Reverse) or the SUPPRESS WHEN clause (Null values).

The "**Node**" tab is shown below.
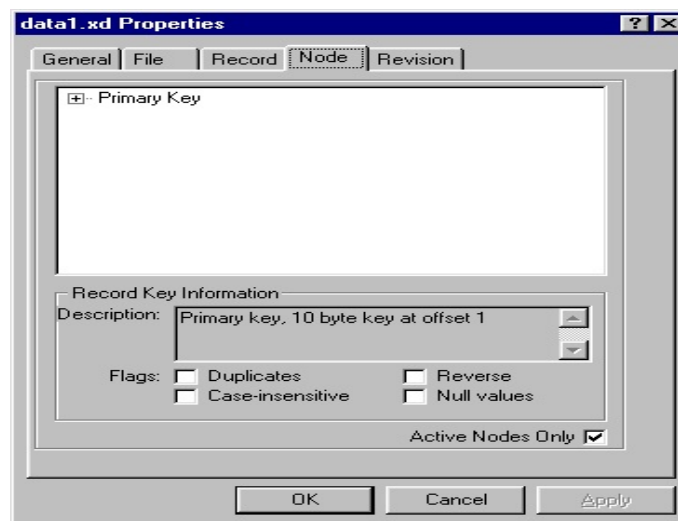
On the "**Node**" tab a tree view of the keys in the file will be shown. The mouse may be used to highlight entries in the tree and to expand or collapse branches of the tree. Under each key are entries for the index nodes starting with the root node (i.e., node 1), continuing through the various index levels, and at a leaf-node, the records offsets are displayed.

Selecting the check box "Active nodes Only" will partially limit the display to data relevant to the current contents of the file. When an entry is highlighted in the tree various pertinent information is displayed in the section below the tree view.

- For a key such as Primary Key you will see a display of information similar to that shown in the "Key Information" section of the "Record" tab.

- For a node various information from the node header is displayed including its node number, the key to which it belongs, its index level and the maximum entry count.

- For an active record entry you will see the key's position in the file, and the key value. The key value can be shown in either ASCII or Hex by selecting the appropriate check box.

If an ICISAM file attributes file (.fa) is selected, ICSHELLX provides an "**Attributes**" tab in addition to the "**Revision**" tab. The "**Attributes**" tab may be selected with a single left-click.

The "**Attributes**" tab for a file attributes file is shown below.
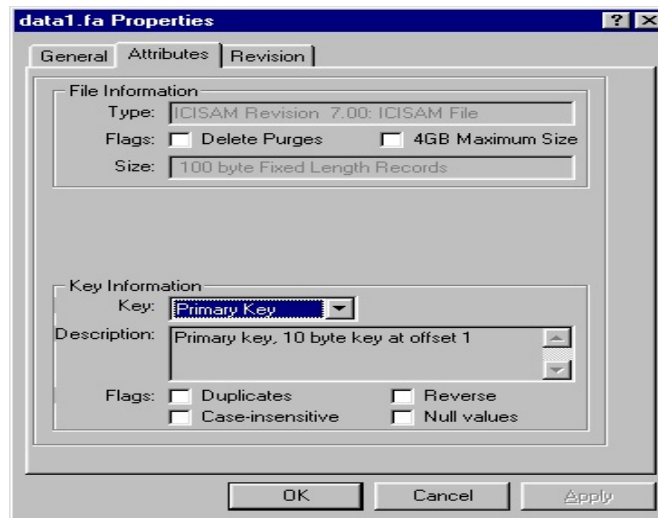
The "**Attributes**" tab will show two sections.

- The "File Information" section includes the file type and revision and the record size. Also shown are two check boxes which will be checked if physical deletes are the default (Delete Purges) and if large file support is enabled (4GB Maximum size).

- The "Key Information" section shows information about a files's keys. The key to be described may be selected via a pull-down list. The information shown includes key size, record offset and composition of each of the keys as well as check boxes for whether the key's description contains the ALLOWS DUPLICATES clause (Duplicates), the ORDER BY ALPHABETIC-UPPER clause (Case-insensitive), the VALUES ARE DESCENDING clause (Reverse) or the SUPPRESS WHEN clause (Null values).

The default file associations are:

When	Ext	File Description / Context Menu Options	Command line
R	.cfi	ICOBOL Configuration File (ini)	
		Configure	Icedcfw %1%
R	.cl	ICOBOL Library File	
D	.co	ICIDE Card-format Source	
		Open	Icide %1%
		Print	Icide /p %1%
R	.cx	ICOBOL Program File	
		Run	Icrun -a -C default %1%
D	.er	ICOBOL Error Listing	
		Open	Notepad %1%
		Print	Notepad /p %1%
		If installed, Icide is used in place of Notepad.	
R	.fa	ICISAM File Attributes	
D	.gsy	ICIDE Global Symbols	
D	.icp	ICIDE Project	
always	.lg	ICOBOL Log File	
		Open	Notepad %1%
		Print	Notepad /p %1%
always	.lgb	ICOBOL Log File (backup)	
		Open	Notepad %1%
		Print	Notepad /p %1%
always	.lic	ICOBOL License File	
		Open	Notepad %1%
		Print	Notepad /p %1%
R,D	.lk	ICOBOL Filename Link File	
R,D	.ls	ICOBOL Program Listing File	
		Open	Notepad %1%
		Print	Notepad /p %1%
		If installed, Icide is used in place of Notepad.	
R	.nx	ICISAM Index File	
		Properties shows File, Record, Node, and Revision tabs	
S	.pan	ICOBOL sp2 Panel File	
		Edit	Icsp2 -a %1%
R	.pq	ICOBOL Printer Queue File	
R	.pti	ICOBOL Printer Translation File (ini)	
		Configure	Icedcfw -a %1%
D	.sr	ICIDE Free-format Source	
		Open	Icide %1%
		Print	Icide /p %1%
D	.sy	ICOBOL Symbol Table File	
R	.tdi	ICOBOL Terminal Description File (ini)	
		Configure	Icedcfw -a %1%
R	.xd	ICISAM Data File	
		Properties shows File, Record, Node, and Revision tabs	
D	.xdb	ICODBC Database Definition File	
		Edit	Notepad %1%
		Print	Notepad /p %1%
D	.xdt	ICODBC Table Definition File	
		Edit	Notepad %1%
		Print	Notepad /p %1%
R	.xl	ICISAM Transaction Log	

Where the "When" column letters refer to what component was installed based on the following:

R - Runtime   U - User Library   D - Development

L - Link Kit   S - Sp2 Development

These file associations can be edited by executing My Computer and choosing the View / Options / File Types menu selections.

XVII. ICSMVIEW

A. **Introduction**

The ICSMVIEW utility allows the shared area created by ICEXEC to be viewed. This utility is available on UNIX and Windows and will only work when ICEXEC is running.

B. **Syntax**

The standard syntax is:

```
icsmview [-a[:aflag]|-A file|dir[:aflag]] [-b] [-B {b|h|l|p}...] [-F file]
    [-G {g|p|s|u}...] [-h|-?] [-I index] [-L {l|p}...]
    [-O {c|d|e|f|h|i|j|p|q|r|s|x|y}...] [-P pid] [-q] [-r] [-v|-V num]
```

Where
   -a[:*aflag*]|A *file*|*dir*[:*aflag*] (Audit)
       Enables auditing (default icsmview.lg). Where *aflag* is a|b|d|p|t|u|da|db|pa|pb|ta|tb|ua|ub, defined as a-
       append, b-backup, d-date, p-pid, t-time, and u-username.
   -b (Bypass)
       Bypass access conventions.
   -B {b|h|l|p}... (Buffers)
       Show Buffers: b-buffer descriptors, h-buffer hash descriptors, l-buffer LRU, p-buffers per file
   -F *file* (File)
       Show information about file.
   -G {g|p|u|s}... (General)
       Show the areas to dump. Options are::
           g -global information
           p -pid table
           s -semaphores (UNIX)
           u -UID table (UNIX)
   -h|-? (Help)
        Displays help text.
    -I index (Index)
        Specify index number of item to process
    -L {l|p|... (Locks)
       Show Locks: l -record locks, p -record locks per file
    -O {c|d|e|f|h|i|j|p|q|r|s|x}... (Objects)
        Show the specified objects:
           c-CON, d-device, e-SER, f-file, h-handle (obj ids), i-indexed, j-PCQ jobs, p-PRN, q-PCQ, r-relative,
           s-sequential, x-programs, y-pdf formats
   -P *pid* (Pid)
       Find the given *pid* in the shared area and display information about it.
    -q (Quiet)
       Enables quiet operation
    -r (Reverse)
       Reverse processing order
    -v|-V *num* (Verbose)
       Specify verbose mode. If -V num is specified, the greater the number the more verbosity will be generated.
       Num=0 is the same as -v.

Environment variables:

   ICSMVIEW   Command line options

If no actions are specified, the default is:

```
-B b -G gpu -L l -O cdefipqrsxy -bv
```

To get even more information use the above and add the -V 2 to replace the -v.

C. **Description**

ICSMVIEW can be used as a debugging tool to check out information in the shared area maintained by ICEXEC. In particular the bypass option causes no semaphores to be used when accessing the shared area. In hang situations this is required.

Some example output from icsmview is shown below along with the command that generates the output.

icsmview -P 234

```
   Tracing PID (234)...
   proc_num:        5   proc_pid:        234        icthins     sp2logon
```

icsmview -P 234 -v

```
   Tracing PID (234)...
   proc_num:        5   proc_pid:        234        icthins     sp2logon
         con_num     =        2040       username       = Ralph

   Open Handles:
      han_obj_id [3].class = File
      han_obj_id [3].type  = File
      han_obj_id [3].index = 1

   1 opened handles.
```

icsmview -G g

```
   Global Information ...

   Version number      = 1
   Control pid         = 86
   Shared area size    = 221620016 (bytes) 216425 (KB)   211 (MB)
      Buffer part      = 209715200 (bytes) 204800 (KB)   200 (MB)
   Flags               = Ready, Active, Enabled
   Handle/program information
      handle_count     = 128
```

icsmview -G p

```
   Processing PID Table ...

   Total processes            = 2048
   Processed in use           = 8
   Max processes used         = 8
```

icsmview -G p -v

```
   Processing PID Table ...

   Total processes           = 2048
   Processed in use          = 8
   Max processes used        = 8

   proc_num:          0   proc_pid:        86      (icexec)
   proc_num:          1   proc_pid:       158      icrun   logon
   proc_num:          2   proc_pid:       214      icrunrs logon
   proc_num:          3   proc_pid:       210      icrun   logon
   proc_num:          4   proc_pid:       230      icrunrs logon
   proc_num:          5   proc_pid:       234      icthins sp2logon
   proc_num:          6   proc_pid:       209      icthins sp2logon
   proc_num:          7   proc_pid:       163      (icsmview (me))
```

XVIII. ICSORT

A. **Introduction**

The ICSORT utility is a general purpose sort and merge utility with scripting capability. The sort operation takes up to ten(10) input files, sorts on from one(1) to twenty(20) keys, and produces a sequential output file. The merge operation takes up to ten(10) sorted sequential files of the same type and merges them into a single sequential output file.

ICSORT can produce a variety of different output files from a given input file. By selecting only certain portions of the input file, ICSORT can reformat the records for the output file. Thus, ICSORT can be used as a tool for generating tailored reports from a master file. Records can be sorted on any data type in ascending order, descending order, or according to a user-defined collating sequence. Translation between ASCII and EBCDIC and uppercase and lowercase are also supported. If sorting or translation is not part of the reporting process, ICREORG is more efficient in this process.

B. **Syntax**

The standard syntax is:

```
icsort [-a[:aflag]|-A file|dir[:aflag]] [-C file] [-d] [-D file] [-e]
    [-F pos:len|-F str:cnt]... [-K pos:len[:a|d[:dtype]]]... [-h|-?] [-m]
    [-M size] [-q] [-t] [-S sfile] {-I itype[:len] input-file}...
    {-O otype[:len] output-file}
```

Where
-a[:*aflag*]|-A *file*|*dir*[:*aflag*] (Audit)
Enables auditing (default icsort.lg). Where *aflag* is a|b|d|p|t|u|da|db|pa|pb|ta|tb|ua|ub, defined as a-append, b-backup, d-date, p-pid, t-time, and u-username.
-C *file* (Collating)
The collating sequence definition is in the specified file.
-d (Delete duplicates)
Delete duplicates from the output.
-D *file* (Write duplicate to file)
Writes duplicates to the indicated *file* (same format as output file)
-e (Exclusive)
Open input file(s) exclusively. This prevents the files from being changed while the sort is in progress and slightly increases performance.
-F *pos*:*len*|-F *str*:*cnt* (Field specifier)
Specifies the next field in the output record. The first format specifies that *len* characters starting at position *pos* in the input record are to be copied to the output record. The second format specifies that *cnt* instances of the character(s) *str* are to be copied to the output record. If the string starts with a quote it must be ended with the same quote, if not started with a quote it ends before the first ":". If the string starts with an integer, it must be enclosed in quotes. If the string includes a ":", it must be enclosed in quotes. The string may use the format *nnn* to specify an arbitrary character in octal format. In order to include the quote character inside the string, the octal form must be used. If one Field specifier is used, the whole output record must be defined using one or more Field specifiers. Up to 33 field specifiers may be supplied.
-K *pos*:*len*[:a|d[:*dtype*]] (Key specification)
Specifies the key fields in the input records on which to sort. At least one key specifier must be listed unless the script option is used. Up to 20 key specifiers may be listed with no more than 1000 bytes total length. The keys are applied in the order they appear in the command-line, with the left-most being most significant in the sort. The first two fields, *pos* and *len*, are like pos and len in output field specifier. The next field is the sort order specifier, which is `a' for ascending and `d' for descending. If the sort order specifier is omitted, the default order is ascending. The final field is the data type (*dtype*) field from one of

the following:

a	ASCII (8-bit) data (default)
b	binary/computational unsigned data
bs	binary/computational signed data
c5	unsigned COMP-5 data
c5s	signed COMP-5 data
n	numeric unsigned display data
nl	numeric signed display leading overpunch data
nls	numeric signed display leading separate data
nt	numeric signed display trailing overpunch data
nts	numeric signed display trailing separate data
p	packed-decimal unsigned data
ps	packed-decimal signed data

If no *dtype* argument is specified, the default is `a' for ASCII data.

-h|-? (Help)

Displays help text.

-I *itype*[:*len*] *inputfile* (Input file)

Specifies the input file type, length (*len*), and name. The input file type (*itype*) is one of the following:

i	ICISAM Indexed file, length is not allowed
l	Line sequential file, length sets the maximum record size (delimited by <nl>, <cr>, <ff>, <nul>, or <cr><nl>), zero-length records are omitted. If not specified, 2047 is assumed.
r	ICISAM Relative file, length is not allowed
s	Fixed-length sequential file, length is required to set the record size
v, vb	Variable binary-length sequential file where the length is stored as a 2-byte binary value. (length is required to set the maximum record size)
va	Variable ASCII-length sequential file where the length is stored as a 4-byte ASCII value. (length is required to set the maximum record size)

-m (Merge)

Merge files instead of sorting. The input files are assumed to be sorted according to the keys specified.

-M *size* (Memory)

Specifies the amount of memory in KB to use for sorts. For memory rich systems, this option can be used to improve the performance for very large sorts. (1024 is 1 megabyte). The default is 1024. Care should be taken when using this option as it can dramatically slow performance when memory saturation is reached.

-O *otype*[:*len*] *outputfile* (Output file)

Specifies the output file type, length, and name of the output file. If field specifiers have been used to construct the output record, the record length must be at least that size; if larger, the record is padded with spaces. If length is not given, it defaults to the maximum input record size or to that specified by the Field specification. The output file type (*otype*) is one of the following letters:

l	Line sequential file (<cr><nl> delimiter on Windows, else <nl> delimiter)
s	Fixed-length sequential file
v, vb	Variable binary-length sequential file where the length is stored as a 2-byte binary value.
va	Variable ASCII-length sequential file where the length is stored as a 4-byte ASCII value.

-q (Quiet)

Enables quiet operation.

-S sfile (Script file)

Specifies a script file to control operations. If specified, none of the -C, -d, -D, -F, -I, -K, -m, -O, or -t options may be specified on the command line.

-t (Tag sort)

Perform a tag sort instead of a record sort. When disk space is tight (such that allowing ICSORT to use a record sort would result in an out-of-disk-space error) then a tag sort should be done. Only the key itself along with a pointer are stored in the temporary file(s) thus greatly decreasing the amount of temporary disk space required. Generally this tradeoff is with performance causing the sort to run slower. When using a tag sort, the input file(s) must be locked if the -e option was not specified to prevent changes to the file while the sort is in progress. This could cause other users to pend on updates to these file(s) until ICSORT is finished.

Environment variables:

      ICSORT   Command line options
      ICTMPDIR   Temporary files

C. **Description**

A specific collating sequence can be defined by the user by creating a file that contains an alternate sequence and specifying this sequence's filename on the command line.

The input and output record counts are displayed at termination. Warnings are generated if records are truncated to fit in the specified output length of if input records contain no data.

ICSORT uses program-generated temporary work files of the form icd<pid>.tmp (data), ick<pid>.tmp (key), and ict<pid>.tmp (tag) for the sort operation.

The order in a sort is usually determined by straight ASCII (8-bit) sequence. The ASCII characters are represented by internal codes consisting of decimal integers from 0 through 255. When specified, the alternate collating sequence applies only to keys having the ASCII data type. The alternate collating sequence has no effect on keys having numeric or computational data types.

The alternate collating sequence file can be built by specifying up to 256 lines with the characters to be sorted. The default file is basically a file with 256 lines starting with \000 on line 1, \001 on line 2, ..., through \377 on line 256. Each line is of the form {*chr*}... where *chr* can either be the ASCII character or the character as specified by the octal format *nnn*. Any unspecified characters are assigned remaining collating positions in order.

On UNIX, when using the Field specifier (-F) switch you must be careful about entering characters with the *nnn* octal specification as the shell interprets a single "\\" to mean take the next character literally when not within quotes and as an octal representation when within quotes. Thus, to use the *nnn* specification you should enclose the string within quotes. When using quotes you must also understand that the shell strips the quotes after using them to delimit a single argument. I.E., a "12HH" will be passed to ICSORT as 12HH which ICSORT will attempt to treat as a *pos* argument since it begins with a digit. (12HH is not legal as a *pos* argument and an error will be generated.) You must enter it as \"12HH\" to have the shell pass a "12HH" to ICSORT.

D. **Script Files**

Script files are text files containing a series of commands that tell ICSORT how to process the data. These files may be created with any text editor. When using script files, no other ICSORT command-options that effect data may be entered on the command line. Scripting provides for all aspects of the sort/merge process from input and output descriptions, key descriptions, how to sort/merge/copy, and how to massage the file output including data translation.

The script file is composed of three different types of statements: Definition, When, and Imperative. The Definition and Imperative are required while the When statement(s) are optional and may be repeated. Comments are denoted by the standard COBOL comment character of "*" in the first position or "*>" for end-of-line comments.

A sort-merge script is structured as a definition section that is bracketed by DEFINE and END-DEFINE, followed by zero or more massage sections that are bracketed by WHEN and END-WHEN, and concluded with an imperative statement that gives the primary operation of the script.

Syntax of a script:

<u>DEFINE</u>

> *define-statement...*

<u>END-DEFINE</u>

$$\left[\underline{\text{WHEN}} \left\{ \begin{array}{c} \underline{\text{READ}} \text{ [OF } \textit{file-identifier} \text{]} \\ \underline{\text{WRITE}} \end{array} \right\} \atop \begin{array}{l} \textit{massage-statement...} \\ \underline{\text{END-WHEN}} \end{array} \right] ...$$

> *imperative-statement*

A *define-statement* is one of the following:

<u>INPUT</u> FILE [*file-identifier*] IS *filename-string* [*records-clause*] [.]

<u>OUTPUT</u> FILE IS *filename-string* [*records-clause*] [.]

<u>COLLATING</u> <u>SEQUENCE</u> *collate-identifier* <u>FROM</u> *filename-string* [.]

<u>TABLE</u> *table-identifier* <u>FROM</u> *filename-string* [.]

The *records-clause* is defined as follows:

$$\underline{\text{RECORDS}} \text{ ARE} \left\{ \begin{array}{l} \textit{integer} \text{ CHARACTERS} \\ \underline{\text{DATA-SENSITIVE}} \text{ UP TO } \textit{integer} \text{ CHARACTERS [} \underline{\text{DELIMITED}} \text{ BY } \textit{string...} \text{]} \\ \underline{\text{VARIABLE}} \left[\begin{array}{l} \underline{\text{ASCII}} \\ \underline{\text{BINARY}} \end{array} \right] \text{UP TO } \textit{integer} \text{ CHARACTERS} \\ \underline{\text{INDEXED}} \\ \underline{\text{RELATIVE}} \end{array} \right\}$$

A *massage-statement* is one of the following:

$$\underline{\text{APPEND}} \left\{ \begin{array}{c} \text{RECORD } \underline{\text{COUNT}} \\ \textit{string} \end{array} \right\} \text{[.]}$$

$$\underline{\text{CONVERT}} \textit{ location} \text{ [, } \textit{location} \text{]... } \underline{\text{USING}} \left\{ \begin{array}{l} \textit{table-identifier} \\ \underline{\text{ASCII}} \\ \underline{\text{EBCDIC}} \\ \underline{\text{LOWER}} \\ \underline{\text{UPPER}} \end{array} \right\} \text{[.]}$$

<u>DELETE</u> <u>AREA</u> *location* [.]

$$\underline{\text{DELETE}} \text{ CHARACTERS [} \underline{\text{IN}} \textit{ location} \text{] [} \underline{\text{MOVE}} \text{ REMAINDER} \left\{ \begin{array}{l} \underline{\text{LEFT}} \\ \underline{\text{RIGHT}} \end{array} \right\} \text{[} \underline{\text{PAD}} \text{ WITH } \textit{string} \text{]] [.]}$$

<u>IF</u> *condition* <u>THEN</u>

> *massage-statement...*

$$\left[\begin{array}{l} \underline{\text{ELSE}} \\ \quad \textit{massage-statement...} \end{array} \right]$$

<u>END-IF</u>

INSERT { RECORD <u>COUNT</u> / string } <u>BEFORE</u> *integer* [.]

<u>NEXT RECORD</u> [.]

<u>PAD TO</u> *integer* CHARACTERS [<u>WITH</u> *string*] [.]

<u>REFORMAT</u> { *location* / *string* } [, { *location* / *string* }]... [.]

<u>REPLACE TAB</u> CHARACTERS [<u>IN</u> *location*] [<u>WITH</u> *string*] [<u>AT</u> *integer* [, *integer*]... / <u>EVERY</u> *integer*] [.]

<u>REPLACE</u> [<u>FIRST</u> / <u>ANY</u> / <u>ALL</u>] *string1* [<u>IN</u> *location*] <u>WITH</u> *string2* [.]

<u>SEND</u> RECORD <u>TO</u> *filename-string* [.]

<u>STOP</u> [.]

The *condition* in the IF statement has the following syntax:

[<u>NOT</u>] { <u>BLANK</u> / *relation* / (*condition*) } [{ <u>AND</u> / <u>OR</u> } [<u>NOT</u>] { <u>BLANK</u> / *relation* / (*condition*) }] ...

where *relation* is:

{ *string* / *location* / RECORD <u>COUNT</u> / RECORD <u>LENGTH</u> } { = / <> / < / <= / > / >= / :=: / :<>: } { *string* / *location* / *integer* }

The *imperative-statement* is one of the following:

<u>COPY</u> [.]

<u>MERGE</u>
    [{ <u>DELETE DUPLICATES</u> / <u>WRITE DUPLICATES TO</u> *filename-string* } [.]]
    { <u>KEY</u> *location* [IS *type-clause*] [<u>ASCENDING</u> / <u>DESCENDING</u>] [.] }...
<u>END-MERGE</u>

105

```
SORT
    [ USE { KEYS ONLY } [.] ]
          { RECORDS   }

    [ {          DELETE DUPLICATES            } [.] ]
      { WRITE DUPLICATES TO filename-string    }

    { KEY location [ IS type-clause ] [ ASCENDING  ] [.] }...
                                     [ DESCENDING ]
END-SORT
```

In the MERGE and SORT statements, the *type-clause* is defined as follows:

```
{                                                { identifier }    }
{                                                {   ASCII    }    }
{ ALPHANUMERIC [ COLLATING SEQUENCE IS          {   EBCDIC   } ]   }
{                                                {   LOWER    }    }
{                                                {   UPPER    }    }
{                                                                  }
{                  [             { BINARY [ WITH SIGN ]         } ] }
{                  [             { COMPUTATIONAL-5 [ WITH SIGN ]} ] }
{                  [ USAGE IS    { PACKED-DECIMAL [ WITH SIGN ] } ] }
{ NUMERIC          [             { DISPLAY [ SIGN IS { LEADING  } [ SEPARATE ] ] } ] }
{                  [             {                   { TRAILING }                } ] }
{                  [ SIGN IS { LEADING  } [ SEPARATE ]                             ] }
{                  [          { TRAILING }                                          ] }
```

As used in the formats above, a *string* is defined as:

```
{ quoted-string }
{      CR       }
{      ESC      }
{      FF       }
{      LF       }
{      NUL      }
{     SPACE     }
```

As used in the formats above, a *location* is defined as:

```
{ start-integer / { end-integer } }
{                 {    LAST      } }
{                                 }
{ start-integer : length-integer  }
```

A *filename-string* is a quoted string containing a valid operating-system pathname of the file.

Abbreviations:

ASC	- ASCENDING
CHAR	- CHARACTER
CHARS	- CHARACTERS
COMP-5	- COMPUTATIONAL-5
DESC	- DESCENDING
PACKED	- PACKED-DECIMAL

The punctuation characters are period and comma. In general, they are optional.

The ASCII, EBCDIC, ASCII to EBCDIC, and EBCDIC to ASCII tables are provided in Appendices A thru D on

pages [153](#), [155](#), [156](#), and [157](#) respectively. The lowercase to uppercase table has the characters "a" - "z" mapped to "A" - "Z" respectively, while the uppercase table has the characters "A" - "Z" mapped to "a" - "z" respectively. The lowercase sort table has the uppercase letters "A" - "Z" equivalent to the respective lowercase letters "a" - "z". While the uppercase sort table is just the reverse.

E. **Examples**

Command-Line Usage

To specify an alternate collating sequence where the lower-case letters are equivalent to their upper-case counterparts, the file would be as follows. Lines 1 through 65 would contain the values \000 through \100. Line 66 would contain an "Aa", line 67 an "Bb", etc. through line 91 which would contain "Zz". Lines 92-97 would remain \133 through \140. Lines 98 through 122 would be deleted. The next line would start with \173 and continue line by line through \377.

To specify an alternate collating sequence where the upper-case letters are before the numeric digits, the file would be as follows. Lines 1 through 48 would contain the values \000 through \057. The next 26 lines (lines 49 through 75) would contain an "A" through "Z". Lines 76 and on would be the remaining values through \377.

Scripting Usage

Below is a sample script

```
* This is a test sort script

DEFINE
   INPUT FILE seq1 IS "seq1"
      RECORDS ARE 100 CHARACTERS.
   INPUT FILE seq2 IS "seq2"
      RECORDS ARE 100 CHARACTERS.

   OUTPUT FILE IS "joeout"
      RECORDS ARE VARIABLE ASCII UP TO 200 CHARACTERS.
      WORK RECORD UP TO 200 CHARACTERS.
END-DEFINE

WHEN READ OF seq1
   delete area 30/49.
   insert "seq1" Before 30.
END-WHEN

WHEN READ
   delete area 30/49.
   insert "seq2" before 30.
   append "*******************"
*       replace all "bf" in 30/50 with "cdefb".
*       replace all "bffff" in 30/50 with "fb".
END-WHEN

SORT
   KEY    3/10 NUMERIC USAGE binary SIGN.
   KEY    30:4 IS ALPHANUMERIC DESC
END-SORT


end of sample script.
```

XIX. ICSTAT

A. **Introduction**

The ICSTAT utility analyzes ICISAM indexed and relative files, and reports useful information and statistics.

B. **Syntax**

The standard syntax is:

```
icstat [-a[:aflag]|-A file|dir[:aflag]] [-h|-?] [-N s] [-p] [-q]
    [argument]...
```

Where
  -a[:*aflag*]|-A *file*|*dir*[:*aflag*] (Audit)
    Enables auditing (default icstat.lg).   Where *aflag* is a|b|d|p|t|u|da|db|pa|pb|ta|tb|ua|ub, defined as a-append,
    b-backup, d-date, p-pid, t-time, and u-username.
  -h|-? (Help)
    Displays help text.
  -N s (No options)
    Specifies NO options. Valid NO options are:
        s (No-scan)   Causes the file to NOT be scanned, only read the headers.
  -p (Progress)
    Displays an ongoing display of the processing, which typically runs much slower for smaller files. The
    reporting interval is 1% of the file processed.
  -q (Quiet)
    Enables quiet operation.
  argument
    Specifies any filename or template to be checked. If not given it defaults to all ICISAM files in the current
    directory.

Environment variables:

  ICSTAT   Command line options
  ICTMPDIR   Temporary files

C. **Description**

The following information about an ICISAM file is given:

- A possible Warning message that the end-of-file of the .XD file is not on a 2048 byte boundary
- The ICISAM version of the file
- The default deletion type (logical or physical) is given along with the maximum file size
- The number of alternate keys for indexed files, record size, and number of records allocated
- The total number of record slots available for records before the .XD end-of-file
- For each key it reports:
    - Whether the key is the Primary or an Alternate along with seven possible attributes shown by the possible
      letters "dursaop" or "-------" if no attributes were given for the file. The possible attributes are shown
      below:
          d duplicates are allowed,
          u upper-case only,
          r reverse (or DESCENDING) storage,
          s suppress certain key values (the suppressed value is shown later),
          a multiple scattered keys using ALSO clause,

        o multiple tabular keys using OCCURS clause, and

        p this key has suffixes using PLUS clause.

- Finally the number of keys, records, and purged keys are shown. If the No-scan switch was given, these numbers will be zero.
- For each level in the index for that key the number of nodes, the number of keys in that node, and the average density of the nodes if the No-scan switch was NOT given.
- The key length and offset in the record including suffixes (PLUS), OCCURS, scattered keys (ALSO), and any suppression value (SUPPRESS WHEN) shown as LOW-VALUE, HIGH-VALUE, SPACE, ZERO, or its octal value. Finally the maximum key entries per index node is given.
- The total number of indexed nodes if the No-scan switch was NOT given.
- The number of logically deleted records in the file if the No-scan switch was NOT given.

The Warning about the end of the .XD file not being on a 2048 byte boundary can be ignored unless the file must be moved back to an older Interactive COBOL system (1.5x - 1.7x) or used on AOS/VS.

The total number of records allocated (written) and the total number of record slots available will be different in most cases due to the rounding of the data file to a 2048 byte boundary by default or because of specific padding of the data file by the user.

In the key description information, purged keys are keys that are no longer being used because a REWRITE changed a key value or for some reason a WRITE failed after the particular key was inserted and it was backed out. ICPACK will remove purged keys.

In the key attributes, no `d' means duplicates are not allowed, no `u' means that the key entry is used as given, no `r' means that the key is stored in ascending order, no `s' means there are no suppressed key values, no `a' or `o' means there are not multiple keys, and no `p' means there are no suffixes on the key.

An example of an ICSTAT output is shown in SCREEN 3.

```
icstat Revision 4.70 (UNIX)
Copyright(C) 1988-2012, Envyr Corporation  All rights reserved.
d:\test\armaster processed on 08/03/2012 at 15:52:59
    ICISAM Revision 7.00
    Delete is logical, Maximum file size is 2GB
    One Alternate Key    85 byte Records    1 Record allocated
    Records available before the .XD EOF:
    Primary Key      -------   Keys:      1  Records:      1
        Level:  1   Nodes:  1  Entries:  2  Avg Den: 2%
        11 byte key at offset 1, 127 nodes entries maximum.
    Alternate Key: 1d--s---   Keys:      1  Records:      1
        Level:  1   Nodes:  1  Entries:  2  Avg Den: 2%
        14 byte key at offset 12.
            suppressed when all bytes are LOW-VALUE, 92 node entries maxi
    Total number of indexed nodes:  2
    No Deleted Records
1 files/arguments were processed.  All are reliable.
icstat is finished.
```

SCREEN 3. ICSTAT

An example of an ICSTAT output using the No-scan switch is shown in SCREEN 4.

```
icstat Revision 4.70 (UNIX)
Copyright(C) 1987-2012, Envyr Corporation  All rights reserved.
d:\test\armaster processed on 08/03/2012 at 15:52:59
    ICISAM Revision 7.00
    Delete is logical, Maximum file size is 2GB
    One Alternate Key    85 byte Records     1 Record allocated
    Records available before the .XD EOF:
    Primary Key     -------   Keys:      0  Records:      0
       11 byte key at offset 1, 127 nodes entries maximum.
    Alternate Key: 1d--s---   Keys:      0  Records:      0
        14 byte key at offset 12.
            suppressed when all bytes are LOW-VALUE, 92 node entries maxi
1 files/arguments were processed.  All are reliable.
icstat is finished.
```

SCREEN 4. ICSTAT with No-Scan

XX. ICWEBMSG

A. **Introduction**

The ICWEBMSG utility facilitates the sending of an HTTP or HTTPS request and receiving the response via a command-line utility. The intention is that COBOL programs can use the CALL facility to execute this utility to implement various forms of web communication. The default is to send an XML request file and receive a response. ICWEBMSG acts as a very simple web client.

ICWEBMSG is currently available for Windows and Linux. Under Windows, it is part of the Full installation package and can be installed as an additional optional component. It is not installed by default.

ICWEBMSG is a stand-alone utility. It does not require the shared area allocated by ICEXEC. Under UNIX, it does not use any ICOBOL shared libraries, so it can be moved back to any previous version of **ICOBOL**. (Just remember to move the help file icwebmsg.hf, also). Under Windows, it uses the icsys.dll and two additional .dll's, ssleay32.dll and libeay32.dll.

B. **Syntax**

The standard syntax is:

```
icwebmsg [-a[:aflag]|-A file|dir[:aflag]] [-C sec] [-F value] [-h|-?]
    [-I path] [-L value] [-M GET|POST|HEAD] [-P data] [-q] [-R sec]
    [-s|-S value] [-t | -T value] [-U value] [-v] [-V 1.0|1.1] target-url
    [output-file]
```

Where
    -a[:*aflag*]|A *file*|*dir*[:*aflag*] (Audit)
        Enables auditing (default icwebmsg.lg). Where *aflag* is a|b|d|p|t|u|da|db|pa|pb|ta|tb|ua|ub, defined as a-append, b-backup, d-date, p-pid, t-time, and u-username.
    -B *username*:*password* provide Basic authorization
        passes username and password as the basic authorization string.
    -C *sec* (Connection timeout)
        Connection wait timeout in seconds (default is to wait forever)
    -E *filename* (External certificate)
        Use the external client certificate specified by *filename*.
    -F *value* (From value)
        emit From: header with "value" (default: no From: header)
    -h | -?
        display Help text
    -I *path* (Include)
        Include content of "*path*", read and send it
    -L *value* (Language header)   (POST only)
        emit content-Language: header with "value" (default: en-US)
    -M GET|POST|HEAD   (Mode)
        request Mode for HTTP (default is POST)
    -P *data*
        Emit data as POST or GET data
    -q (Quiet)
        Quiet operation
    -R *sec* (Response timeout)
        Response wait timeout in seconds (default is to wait forever)
    -s | -S *value*   (POST only)
        emit SOAPAction: header with no value or emit SOAPAction: header with "value"
    -t | -T *value* (Type header)   (POST only)

emit content-Type: header with emit "application/x-www-form-urlencoded", emit header with "value" or if
not specified at all it defaults to "text/xml"

-U *value* (UserAgent header)
emit User-Agent: header value appended after %P/%r

-v (Verbose)
Verbose logging to the audit file

-V 1.0|1.1 (Version)
Version of HTTP to use (default is 1.0)

target-url   (required)
specifies either a http or https web address.

output-file   (optional)
specifies a file for any output, if not specified STDOUT is used

Notes:
-L, -s, -S, -t, -T only apply to the POST method
-t and -T are mutually exclusive
-P and -I are mutually exclusive, and at least one is required for a POST

Environment variables:

ICWEBMSG   Command line options

C. **Description**

The first argument is required and is the target URL. The second argument is the path to the file that will receive the result, if not specified, it defaults to STDOUT. The URL must start with either HTTP or HTTPS. The Secure Sockets Layer (SSL) services are provided by the openssl project. The copyrights are included in the readme, and a snapshot of the sources used to build with are available upon request.

Several of the parameters control the various HTTP headers that are sent with the request. If the structure of an HTTP request is not familiar, there is a simple and helpful HTTP tutorial at:

```
http://www.jmarshall.com/easy/http/
```

-B *username:password*   (Basic authorization)

Sends the username password as the Basic authorization for the SSL connection.

-C *seconds*   (CONNECT WAIT TIMEOUT)

This parameter specifies the time in seconds that the utility will wait for a connection to the target-url. If no response is received within this timeframe a "Device Timeout" error will be given and the utility will terminate with an exit code of 1. Valid values are 0 to 6300. (I.E., 1 hr and 45 minutes). If not specified, ICWEBMSG will wait forever.

-E *filename*   (External certificate)

Uses the external client certificate as part of the SSL connection. Specifies a .pem file.

-F *value*   (FROM)

The From: header usually specifies the email address of the person or company responsible for sending the request. It is optional. This field is often transcribed into the web server logs. It may be helpful for debugging if the username or console number of the person making the request is supplied. This header is omitted by default.

 -I *path*   (INCLUDE)

When a POST request is sent, the data that the web server is to process must be supplied. This parameter specifies the path to a file that will be sent. The file is sent, byte by byte to the web server as the "content". There is no translation of any sort. POST is the method used by XML servers, and the content is the xml file with the <?xml version="1.0"?> line as the first line.

For a GET request, this data is sent as part of the query. The file is sent, byte by byte to the web server as part of the query. There is no translation of any sort.

 -L *value*   (LANGUAGE)

The Language-Encoding header is supplied when POST is used and indicates to the web server the language being used by the content being sent. The value supplied here is simply passed on in the header line. en-US is supplied as the default.

 -M GET|POST|HEAD   (METHOD)

This parameter selects the basic request method being used. The default value is POST.

   GET is the method used by a web browser when a URL is entered to get a page. The content that is returned is copied byte-by-byte into the output file. The returned headers are not copied.

   HEAD only returns status headers and no content. Those headers are copied to the output file instead of content.

   POST is the method that is often used by a browser to send the data that was filled out in a form. As previously mentioned, it is also often used by XML servers that communicate via HTTP.

-P *data*   (PUT data)

Just like an include except that the data is specified on the command line.

 -R *seconds*   (RESPONSE WAIT TIMEOUT)

This parameter specifies the time in seconds that the utility will wait for an initial response to its request to the target-url. If no response is received within this timeframe a "Device Timeout" error will be given and the utility will terminate with an exit code of 1. Valid values are 0 to 6300. (I.E., 1 hr and 45 minutes). If not specified, ICWEBMSG will wait forever.

 -s | -S *value*   (SOAPACTION)

This parameter specifies that the SOAPAction header should be sent and the value to supply with that header. The -s option says to send the header, but the value is empty. The -S option passes whatever value specified. This header is required by some web/SOAP servers, but it is one that has been replaced by other fields in the SOAP content. If your system doesn't need it, don't set it.

-t | -T *value*   (TYPE)

This parameter specifies the Content-Type header that is supplied when POST is used. If neither -t or -T is specified the default value is text/xml. For -t the value emitted is application/x-www-form-urlencoded. Supply whatever value is appropriate for the server and the type of request you are sending. It should be a valid MIME type or the web server may return an error.

 -U *value*   (USERAGENT)

One of the headers that is always sent is the User-Agent header. This header usually specifies the name and revision of the program that is sending the request, and it is usually logged in the web server logs. We always set it to icwebmsg/3.xx. If this parameter is specified, a space will be appended followed by the value specified. For example, if -V CreditCheck/2.0 was specified, the header will look like:

     User-Agent: icwebmsg/4.07 CreditCheck/2.0

If this is set to the name and revision of the COBOL program that calls ICWEBMSG, it will help as you debug your system.

 -v   (verbose)

This parameter indicates that ICWEBMSG is to perform verbose logging to the audit log. When this parameter is NOT set, only the data that would come out on the screen is sent to the audit log. When specified, the following additional information is logged:

    A copy of the headers that are sent
    Progress and size of any content that is sent
    Headers that are received as part of the response
    Progress and size of any content that is received
    Any trailers (possible with chunked data in HTTP/1.1) received

An item to note is the "Finished ... lines" provided by icwebmsg have a timestamp (HH:mm:ss.hh) at the end of each line. This timestamp can provide timing information in case the Connect-wait and/or Response-wait switches need to be given, or just to see the portions of time in the round-trip for each section.

General Format for verbose switch logging:

```
    Secure:      No/Yes
    Host:        xx
    Port:        n
    URL:         xx
    Input:       xx                        (for POST)
    Output:      xx
    SSL Connection informatiom...          (for Secure Sockets)
    Finished connecting:       HH:mm:ss.hh
    Writing Headers...
       method url HTTP/version
       Host:  xx
       Connection: xx
       From: xx                            (optional)
       User-Agent: xx
       SOAPAction: xx                      (optional)
       Content-Language: xx         (for POST)
       Content-Type: xx             (for POST)
       Content-Length: n            (for POST)
    Finished writing headers:  HH:mm:ss.hh
```

```
   Writing <n> bytes of content...         (for POST)
   Finished writing content:  HH:mm:ss.hh (for POST)
   Reading Response...
      ...(actual response)
   Finished reading response: HH:mm:ss.hh
   Reading headers...
      ...(actual header)
   Finished reading headers:  HH:mm:ss.hh
   Reading <n> bytes of content            (for GET and POST)
      or
   Reading content in chunks               (this section for chunk response)
   Reading chunk of <n> bytes                   .
   End of chunks                                .
   Content size was <n> bytes                   .
   Reading trailer...                      (optional)
      ...(actual trailer line(s))          (optional)
      or
   Reading content of unknown length
   Content size was <n> bytes
   Finished reading content:  HH:mm:ss.hh
```

   -V 1.0|1.1   (VERSION)

This parameter specifies the HTTP version that is set in the headers that are sent with the request be set. The default is 1.0, which should be recognized by virtually any web server. If version 1.1 is set, icwebmsg sends some additional headers needed (or desired) by the 1.1 protocol. The simple tutorial referenced above explains some of these.

Error Handling

Errors will be reported in the usual way - via stderr and into the audit file if one is specified - and they will result in a nonzero exit code. The exit code values follow the same conventions as other **ICOBOL** utilities.

We strongly suggest using an audit log - perhaps with a filename based on console number, and using the append option. That way you can see the entire set of transactions for a particular console. While you're getting the application up and running, use the -v switch to see everything going back and forth. It will let you track down most problems very quickly.

Besides errors from processing the command line, there can be errors for a failure to make a connection, i/o errors on the connection once it is made, filename processing errors, file open/create and i/o errors writing the data coming back, malformed responses or response headers, etc.

The other errors that can come back are HTTP errors, like 404 NOT Found. The utility inspects this value. Any value 300 or greater will be treated as an error. It will be reported and processing will stop after the header is read.

If there is an error and if nothing has been written yet to the output file, the output file will be deleted. If, however, some data was written, it is not deleted. Sometimes inspecting the partial result will help track down failures on the web-server side.

Some Web sites that may be helpful while debugging are:

```
   http://web-sniffer.net                       (provides an http viewer)
   http://www.rexswain.com/httpview.html        (another http viewer)
```

You can also manually experiment with HTTP by using telnet.

Using telnet, open an interactive socket to an HTTP server. Then manually enter a request, and see the response

written to the screen. It's a great help when learning HTTP, to see exactly how a server responds to a particular request. It also helps when troubleshooting.

For example, to open a connection to an HTTP server would be something like:

```
telnet www.somehost.com 80
```

Then enter your request line by line, like:

```
GET / HTTP/1.0
[headers here, if any]
blank line here
```

You may have to send <cr><nl> to end each line (Ctrl-M, Ctrl-J) and you may also want to enable echoing in the telnet to see what you are typing. Also the blank line is REQUIRED to end the request.

After finishing your request with the blank line, you'll see the raw response from the server, including the status line, headers, and message body.

D. **Examples**

Note that all the samples use the -a and -v switches to specify auditing and verbosity.

Example 1 - GET

One very simple example is using the GET method to read a standard web page. To get the default icobol web page you could use:

```
icwebmsg -av -M GET http://www.icobol.com icobol.in.htm
```

This will get the default web page from the icobol.com site and store it in the file icobol.in.htm

Example 2 - POST

Another example using the POST method would be:

```
icwebmsg -av -I request.xml http://<server>/cgi-bin/HelloService.cgi
   response.htm
```

Where the input file (request.xml) might be from the OpenSOAP project):

```
<?xml version="1.0"?>
<SOAP-ENV:Envelope
 xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Body>
        <m:Hello xmlns:m="http://services.opensoap.jp/samples/Hello/">
            <MyName>foo</MyName>
        </m:Hello>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

The response (response.htm) received back looks like:

```
<?xml version="1.0" encoding="US-ASCII"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <m:HelloResponse xmlns:m="http://services.opensoap.jp/samples/Hello/">
```

```
        <Reply>Hello, foo!</Reply>
      </m:HelloResponse>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

From the OpenSoap project.

   www.opensoap.jp

Example 3 - POST

Another example using the POST method that gets the current time from www.soapware.org

An example input file (called gettime.xml) would be (from www.soapware.org):

```
<?xml version="1.0"?>
<SOAP-ENV:Envelope
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsd="http://www.w3.org/1999/XMLSchema"
    xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance">
<SOAP-ENV:Body>
<getCurrentTime></getCurrentTime>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

The response (gettime_res.htm) received back looks like:

```
<?xml version="1.0"?>
<SOAP-ENV:Envelope
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/1999/XMLSchema"
xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance">
   <SOAP-ENV:Body>
      <getCurrentTimeResponse>
         <Result xsi:type="xsd:timeInstant">2004-07-15T07:01:38-04:00</Result>
         </getCurrentTimeResponse>
      </SOAP-ENV:Body>
   </SOAP-ENV:Envelope>
```

The command line to implement this used the defaults and would be something like this:

```
   icwebmsg -av -I gettime.xml -S \"/currentTime\" http://time.soapware.org
      gettime_res.htm
```

NOTE:
   The quote characters in the -S argument must be escaped such that the shell or command processor does not
   remove them as the entire string "/currentTime" (including the quotes) is required for the SOAPAction.

Another example is to perform a pipe-open in **ICOBOL** and read the input. For example using the sample logon
program, select U for utilities, and then L for list. Enter "|<icwebmsg -q -M GET http://www.icobol.com". This
will list the html code returned from icobol.com. The -a switch could have also been given to build an audit log.

More samples and SOAP information can be found at:

```
www.soapware.org
www.xmlrpc.com
```

XXI. ICWHOHAS

A. **Introduction**

The ICWHOHAS utility uses the shared area created by ICEXEC to show open file and/or lock information along with what process has the file opened or locked. This utility is available under both Windows and UNIX and requires that ICEXEC be running.

B. **Syntax**

The standard syntax is:

```
icwhohas [-a[:aflag]|-A file|dir[:aflag]] [-C cols] [-F file] [-h|-?]
    [-L pos[:ext]] [-O {g|i|r|s}...] [-P pid] [-q] [-T term] [-W o|l]
```

Where
   -a[:*aflag*]|A *file*|*dir*[:*aflag*] (Audit)
        Enables auditing (default icwhohas.lg). Where *aflag* is a|b|d|p|t|u|da|db|pa|pb|ta|tb|ua|ub, defined as a-append, b-backup, d-date, p-pid, t-time, and u-username.
   -C *cols* (Columns to show)
        Valid columns selections (*cols*) are: c-COBOL program, e-extent of lock, f-file name, i-identifier of process, n-number of opens, o-open status, p-position of lock, r-run state of program, s-size of the file, t-terminals (@CON), u-username. No column selector can be picked more than once.
   -F *file* (File)
        Show information about the indicated *file*.
   -h|-? (Help)
        Displays help text.
   -L *pos*[:*ext*] (Lock)
        Show only a lock at the given byte position (*pos*) and optional extent (*ext*).
   -O {g|i|r|s}... (file Organization)
        Show only files of the selected file organization: g-generic, i-indexed, r-relative, and s-sequential.
   -P *pid* (Pid)
        Show only file/locks in use by process *pid*.
   -q (Quiet)
        Enables quiet operation
   -T *number* (Terminal)
        Show only files/locks in use by terminal (@CON) *number*.
   -W o|l (What kind)
        Show only o-open files or l-locks.

Environment variables:

   ICWHOHAS   Command line options

C. **Description**

ICWHOHAS can be used as a debugging tool to check out information in the shared area maintained by ICEXEC.

If no Column switch (-C) is given, the default is "tiupof" with no What switch (-W), the default is "tiuof" if -W o and the default is "tiupf" with -W l.

If no File switch (-F *file*) is given, all selected files are shown.

If no file Organization switch (-O) is given, all files are shown.

If no Process switch (-P *pid*) is given, all pids on the selected files are shown,

If no Terminal switch (-T *term*) is given, then all terminals using the selected files are shown.

If no What switch (-W o|l) is given, then all open files and all locked records are shown.

When using the Column switch (-C) with the filename column option, the file name should be the last column since file names can be a variable length..

If nothing is displayed, an exit code of 10 is returned.

Position of lock is the byte position in the file of the lock.

Extent of lock is the number of bytes locked at that point. For a record lock it should be 1. If no extent, then a "--------" will be shown.

Column Header descriptions:

c   COBOL program names are shown with a header of "Program.........". If the process is not a runtime then the process name itself will be give, for example an icios process will show as "(icios)".

e   Extent of a record lock is shown with a header of "....Extent".

f   File names are shown with a header of "File...".

i   process Identifiers (pids) are shown with the header ".......PID".

o   Open status is shown with a header of "Status" and then I for input, O for output, E for exclusive, B for buffered. M for file has been modified, L for file is locked, and S for a remote slave file.

n   Number of opens are shown with a header of " Opens".

p   Position of a record lock is shown with a header of ".Lposition". If no record lock then a " ---------" will be shown.

r   The Run state of the program is shown with a header of " Runstate". If the run state cannot be determined an "Unknown" will be shown.

s   Size of the file is shown with a header of "......Size".

t   Terminal numbers are shown with the header ".Term". An N/A is shown when no console number can be determined. This will usually imply the process is an icios process. The program name selection will show that case as "(icios)".

u   Usernames are shown with a header of "User........".

w   lock pids are shown with a header of " ..Lock PID". If no lock a "None" is shown.

PART 2 - WITHIN COBOL

XXII. Abort Terminal (COBOL)

A. **Introduction**

The Abort Terminal function is entered from a COBOL program with a call to the IC_ABORT_TERM builtin.

The Abort Terminal function allows active COBOL terminals to be aborted either to facilitate a system shutdown or for other reasons. Upon invocation, a terminal status window of all logged-on terminals will be displayed. You are then prompted as to which terminal you wish to abort. Once that terminal is aborted, you will see the confirmation in the status window. Aborting a terminal will not remove it from the terminal status window but will mark the terminal as `Stopped' in the terminal status window.

On UNIX and Windows, this function uses the information in the shared area maintained by ICEXEC.

On UNIX, the runtime requests ICEXEC to issue a UNIX Signal of SIGUSR1 to the PID corresponding to the console number selected.

B. **Use**

Upon invocation a terminal status window of all logged on terminals is displayed. You are then prompted for the terminal number that you wish to abort.

```
                     Abort Terminal Utility Revision 4.70
 T.S.Program          T.S.Program          T.S.Program          T.S.Program
 0 I logon            2 R sample           4 R csls01           5 S csmenu
 7 I logon           11 I logon




 (S)tatus:(D)ebug,(L)ogin,(I)nactive,(P)ushed,(R)unning,(S)topped, (W)atching
 Terminal number of job to abort. __
                                 ESC Exit
```
SCREEN 5. ABORT TERMINAL

XXIII. Kill Terminal (COBOL)

A. Introduction

The Kill Terminal function is entered from a COBOL program with a call to the IC_KILL_TERM builtin.

The Kill Terminal function allows active COBOL terminals to be terminated either to facilitate a system shutdown or for other reasons. Upon invocation a terminal status window of all logged-on terminals will be displayed. You are then prompted as to which terminal you wish to terminate. Once that terminal is terminated you will see the confirmation in the status window. Killing a terminal will remove it from the terminal status window.

On UNIX and Windows, this function uses the information in the shared area maintained by ICEXEC.

On UNIX, the runtime requests ICEXEC to issue a UNIX Signal of SIGTERM to the PID corresponding to the console number selected.

B. Use

Upon invocation, a terminal status window of all logged on terminals is displayed. You are then prompted for the terminal number that you wish to kill.

```
         Terminate COBOL Process Utility Revision 4.70
 T.S.Program        T.S.Program        T.S.Program        T.S.Program
 0 I logon          2 R sample         4 R csls01         5 S
 7 I logon         11 I logon




 (S)tatus:(D)ebug,(L)ogin,(I)nactive,(P)ushed,(R)unning,(S)topped,(W)atching
 Terminal number of process to terminate. _____

                      ESC Exit
```

SCREEN 6. KILL TERMINAL

127

XXIV. Message Sending (COBOL)

A. **Introduction**

The Message Sending function is entered from a COBOL program with a call to the IC_SEND_MSG builtin.

The Message Sending function allows the user to send a message to one, several, or all logged-on Interactive COBOL users, either active or inactive on the **same** machine.

This function uses the information in the shared area maintained by ICEXEC.

B. **Use**

Upon invocation, a terminal status window of all logged on terminals is displayed. You are then prompted for the message that you wish to send. You are then prompted for the terminal number to send the message to. If none, the message is sent to all logged-on users.

```
             Message Sending Utility Revision 4.70
  T.S.Program          T.S.Program        T.S.Program        T.S.Program
  0 I logon         2 R sample       4 R csls01        5 S
  7 I logon        11 I logon




  (S)tatus:(D)ebug,(L)ogin,(I)nactive,(P)ushed,(R)unning,(S)topped,(W)atching
  Message: _____
  Terminal Number: _____
                    ESC Exit
```

SCREEN 7. MESSAGE SENDING

XXV. Printer Control (COBOL)

A. **Introduction**

The Printer Control utility is entered from a COBOL program with a call to the IC_PRINT_STAT builtin.

The Printer Control utility enables the user to view and change the current status of the print spooling system including the files in the system, the files currently queued to a print queue or printing, and the files that have been printed.

The Printer Control utility provides for the spooling and separate printing of files. The Printer Control utility uses the printer control file to hold the filenames that are currently in the printer control queue. The printer control file can handle up to 1024 files based on what ICCONFIG has allowed. Once that maximum is reached an OPEN of a file that would have been placed in the printer control file will fail with a File Status 99 (Exception Status 44).

The Printer Control utility can be configured to automatically print a file once it has been entered or to allow each file to be queued separately to a printer by a user.

The printer control file is managed by the ICEXEC process. When the printer control file is initially read at startup, all entries are checked to see if the file still exists at its specified location. If a "File not Found" error is detected then the entry is removed from the printer control file. Any other error, along with a good return, causes the entry to remain in the printer control file.

B. **Use**

B.1.   Overview

SCREENS 8, 9, and 10 show the screens that will be displayed for the Printer Control utility. SCREENS 8 and 9 are composed of three(3) different windows. These are from top to bottom 1) the file list window, 2) the file status window, and 3) the command window. These windows can be seen below alongside a partial printer control screen.

```
                                    ┌──────────────────────┐
                                    │          Printer Contro
                    1) file list    │ Filename...Status Filen
                       window       │ .
                          .         │ .
                          .         │ .
                          .         │ .
                    2) file status  a)│ Filename: <fullpath>
                       window       b)│ Size:    87654
                          .         c)│ Status: Not yet printed
                          .         d)│ Owned by:
                          .         e)│ PCQ:  1  Priority: 127
                          .         f)│ <options>...
                          .         g)│ Delete, Keep, or Remove
                    3) command      │ Command:    (Delete, Lo
                       window       │         Cursor Keys to
                                    └──────────────────────┘
```

131

```
                    Printer Control Utility Revision 4.70
Filename......Status..    Filename......Status..    Filename......Status..
artable        N000-201   rep1019        P000-205   test           N000-201
payroll        Q001-001   holdtab        *001-216   joelist        N000-210
ven            P000-090




Filename: /usr/ralph/artable
Size:    65432            Last modified: Jan-03-1994 08:23
Status: Not yet printed.  Last printed:
Owned by: ralph  (201)     Printed by: ralph    (201)
PCQ:  0  Priority: 127  Copies:  1  Notify: N
Nobanner: Y
Delete, Keep, or Remove: K  All Pages: Y
Command: _   (Delete, pdF print, Local Print, Modify options, Print, Remove, View)
CursorKeys Position, TAB Devices, Ctrl-F Filters, F1 Compress, F2 PDF, ESC exit.
```

SCREEN 8.   UNIX PRINTER CONTROL

```
                    Printer Control Utility Revision 4.70
Filename......Status..    Filename......Status..    Filename......Status..
artable        N000-201   rep1019        P000-205   test           N000-201
payroll        Q001-001   holdtab        *001-216   joelist        N000-210
ven            P000-090




Filename: c:\test200\artable
Size:    65432            Last modified: Jan-03-1994 08:23
Status: Not yet printed.  Last printed:
Owned by: ralph  (201)     Printed by: ralph    (201)
PCQ:  0  Priority: 127  Copies:  1
FF at Begin: N  FF per Copy: Y  FF at End: N
Delete, Keep, or Remove: K  All Pages: Y
Command: _   (Delete, pdF print, Local Print, Modify options, Print, Remove, View
CursorKeys Position, TAB Devices, Ctrl-F Filters, F1 Compress, F2 PDF, ESC exit.
```

SCREEN 9.   WINDOWS PRINTER CONTROL

The *file list window* is composed of at least 12 lines of 3 files apiece that show the simple filename (up to 14 characters) for the print file stored in the printer scheduler and a short-status indicator. If your ICLINES setting is greater than 24 then this window will be larger. The short-status indicator is defined as:

*qqq-ooo	(blinking *) file is currently being modified by a COBOL program
Aqqq-ooo	file has already been printed and/or submitted to be printed
Eqqq-ooo	file had been printing but was terminated before it finished
Hqqq-jjj	queued file that has been held
Nqqq-ooo	new file (not yet printed/submitted)
Pqqq-ppp	(blinking) actively printing file
Qqqq-jjj	queued file

Where
  jjj
       Is the queue entry number for that printer control queue. 1 implies it is the next file to be printed after the current one is finished, 2 is the second file, etc.
  ooo
       Is the user-id (On UNIX) or terminal number (On Windows) of the last user to modify the file, i.e., the owner
  ppp
       Is the user-id (On UNIX), terminal number (On Windows) of the last user to have printed/submitted the file
  qqq

       Is the printer control queue number for the file (0-127)

The *file status window* is the next seven lines (1-7). This window shows detailed status information for the file currently highlighted in the file list window. This detailed information includes:

    a) the full pathname of the file (up to 70 characters).
    b) the size of the file in bytes, and when the file was last modified.
    c) status information on the file such as;
        i) whether it has been printed and if so when,
        ii) while printing it shows the percentage of the file printed for each copy,
        iii) if an error occurs while printing the error is shown here;
        iv) whether the file is being updated.
    d) <u>On UNIX</u>, the username and user-id of the person that last modified the file (Owned by), along with the same information of the person that last printed the file (Printed by).
    d) <u>On Windows</u>, the username and the terminal number of the person that last modified the file (Owned by), along with the same information of the person that last printed the file (Printed by).
    e-g) the current print options.

Files can be highlighted (in reverse video) in the file list window by using the arrow keys along with HOME and END to move around.

A single HOME will move to the first entry currently displayed in the file list window, while a single END will move to the last entry currently displayed in the file list window. A HOME HOME (a HOME followed by another HOME) will move to the very first entry in the printer control queue, while an END END will move to the very last entry in the printer control queue. Most keyboard configuration files default to Ctrl-A being an END and Ctrl-P being a HOME.

The *command window* consists of the final two lines and shows the current valid commands that can be executed for the highlighted file. The first line of the command window shows Commands as described in the next section. The second line indicates that the cursor keys can be used to position to an entry, switch to the device screen, switch to the filter screen, switch to compressed mode, switch to pdf formats screen, and finally exit.

While in the main Printer Control screen the Printer Control Status screen can be shown by entering a TAB. SCREENs 10 - 13 show examples of a printer control status screens.

```
        Printer Control Utility Revision 4.70 (Unix)
Unit Device.....................  Status.........   Translation....
    0 LPT1                        Available
    1 LPT2                        Available
    3 COM1                        Not available
    4 IOP7                        Printing




        ENTER Update immediately, ESC Exit.
```
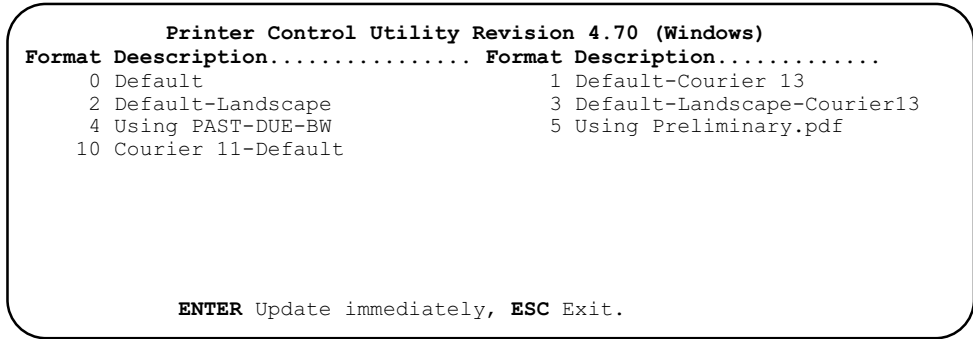
SCREEN 10. PRINTER CONTROL STATUS (Unix)

```
           Printer Control Utility Revision 4.70 (Windows)
Unit Device...................... Status.........  Translation....
   0 LPT1                         Available
   1 LPT2                         Available
   3 COM1                         Not available
   4 IOP7                         Printing




         ENTER Update immediately, ESC Exit, TAB Toggle display.
```

SCREEN 11. PRINTER CONTROL STATUS (Windows)

```
           Printer Control Utility Revision 4.70 (Windows)
Unit Device...................... Status.........  Location.......
   0 LPT1                         Available
   1 LPT2                         Available
   3 COM1                         Not available
   4 IOP7                         Printing




         ENTER Update immediately, ESC Exit, TAB Toggle display.
```

SCREEN 12. PRINTER CONTROL STATUS (Windows)

```
           Printer Control Utility Revision 4.70 (Windows)
Unit Device...................... Status.........  Comment........
   0 LPT1                         Available
   1 LPT2                         Available
   3 COM1                         Not available
   4 IOP7                         Printing




         ENTER Update immediately, ESC Exit, TAB Toggle display.
```

SCREEN 13. PRINTER CONTROL STATUS (Windows)

The printer control status screen shows all enabled printer control queues along with its specified device and a status of "Not available", "Available", or "Printing". "Not available" says that at this time this queue cannot be accessed by Interactive COBOL. "Available" says that this queue is idle and available for printing. "Printing" says that this queue is currently printing a job and is available to queue more files for printing. The Translation column shows any translation files being used for a printer. On Windows, a TAB can be used to toggle the last column among Windows Location and Windows Comment in addition to the Translation column entries for each printer device.

On Windows, the status can also be "Paused", "Offline", or "Needs attention". Also a "*" will be shown in front of the Device name for the default Windows printer. (I.E., configured to blank in the configuration file.)

While in the main Printer Control screen the PDF Formats that are available can be shown by entering a F2. The . Screen below shows an example.

```
                Printer Control Utility Revision 4.70 (Windows)
      Format Deescription................ Format Description.............
            0 Default                          1 Default-Courier 13
            2 Default-Landscape                3 Default-Landscape-Courier13
            4 Using PAST-DUE-BW                5 Using Preliminary.pdf
           10 Courier 11-Default




                 ENTER Update immediately, ESC Exit.
```

SCREEN 14. PDF FORMATS STATUS

B.2.   Printer Control Commands

Available commands within the Printer Control utility are defined in TABLE 1. Each command works on the currently selected print file although not all are allowed for every file. Only the valid commands for the selected file are shown in the command window. To execute a command, type the highlighted first letter.

To actually use a particular command it must be displayed as a valid command option and the particular user must have the appropriate access to the print job entry and possibly the associated file.

To start a file printing from the Printer Control utility, the user must issue the Print command.

Command	Printer Control Function
Cancel	Cancel the file from the print queue
Delete	Delete the file from disk storage and remove its name from the print control list
pd**F** print	Generate a PDF file from this entry, the pdf format is requested
Hold	Hold the file in the print queue and keep it from printing
Local print	Print the file to the locally attached printer
Modify options	Change the current print options for PCQ; Priority; Number of copies; Notify; Disposition options of Delete, Keep, or Remove the file after printing; print All pages or the Starting and Ending pages. (On Windows) form-feed options of at Beginning, per Copy, and at End; or (On UNIX) Nobanner;
Print	Print the file with its current options
Remove	Remove the filename from the print control list
Terminate	Interrupts the printing file
Unhold	Unhold the file in the print queue
View	View the file on the screen

TABLE 1. PRINTER CONTROL COMMANDS

On Windows, any user with printer control management privilege can perform any operation to a file provided he has access to that file from the operating system. If a username has been stored into the printer control file for either owner or printer then it is used when matching on owner or printer, otherwise the owner's and printer's terminal number is used. The username convention will allow a given username to operate on files from different terminals on the system.

On UNIX, super user or any user with printer control management privilege can perform any operation to a file that

he has the appropriate access to. Other conditions will be specified with each commands description below. When a file is placed into printer control, the current user-id and group-id are stored as the owner-user-id and owner-group-id. When an entry is printed, the current user-id and group-id are stored as the printed-by-id and printed-by-group-id.

Cancel removes a file from the print queue to keep it from printing. You cannot Cancel the file currently printing; instead, it must be terminated to stop further printing. On UNIX, a user must be the owner or in the owner's group.

pdf print provides the ability to generate a .PDF file using a particular PDF Format that is requested.

Delete removes the file from the printer control file and deletes the file from storage. The user will be prompted with a confirmation message before the delete is done. If the file no longer exists, Delete gives an error and the Remove command must be used to remove the entry. On UNIX, a user must be the owner, or in the owner's group and have delete permission (i.e., write access to the directory that the file is in) on the file.

Hold will temporarily keep a file from printing but keep it queued and unavailable for anyone to modify. A user can Hold those files printed by him or owned by him, and on UNIX in the same group as the owner or the one who printed it.

Local print sends the file to the printer attached to the current console using printer-pass-thru. It uses the current settings for Start and End pages set with the Modify options. A console interrupt can be used to terminate a local print. Files that are printed locally will not be shown as Already printed.

Modify options is used to change any of the print options for a particular file. When the Modify options command is selected you are positioned to the Options section of the file status window. You can then fill in the form to define how you wish to print this particular file. Up-arrow and down-arrow can be used to move back and forth in the options form. On UNIX, only the owner, or the owner's group can change the Delete or Remove part of the options, the other options can be changed if you have read access to the file.

The modifiable options are:

> *PCQ* can be any available PCQ from 0 - 2047. The default is the value that was placed in the printer control file when the entry was created.
> *Priority* can be a number from 0 (the highest) to 39 (the lowest) on UNIX, or 1 (the highest) to 99 (the lowest) on Windows . The default is 127 and no priority respectively. On UNIX, Interactive COBOL uses the `-q' to set the priority to the UNIX spooler if a priority is given.
> *Copies* can be a number from 1 to 9999 for the number of copies to print. The default is 1.
> *Notify* can be either Yes or No to specify whether the printer control utility should send a message to the user who printed the file when the file has successfully printed. The default is N.
> On Windows, *FF at Beginning* can be either Yes or No and instructs the printer control utility to generate an initial Form-Feed before printing the file. The default is that specified in ICCONFIG for this particular printer control queue.
> On Windows, *FF per Copy* can be either Yes or No and instructs the printer control utility to generate a Form-Feed at the end of each copy. The default is that specified in ICCONFIG for this particular printer control queue.
> On Windows, *FF at End* can be either Yes or No and instructs the printer control utility to generate a final Form-Feed after printing the file. The default is that specified in ICCONFIG for this particular printer control queue.
> On UNIX, *Nobanner* can be either Yes or No to specify whether the nobanner option should be specified to the UNIX spooler. The default is that specified in ICCONFIG for this particular printer control queue.
> Disposition of *Delete, Keep, or Remove* can be D, K, or R to specify whether to Delete the entry and remove it from the printer control file, Keep the file and the printer control file entry, or Remove the printer control file entry upon a successful completion of printing the file. The default is that specified in ICCONFIG for that particular printer control queue.
> *All pages* can be either Yes or No on whether to print the entire file. If no is given then the utility prompts for *Start page* and *End page*.

If a user is changing a file's options, no other user will have access to that file. That second user will get the message "The item is currently in use" indicating another user has it.

Print will queue the file for printing with the currently displayed options. To print with a different set of options, the Modify options command must first be used. On a Print, the Printer Control utility opens the file and verifies that it is available and there is a valid start page. If there is an error in Print, an error message is displayed in the status section for that particular file entry. A Print of a file with zero length will generate an error saying the start page could not be found. A user must have read access to the file to print it.

Remove removes the file from the printer control file. Remove can be used to remove a non-existent file from the printer control file. On UNIX, only the owner, or anyone in the owner's group can Remove it.

Terminate will terminate the currently printing file. Until the terminate is complete. the status for that particular printer will show that a Termination is being done. A user can only Terminate files that he owns or printed. On UNIX, a user must be super user, the owner, or in the owner's group.

Unhold releases a held job and allows it to print. A user can Unhold a file if he is the owner, the one who printed it, or (On UNIX) in the same group as the owner or the one who printed it.

View allows for a file to be viewed on the screen. A user can View a file if he has read access to the file. The View command uses the file list window to show portions of the selected file. The user will be prompted at the bottom to continue with selected options. Generally the options are *newline* for next screen, *l* for next line, *h* for next half screen, *e* for the end screen, and *q* to quit viewing the file. On a page break (i.e., a form-feed), the message will change to ask whether you wish to see the next page.

B.3.   Printer Control Utility Display Filtering

The set of files displayed in the Printer Control Utility may be filtered to display only a subset of all files available. The display may be temporarily altered by using the filter menu of the Printer Control Utility. This menu is accessed by pressing the Word Forward Key (usually Ctrl-F) on the main Printer Control Menu. Changes made at this menu apply only during the current session of the Printer Control Utility and do not persist after the utility has exited. If a default filter is already applicable (either from the ICPCQFILTER environment variable or because the utility was entered with one from the IC_PRINT_STAT builtin function), its values are pre-loaded into the filter screen.

The filter menu screens are shown in the following screens:

```
           Printer Control Utility Revision 4.70

   Filter           Enable  Values

   PCQ number          N    Min: ___        Max: ___
   Owner name          N    Username: _____
   Owner user-id       N    Min: ____       Max: ____
   Printed-by name     N    Username: _____
   Printed-by user-id  N    Min: ____       Max: ____
   Simple filename     N    Filename: _____
   File size           N    Min: _____   Max: _____
   Read access         N
   Directory name      N    Dir: _____
   Status              N    Status: _

      Status codes:  Not yet printed  Update in progress  Printing
                     Already printed  Queued to print     Retrying
                     Error occurred   Holding in queue    Terminating


      The filter values have changed.  Apply the new filter? _

                      ESC Exit.
```

SCREEN 15. UNIX FILTER MENU

```
                    Printer Control Utility Revision 4.70

     Filter            Enable  Values

     PCQ number          N     Min: ___          Max: ___
     Owner name          N     Username: _____
     Owner user-id       N     Min: ____        Max: ____
     Printed-by name     N     Username: _____
     Printed-by user-id  N     Min: ____        Max: ____
     Simple filename     N     Filename: _____
     File size           N     Min: _____ Max: _____
     Read access         N
     Directory name      N     Dir: _____
     Status              N     Status: _

       Status codes:  Not yet printed  Update in progress  Printing
                      Already printed  Queued to print     Retrying
                      Error occurred   Holding in queue    Terminating


     The filter values have changed.  Apply the new filter? _

                         ESC Exit.
```

SCREEN 16. WINDOWS FILTER MENU

The following items are available for controlling the filters. If any is specified then only those files which meet the specified criterion will be displayed. If more than one of these items is specified, then only files which meet ALL of the specified criteria will be displayed:

   (a) a range of printer control queues (PCQs) from 0 to the maximum configured PCQ;
   (b) the username of the job owner;
   (c) a range of owner values (console numbers on Windows and user-id numbers on UNIX);
   (d) the username of the last user who printed the file;
   (e) a range of printed-by values (console numbers on Windows, user-id numbers on UNIX);
   (f) the simple filename of the print job;
   (g) a range of file sizes;
   (h) whether or not the user of the printer control utility has read access to the file. (This option may make the Printer Control Utility appear somewhat sluggish when there are a large number of files in the queue as each file must be queried as to its read status.);
   (i) the current status of the job;
   (j) a directory holding the print job or a subdirectory that holds the print job.

To exit the filter menu, press the ESC key. If no changes have been made to the filter, the main screen of the Printer Control Utility will be displayed. If any changes have been made, the following prompt will appear:

   The filter values have changed. Apply the new filter?

Answer yes (Y) to activate the new values and return to the main screen of the Printer Control Utility. Answer no (N) to ignore the new values and return to the main screen of the Printer Control Utility. Pressing ESC at this prompt will return to the enable column of the PCQ number line.

B.4.   Notes

The Printer Control utility allows multiple files to be queued to a printer at a time. Each file is scheduled to print in the order in which it was queued with the print command unless its priority is set to a lower value (i.e., higher priority) than a preceding job. In that case the file is moved in front of jobs with a higher number. As soon as the print file ahead of it is finished, it will start printing.

Entering a space or a newline while at the command prompt will refresh the screen with updated information, otherwise the screen is refreshed every ten seconds.

To keep a file from printing that is already queued, you can either Hold it, which will keep the file from printing but keep it in the queue and unavailable for modification; or Cancel it, in which case the file is no longer queued and returns to the status that it had before it was queued to the printer.

Remember, the printer control file can only hold the number of files set in ICCONFIG's System Parameters before subsequent OPEN's will fail with a File Status 99. Thus it would be wise to frequently check to see that print files have been either removed or deleted from the queues.

If the Printer Control utility gets a printer error, the appropriate entry is marked in error and anyone checking the Printer control utility will see the error and be able to correct it. The Printer Control utility will continue to retry on printer errors until the error condition is removed or the print job is terminated.

On UNIX, the user-id for a particular user can be determined by looking in the file /etc/passwd for the user-name. The default group-id for a user is also given in the /etc/passwd file. For a particular group-id, its name can be determined by looking in the file /etc/group along with the users who are allowed access to the group. In addition the UNIX groups command can be used.

XXVI. System Information (COBOL)

A. **Introduction**

The System Information function is entered with a call to the IC_SYS_INFO builtin.

The System Information function provides system information for the entire Interactive COBOL system that includes current values, maximum values encountered since invocation, and the maximum value configured for many tunable configuration parameters.

This function uses the information in the shared area maintained by ICEXEC.

B. **Use**

For System Information, Interactive COBOL provides a screen of statistical information about various Interactive COBOL parameters. For the named resource, three numbers are displayed. These are:

```
In Use      is the number currently in use
MaxUsed        is the most this has ever been, for this invocation
Max            is the maximum number configured
```

The MaxUsed values can be used to either raise or lower individual System Parameters in ICCONFIG, or in the UNIX Kernel (On UNIX) to provide a better tuned system.

```
        System Information Display Utility Revision 4.70

    Resource                     In Use  MaxUsed       Max

    Process Count:                    1        1       156
    Terminal Count:                   1        1       132
    Run Program Terminals:            1        1       132
      Detached/CGI Terminals:         0        0         3
    SEQUENTIAL Files:                 1        1       200
    INDEXED Files:                    1        1       100
    RELATIVE Files:                   1        1         8
    Record locks:                     0        3      1024
    Unique UNIX files:                2        3      1040
    @PRN devices:                     0        1        14
    @PCQ devices:                     0        2         9
    @SER devices:                     0        1         9
    @CON devices:                     1        1       132
    Buffers (KB) assigned:           32      560      5000
    Buffers (KB) accessed:           32       48      5000
    Unique UNIX devices:              1        1       467



    ENTER Update immediately, ESC Exit.
```

SCREEN 17. UNIX SYSTEM INFORMATION

```
              System Information Display Utility Revision 4.70

        Resource                    In Use   MaxUsed      Max

        Process Count:                  1         1       156
        Terminal Count:                 1         1       132
        Run Program Terminals:          1         1       132
          Detached/CGI Terminals:       0         0         3
        SEQUENTIAL Files:               1         1       200
        INDEXED Files:                  1         1       100
        RELATIVE Files:                 1         1         8
        Record locks:                   0         3      1024
        Unique WINDOWS files:           2         3      1040
        @PRN devices:                   0         1        14
        @PCQ devices:                   0         2         9
        @SER devices:                   0         1         9
        @CON devices:                   1         1       132
        Buffers (KB) assigned:         32       560      5000
        Buffers (KB) accessed:         32        48      5000



        ENTER Update immediately, ESC Exit.
```

SCREEN 18. WINDOWS SYSTEM INFORMATION

XXVII. Terminal Control (COBOL)

A. **INTRODUCTION**

The Terminal Control function is entered with a call to the IC_TERM_CTRL builtin function.

This function allows the user to view the status of all Interactive Cobol runtime users on the system. The display includes detailed information on the highlighted terminal, including the terminal number, user name, program name, process-id, internal id, and program status. The privilege settings for the terminal are also displayed.

With the appropriate privileges, the user can perform the following actions on individual terminals:

    * goto a particular terminal
    * abort a terminal
    * kill a terminal
    * send a message to one or more terminals
    * control a terminal (takes over the terminal's keyboard and watches the screen)
    * watch a terminal's screen

This function uses the information in the shared area maintained by ICEXEC.

B. **USE**

The Terminal Control Function displays information in two windows called the terminal status window and the terminal details window. The user enters commands through a third window called the command window.

The terminal status window contains a display for each logged-in terminal. The display contains the terminal number, a status indicator, and either the program name or the PID and the username. An asterisk is printed if the terminal is being watched. The program or PID/username views can be toggled between with the TAB key. The terminals are displayed in ascending order by terminal number. Initially the terminal displayed at the upper left corner is highlighted in reverse video indicating that is the selected terminal. Terminals can be highlighted by using the Goto command to specify the terminal number or by using the arrow keys along with HOME and END to move around. Up to 3 terminals are displayed per line. If more terminals are logged on than can be shown in one screen, then indicators show whether more terminals are found before and/or after the currently displayed group. The size of the screen is used to display the most possible status lines.

Pressing the HOME key highlights the terminal displayed at the upper left corner of the current terminal status window. Pressing the HOME key two times in a row (HOME HOME) causes the lowest-numbered logged-on terminal to be displayed in the upperleft corner.

Similarly, pressing the END key highlights the terminal displayed at the lower-right corner of the current terminal status window. Pressing the END key two times in a row (END END) causes the highest-numbered logged-on terminal to be displayed in the lower-right corner.

The terminal details window shows details for the highlighted terminal including terminal number, status, complete program name, user name, process- id, privilege settings, and whether the terminal is watching another terminal or is being watched. The privileges are shown with letters indicating the current privileges available as:

    A - Abort terminal privilege,   I - System information privilege,
    M - Message sending privilege,   T - Terminal status privilege,
    P - Printer control privilege,   C - printer Control management privilege,
    S - System Shutdown privilege,   O - Detach/Host programs privilege,
    B - console interrupt privilege,   D - Program debugging privilege, and
    W - Watch other terminals privilege   X - eXclude this terminal from being watched.

143

If the highlighted terminal is involved in a watch relationship, the other terminal's number is displayed. The details window also displays the number of terminals currently logged on.

The terminal running the function is indicated in bold. When this terminal is the highlighted terminal, the additional note "This is your terminal" is displayed in the terminal details window.

The display updates approximately every 10 seconds. Pressing the Enter or Newline key causes an immediate update. The program is exited by pressing ESC.

The command window displays a menu of available commands, depending on the privileges of the user running the program. A command is chosen by typing the 1st letter of its name (case does not matter). The possible commands are:

* Goto - always available. Prompts for a terminal number, and makes it the highlighted terminal. If the terminal is not visible on the current screen and its terminal number is lower than any displayed terminal, the screen is painted with the selected terminal highlighted in the upperleft corner. If the terminal number is higher than any displayed terminal, the screen is repainted with the selected terminal highlighted in the bottom right corner.

* Abort - available to users with the Abort terminal privilege. Allows the user to abort the highlighted terminal. The user is prompted with the highlighted terminal's number and asked to confirm that it is to be aborted. If the user responds by typing a Y, an Abort is sent to the highlighted terminal. Note that it is possible to abort your own terminal.

* Kill - available to users with the Abort terminal privilege. Allows the user to kill the highlighted terminal. The user is prompted with the highlighted terminal's number and asked to confirm that it is to be terminated. If the user responds by typing a Y, a Stop is sent to the highlighted terminal. Note that it is possible to kill your own terminal.

* Message - available to users with the Message sending privilege. This command puts the user in message mode. The user is prompted to enter a brief (55 byte) message and a terminal number to be sent the message; the default is the highlighted terminal. To send the message to a different terminal, type in the terminal's number. To send to all terminals, type spaces over the default terminal number.

After the message is sent, the user can enter another terminal number to receive the same message or can press ESC once to return to the message prompt. Pressing ESC there exits message mode. If another message is entered, the highlighted terminal is again the default.

* Control - Allows the highlighted terminal to be controlled. The user's display is repainted with the highlighted terminal's screen. All subsequent output displayed on the highlighted terminal's screen is displayed on the user's screen as well. Anything the user types at his terminal is typed on the highlighted terminal. The keyboard becomes inactive on the highlighted terminal.

Intr and Quit are NOT sent to the highlighted terminal, but act on the current user. The refresh key (usually Ctrl-U) also is not sent to the highlighted terminal but acts locally to repaint the screen.

* Watch - Allows the user to view the highlighted terminal's screen. The user's display is repainted with the highlighted terminal's screen. All subsequent output displayed on the highlighted terminal's screen is displayed on the user's screen as well. In this mode the user cannot enter data to the highlighted terminal.

* Status bar - Allows the user to place the Watch/Control status indicator. Valid selections are none, left, or right. Left is the default. If visible, the status indicator is shown in reverse on the top of the user's screen.

The Watch facility (Control, Watch, and Statusbar) are only available if the following conditions are meet:

- The Watch license option is included with the **ICOBOL** Runtime License

- The Watch other terminals privilege is enabled for this user

- Console Interrupts are enabled for this user

Watch and Control cannot be used on the current console. I.E., you cannot watch yourself.

Watch and Control cannot be used on another Watcher.

Control requires that ICLINES and ICCOLUMNS on the current terminal be at least that of the terminal to be controlled. When Watching, if the ICLINES and/or ICCOLUMNS are smaller then some data will not be shown.

To quit the Watch command or Control command the Intr key (or Quit) should be pressed. The Watcher immediately returns to the Terminal Control main menu and the Watchee stops sending screen data to the Watcher. In addition, for the Control command the keyboard is switched back to the Controlee.

Data inside a Printer Pass Thru-ON - Printer Pass Thru-Off sequence is not sent to the Watcher/Controller.

If the Watchee/Controlee program pushes to another executable, the Watcher will pend until that executable returns to the runtime. A message will be displayed on the Watcher's screen indicating that the Watchee has pushed off. This message is cleared when the Watchee returns from the pushed program. For a Control command, the keyboard has been switched back to the Controlee while in the pushed program. The keyboard is re-switched back to the Controller when the Controlee returns from the pushed program.

If the Watchee/Controlee terminates in some fashion such that the runtime shuts down, a message will be displayed on the Watcher's screen indicating that the Watchee has terminated and showing the last information sent to the screen. A newline will then return the Watcher to the main menu of Terminal Control.

If the Watchee program is using drawlines characters, those characters do not get translated by the Watcher. Funny characters may be displayed on the Watcher screen.

When a Watcher connects to a Watchee, if ICEXEC has logging enabled an Info message is written to the log.

When Watching a ThinClient and gui mode is in use, that information is NOT watched.

Sample screens:

SCREEN 19 would be seen by a user who has the Terminal status privilege and Message sending privilege, but not the Abort terminal privilege or Watch other terminals privilege:

```
                    Terminal Control Utility Revision 4.70
      T.S.Program               T.S.Program               T.S.Program
      0 R termctrl              1 R cycle0                2 R cycle0
      3 R subpgm1               4 R subpgm1               5 R subpgm1
      6 R cycle0                7 R subpgm1               8 R cycle1
      9 R cycle2              10 R cycle3               11 R cycle2
     12 R cycle3              13 R cycle2               14 R cycle3
     15 R subpgm2             16 R cycle2               17 R subpgm1
     18 R subpgm1             19 R subpgm1              20 R cycle5
     21 R cycle4              22 R cycle5               23 R cycle5
     24 R subpgm2             25 R subpgm2              26 R cycle6
     27 R subpgm2             28 R cycle6               29 R subpgm2
     30 R cycle6              31 R cycle7               >>>>> MORE >>>>>
  Terminal:    0         Username:  carl
  Program:  termctrl                           Process id: 20284
  Status:  Running
  Privileges: _IM_PC_OBD_                      Active Terminals: 40
          Command: _  (Goto, Message)
          Cursor Keys Position, TAB Toggle display, ESC Exit.
```
SCREEN 19. TERMINAL CONTROL (Programs)

145

SCREEN 20 is just like SCREEN 19 above but shows the other view of the screen that is available with the TAB key:

```
                  Terminal Control Utility Revision 4.70
    T.S...PID.Username       T.S...PID.Username       T.S...PID.Username
    0 R 20284 carl           1 R 17855 ralph          2 R 18212 mike
    3 R 17865 jeremy         4 R 25999 mikeb          5 R 25669 ralph
    6 R 28123 quint          7 R 24768 alice          8 R 25668 dan
    9 R 15000 isaac         10 R 20112 ted           11 R 17875 mary
   12 R 17882 brooks        13 R 17884 freddy        14 R 17886 fred
   15 R 17888 jan           16 R 17889 butch         17 R 17890 lou
   18 R 17892 coleman       19 R 17894 becky         20 R 17896 gerry
   21 R 17898 rowland       22 R 20997 carl          23 R 17900 john
   24 R 17904 test1         25 R 17906 jim           26 R 17910 ed
   27 R 17920 root          28 R 17922 carl          29 R 25666 barbara
   30 R 18100 bill          31 R 18102 cathy         >>>>> MORE >>>>>
Terminal:    0        Username:  carl
Program:  termctrl                                   Process id: 20284
Status:  Running
Privileges: _IM_PC_OBD_                              Active Terminals: 40
       Command:    (Goto, Message)
       Cursor Keys Position, TAB Toggle display, ESC Exit.
```

SCREEN 20. TERMINAL CONTROL (Username)

SCREEN 21 would be seen by a user who has the Terminal status privilege, Message sending privilege and Abort terminal privilege, but not the Watch other terminals privilege:

```
                  Terminal Control Utility Revision 4.70
    T.S.Program              T.S.Program              T.S.Program
    0 R termctrl             1 R cycle0               2 R cycle0
    3 R subpgm1             4 R subpgm1              5 R subpgm1
    6 R cycle0              7 R subpgm1              8 R cycle1
    9 R cycle2             10 R cycle3              11 R cycle2
   12 R cycle3             13 R cycle2              14 R cycle3
   15 R subpgm2            16 R cycle2              17 R subpgm1
   18 R subpgm1            19 R subpgm1             20 R cycle5
   21 R cycle4             22 R cycle5              23 R cycle5
   24 R subpgm2            25 R subpgm2             26 R cycle6
   27 R subpgm2            28 R cycle6              29 R subpgm2
   30 R cycle6             31 R cycle7              >>>>> MORE >>>>>
Terminal:    0        Username:  carl
Program:  termctrl                                   Process id: 20284
Status:  Running
Privileges: AIM_PC_OBD_                              Active Terminals: 40
       Command:    (Goto, Abort, Kill, Message)
       Cursor Keys to position, TAB to toggle display, ESC to Exit.
```

SCREEN 21. TERMINAL CONTROL

SCREEN 22 would be seen by a user who has the Terminal status privilege, Message sending privilege, Abort terminal privilege, and Watch other terminals privilege. Notice the setting for Watch is now available.

```
                    Terminal Control Utility Revision 4.70
      T.S.Program              T.S.Program              T.S.Program
      0 R termctrl            1 R cycle0               2 R cycle0
      3 R subpgm1             4 R subpgm1              5 R subpgm1
      6 R cycle0              7 R subpgm1              8 R cycle1
      9 R cycle2             10 R cycle3              11 R cycle2
     12 R cycle3             13 R cycle2              14 R cycle3
     15 R subpgm2            16 R cycle2              17 R subpgm1
     18 R subpgm1            19 R subpgm1             20 R cycle5
     21 R cycle4             22 R cycle5              23 R cycle5
     24 R subpgm2            25 R subpgm2             26 R cycle6
     27 R subpgm2            28 R cycle6              29 R subpgm2
     30 R cycle6             31 R cycle7              >>>>> MORE >>>>>
   Terminal:    0         Username:  carl
   Program:  termctrl                            Process id: 20284
   Status:  Running
   Privileges: AIMTPC_OB_W                       Active Terminals: 40
     Command:   (Goto, Abort, Kill, Message, Control, Watch, Status bar:L)
           Cursor Keys Position, TAB Toggle display, ESC Exit.
```

SCREEN 22. TERMINAL CONTROL

SCREEN 23 shows the user with terminal 11 is watching another terminal, and terminal 22 is being watched.

```
                    Terminal Control Utility Revision 4.70
      T.S.Program              T.S.Program              T.S.Program
      0 R termctrl            1 R cycle0               2 R cycle0
      3 R subpgm1             4 R subpgm1              5 R subpgm1
      6 R cycle0              7 R subpgm1              8 R cycle1
      9 R cycle2             10 R cycle3              11 W termctrl
     12 R cycle3             13 R cycle2              14 R cycle3
     15 R subpgm2            16 R cycle2              17 R subpgm1
     18 R subpgm1            19 R subpgm1             20 R cycle5
     21 R cycle4             22 R*cycle5              23 R cycle5
     22 R*glupd01            25 R subpgm2             26 R cycle6
     27 R subpgm2            28 R cycle6              29 R subpgm2
     30 R cycle6             31 R cycle7              >>>>> MORE >>>>>
   Terminal:   22         Username:  carl
   Program:  cycle5                             Process id: 20997
   Status:  Running                            Watched by terminal 11
   Privileges: _IM_PC_OBD_                      Active Terminals: 40
   Command: _    (Goto, Message)
             Cursor Keys Position, TAB Toggle display, ESC Exit.
```

SCREEN 23. TERMINAL CONTROL

Watch Facility:

The Watch Facility provides the ability to either just view the screen of a particular runtime user or to take over control of the keyboard, as well of that user for any terminal currently running the runtime. This includes direct attached as well as pseudo-ttys. The Watch Facility works with dissimilar terminal types by mapping the needed control codes to the appropriate one's for each terminal type including both input and output. The Watch Facility uses a special privilege that allows only selected users to use the facility.

The Watch Facility is useful as a debugging tool when building COBOL programs that run in a detached (no terminal) state.

The Watch Facility is offered as a licensed option for **ICOBOL** Runtime licenses.

The Watch Facility is accessed through the Terminal Control Utility which is the builtin IC_TERM_CTRL. For a user to use the Watch Facility a new privilege has been added to the Program Environments in ICCONFIG. The Watch other terminals privilege allows the indicated user to Watch and Control another user's terminal. Console 0

defaults to enabled for new configurations while all other consoles default to not enabled.

When the user starts to Watch another terminal either with the Watch command or Control command, the current terminal becomes the "Watcher" and the indicated terminal becomes the "Watchee". The Watchee immediately sends the Watcher its current screen image which the Watcher reformats and displays on its screen using its terminal type settings. As characters are displayed to the Watchee they are transmitted to the Watcher to be displayed.

To quit the Watch command or Control command the Intr key (or Quit) should be pressed. The Watcher immediately returns to the main menu of Terminal Control and the Watchee stops sending screen data to the Watcher. In addition, for the Control command the keyboard is switched back to the Watchee.

While in the Watch command, the Watcher can not interact with the program in progress on the Watchee. The Watcher is an observer.

While in the Control command, the Watcher has control of the keyboard for the program in progress on the Watchee. The Watchee is an observer. Anything the Watcher types (except for Intr or Quit) is sent to the program as if it came from the Watchee's keyboard. If the Watchee types on his keyboard no action is performed (the keystrokes are ignored).

Because of the ability to Watch dissimilar terminal types, the Watcher must be aware that his screen may not totally match the Watchee because of different ICLINES/ICCOLUMNS or attribute settings. The Watch Facility relays the attribute commands from the Watchee to the Watcher but it does not know how each terminal will actually interpret each command.

Data inside a Printer Pass Thru-ON - Printer Pass Thru-Off sequence is not sent to the Watcher.

If the Watchee program pushes to another executable the Watcher will pend until that executable returns to the runtime. A message will be displayed on the Watcher's screen indicating that the Watchee has pushed off. This message is cleared when the Watchee returns from the pushed program. For a Control command the keyboard has been switched back to the Watchee while in the pushed program. The keyboard is re-switched back to the Watcher when the Watchee returns from the pushed program.

If the Watchee program terminates in some fashion such that the runtime shuts down, a message will be displayed on the Watcher's screen indicating that the Watchee has terminated and showing the last information sent to the screen. A newline will then return the Watcher to the main menu of Terminal Control.

XXVIII. Terminal Status (COBOL)

A. **Introduction**

The Terminal Status function is entered with a call to the IC_TERM_STAT builtin.

The Terminal Status function allows the user to view the status of all Interactive COBOL runtime users on the machine as well as current system information.

This function uses the information in the shared area maintained by ICEXEC.

B. **Use**

The status of each logged-on terminal is displayed in the status window as the terminal number, status, and program name. SCREEN 24 shows a sample terminal status screen. On the header line T is for terminal, S for status, and Program for the Program name. Up to 4 terminals will be displayed per line for up to 17 lines, allowing a total of 68 terminals per screen to be shown. If more than 68 jobs are active, then multiple passes of the terminal status screen will be shown by pressing an enter to move to the next screen full of terminals or waiting 10 seconds for the screen to cycle to the next screen.

```
       Terminal and Resource Status Utility Revision 4.70
T.S.Program       T.S.Program       T.S.Program       T.S.Program
0 I logon        2 R sample        4 R csls01        5 S usnews
7 I logon       11 I logon




(S)tatus:(D)ebug,(L)ogin,(I)nactive,(P)ushed,(R)unning,(S)topped,(W)atching
3 of 32 INDEXED files. 8 of 32 SEQUENTIAL files. 8 of 8 RELATIVE
6 of  9 Terminals.    14 of 48 Record locks.     0 of 3856KB buffers
        ENTER Update immediately. ESC Exit.
```

SCREEN 24. TERMINAL STATUS

Terminal numbers (T) are 0 through the highest supported under this system depending on how the logical consoles are configured. The terminal number will only show the lower three digits if the terminal number is greater than 999.

The status (S) position will be set to the following:

D   while actually in the debugger,
I   while inactive (i.e., those in the LOGON program via IC_LOGON or after a STOP RUN and a newline was entered),
L   while in the process of bringing up Interactive COBOL or a utility,
P   while the process is executing an O/S executable,
R   while running or active,
S   while stopped, (i.e., programs that have been aborted or otherwise terminated), and
W   while watching another terminal.

Program is the name of the currently executing COBOL program which may be stopped.

At the bottom of the terminal status display some system information is given. This includes the number of indexed, sequential, and relative files in use and available, the number of ICISAM record locks in use and available, the number of terminals in use and available, and the amount of buffer memory in use and available.

The current revision of Interactive COBOL is displayed on the top line.

149

The number of terminals available is determined at invocation and is the smaller of the maximum number of processes and the maximum number of enabled consoles with the run program option enabled.

While in Terminal Status, pressing a newline will immediately refresh the screen with updated information, otherwise, every ten seconds the screen will be updated. When more than 68 Interactive COBOL processes are active, newline will move to the next screen of terminals.

To exit from Terminal Status hit ESC.

APPENDICES

APPENDIX A.   ASCII CODES

Dec	Oct	Hex	DG Function	Ctrl-code	PC Function/Character	
0	000	00	Null	Ctrl @	NUL	space
1	001	01	Print Screen Form	Ctrl A	SOH	☺
2	002	02	Reverse off	Ctrl B	STX	☻
3	003	03		Ctrl C	ETX	♥
4	004	04		Ctrl D	EOT	♦
5	005	05	Read cursor address	Ctrl E	ENQ	♣
6	006	06		Ctrl F	Ack	♠
7	007	07	Bell	Ctrl G	Bell	●
8	010	08	Cursor Home	Ctrl H	Backspace	◘
9	011	09		Ctrl I	HTab	○
10	012	0A	Newline	Ctrl J	Linefeed	◎
11	013	0B	Erase EOL	Ctrl K	VTab	♂
12	014	0C	Erase Screen	Ctrl L	Form-feed	♀
13	015	0D	Carriage Return	Ctrl M	Carriage Return	♪
14	016	0E	Blink ON	Ctrl N	SO	♫
15	017	0F	Blink off	Ctrl O	SI	☼
16	020	10	Write cursor addr(c,r)	Ctrl P	DLE	►
17	021	11	Print Screen	Ctrl Q	DC1 (XON)	◄
18	022	12	Roll Enable	Ctrl R	DC2	↕
19	023	13	Roll Disable	Ctrl S	DC3 (XOFF)	‼
20	024	14	Underscore ON	Ctrl T	DC4	¶
21	025	15	Underscore OFF	Ctrl U	NAK	§
22	026	16	Reverse On	Ctrl V	SYN	▬
23	027	17	Cursor Up	Ctrl W	ETB	↨
24	030	18	Cursor Right	Ctrl X	CAN	↑
25	031	19	Cursor Left	Ctrl Y	EM	↓
26	032	1A	Cursor Down	Ctrl Z	SUB	→
27	033	1B	Escape	Ctrl [	ESC	←
28	034	1C	Dim ON	Ctrl \	FS	∟
29	035	1D	Dim OFF	Ctrl]	GS	↔
30	036	1E	Command Header	Ctrl ^	RS	▲
31	037	1F		Ctrl _	US	▼

Dec	Oct	Hex	DG	PC	Dec	Oct	Hex	DG	PC	Dec	Oct	Hex	DG	PC
32	040	20	space	space	64	100	40	@	@	96	140	60	`	`
33	041	21	!	!	65	101	41	A	A	97	141	61	a	a
34	042	22	"	"	66	102	42	B	B	98	142	62	b	b
35	043	23	#	#	67	103	43	C	C	99	143	63	c	c
36	044	24	$	$	68	104	44	D	D	100	144	64	d	d
37	045	25	%	%	69	105	45	E	E	101	145	65	e	e
38	046	26	&	&	70	106	46	F	F	102	146	66	f	f
39	047	27	’	’	71	107	47	G	G	103	147	67	g	g
40	050	28	(	(	72	110	48	H	H	104	150	68	h	h
41	051	29	)	)	73	111	49	I	I	105	151	69	i	i
42	052	2A	*	*	74	112	4A	J	J	106	152	6A	j	j
43	053	2B	+	+	75	113	4B	K	K	107	153	6B	k	k
44	054	2C	,	(comma),	76	114	4C	L	L	108	154	6C	l	l
45	055	2D	−	−	77	115	4D	M	M	109	155	6D	m	m
46	056	2E	.	.	78	116	4E	N	N	110	156	6E	n	n
47	057	2F	/	/	79	117	4F	O	O	111	157	6F	o	o
48	060	30	0	0	80	120	50	P	P	112	160	70	p	p
49	061	31	1	1	81	121	51	Q	Q	113	161	71	q	q
50	062	32	2	2	82	122	52	R	R	114	162	72	r	r
51	063	33	3	3	83	123	53	S	S	115	163	73	s	s
52	064	34	4	4	84	124	54	T	T	116	164	74	t	t
53	065	35	5	5	85	125	55	U	U	117	165	75	u	u
54	066	36	6	6	86	126	56	V	V	118	166	76	v	v
55	067	37	7	7	87	127	57	W	W	119	167	77	w	w
56	070	38	8	8	88	130	58	X	X	120	170	78	x	x
57	071	39	9	9	89	131	59	Y	Y	121	171	79	y	y
58	072	3A	:	:	90	132	5A	Z	Z	122	172	7A	z	z
59	073	3B	;	;	91	133	5B	[	[	123	173	7B	{	{
60	074	3C	<	<	92	134	5C	\	\	124	174	7C	|	|
61	075	3D	=	=	93	135	5D	]	]	125	175	7D	}	}
62	076	3E	>	>	94	136	5E	^	^	126	176	7E	~	~
63	077	3F	?	?	95	137	5F	_	_	127	177	7F	DEL	⌂

Dec	Oct	Hex	DGI	PC	Dec	Oct	Hex	DGI	PC
128	200	80		Ç	192	300	C0	Á	└
129	201	81		ü	193	301	C1	À	⊥
130	202	82		é	194	302	C2	Â	┬
131	203	83		â	195	303	C3	Ä	├
132	204	84		ä	196	304	C4	Ã	─
133	205	85		à	197	305	C5	Å	┼
134	206	86		å	198	306	C6	Æ	╞
135	207	87		ç	199	307	C7	Ç	╟
136	210	88		ê	200	310	C8	É	╚
137	211	89		ë	201	311	C9	È	╔
138	212	8A		è	202	312	CA	Ê	╩
139	213	8B		ï	203	313	CB	Ë	╦
140	214	8C		î	204	314	CC	Í	╠
141	215	8D		ì	205	315	CD	Ì	═
142	216	8E		Ä	206	316	CE	Î	╬
143	217	8F		Å	207	317	CF	Ï	╧
144	220	90		É	208	320	D0	Ñ	╨
145	221	91		æ	209	321	D1	Ó	╤
146	222	92		Æ	210	322	D2	Ò	╥
147	223	93		ô	211	323	D3	Ô	╙
148	224	94		ö	212	324	D4	Ö	╘
149	225	95		ò	213	325	D5	Õ	╒
150	226	96		û	214	326	D6	Ø	╓
151	227	97		ù	215	327	D7	Œ	╫
152	230	98		ÿ	216	330	D8	Ú	╪
153	231	99		Ö	217	331	D9	Ù	┘
154	232	9A		Ü	218	332	DA	Û	┌
155	233	9B		¢	219	333	DB	Ü	█
156	234	9C		£	220	334	DC	space	▄
157	235	9D		¥	221	335	DD	Ÿ	▌
158	236	9E		₧	222	336	DE	space	▐
159	237	9F		ƒ	223	337	DF	space	▀
160	240	A0	space	á	224	340	E0	á	α
161	241	A1	→	í	225	341	E1	à	β
162	242	A2	½	ó	226	342	E2	â	Γ
163	243	A3	µ	ú	227	343	E3	ä	π
164	244	A4	²	ñ	228	344	E4	ã	Σ
165	245	A5	³	Ñ	229	345	E5	å	σ
166	246	A6	¤	ª	230	346	E6	æ	µ
167	247	A7	¢	º	231	347	E7	ç	τ
168	250	A8	£	¿	232	350	E8	é	Φ
169	251	A9	ª	⌐	233	351	E9	è	θ
170	252	AA	º	¬	234	352	EA	ê	Ω
171	253	AB	¡	½	235	353	EB	ë	δ
172	254	AC	¿	¼	236	354	EC	í	∞
173	255	AD	©	¡	237	355	ED	ì	φ
174	256	AE	®	«	238	356	EE	î	ε
175	257	AF	‡	»	239	357	EF	ï	∩
176	260	B0	»	░	240	360	F0	ñ	≡
177	261	B1	«	▒	241	361	F1	ó	±
178	262	B2	¶	▓	242	362	F2	ò	≥
179	263	B3	™	│	243	363	F3	ô	≤
180	264	B4	ƒ	┤	244	364	F4	ö	⌠
181	265	B5	¥	╡	245	365	F5	õ	⌡
182	266	B6	±	╢	246	366	F6	ø	÷
183	267	B7	≤	╖	247	367	F7	œ	≈
184	270	B8	≥	╕	248	370	F8	ú	°
185	271	B9	·	╣	249	371	F9	ù	·
186	272	BA	` (grave)	║	250	372	FA	û	·
187	273	BB	§	╗	251	373	FB	ü	√
188	274	BC	° (degree)	╝	252	374	FC	β	ⁿ
189	275	BD	¨ (umlaut)	╜	253	375	FD	ÿ	²
190	276	BE	´ (acute)	╛	254	376	FE	space	■
191	277	BF	↑	┐	255	377	FF	space	space

▸ **Notes:**
  1. Decimal codes 128 - 159 for DGI are the same as their 7-bit counterparts by default.
  2. DGI is as defined by a D216E+/D217/D413/D463 terminal.

APPENDIX B.   EBCDIC CODES

Dec	Oct	Hex	Char	Dec	Oct	Hex	Char	Dec	Oct	Hex	Char	Dec	Oct	Hex	Char
0	000	00	NUL	32	040	20	DS	64	100	40	space	96	140	60	-
1	001	01	SOH	33	041	21	SOS	65	101	41		97	141	61	/
2	002	02	STX	34	042	22	FS	66	102	42		98	142	62	
3	003	03	ETX	35	043	23		67	103	43		99	143	63	
4	004	04	PF	36	044	24	BYP	68	104	44		100	144	64	
5	005	05	HT	37	045	25	LF	69	105	45		101	145	65	
6	006	06	LC	38	046	26	ETB	70	106	46		102	146	66	
7	007	07	DEL	39	047	27	ESC	71	107	47		103	147	67	
8	010	08		40	050	28		72	110	48		104	150	68	
9	011	09		41	051	29		73	111	49		105	151	69	
10	012	0A	SMM	42	052	2A	SM	74	112	4A		106	152	6A	|
11	013	0B	VT	43	053	2B	CU2	75	113	4B	.	107	153	6B	,
12	014	0C	FF	44	054	2C	DC4	76	114	4C	<	108	154	6C	%
13	015	0D	CR	45	055	2D	ENQ	77	115	4D	(	109	155	6D	_
14	016	0E	SO	46	056	2E	ACK	78	116	4E	+	110	156	6E	>
15	017	0F	SI	47	057	2F	BEL	79	117	4F	|	111	157	6F	?
16	020	10	DLE	48	060	30		80	120	50	&	112	160	70	
17	021	11	DC1 (XON)	49	061	31		81	121	51		113	161	71	
18	022	12	DC2	50	062	32	SYN	82	122	52		114	162	72	
19	023	13	DC3(XOFF)	51	063	33		83	123	53		115	163	73	
20	024	14	RES	52	064	34	PN	84	124	54		116	164	74	
21	025	15	NL	53	065	35	RS	85	125	55		117	165	75	
22	026	16	BS	54	066	36	UC	86	126	56		118	166	76	
23	027	17	IL	55	067	37	EOT	87	127	57		119	167	77	
24	030	18	CAN	56	070	38		88	130	58		120	170	78	
25	031	19	EM	57	071	39		89	131	59		121	171	79	`
26	032	1A	CC	58	072	3A		90	132	5A	!	122	172	7A	:
27	033	1B	CU1	59	073	3B	CU3	91	133	5B	$	123	173	7B	#
28	034	1C	FS	60	074	3C		92	134	5C	*	124	174	7C	@
29	035	1D	GS	61	075	3D	NAK	93	135	5D	)	125	175	7D	'
30	036	1E	RS	62	076	3E		94	136	5E	;	126	176	7E	=
31	037	1F	US	63	077	3F	SUB	95	137	5F	~	127	177	7F	"

Dec	Oct	Hex	Char	Dec	Oct	Hex	Char	Dec	Oct	Hex	Char	Dec	Oct	Hex	Char
128	200	80		160	240	A0		192	300	C0	{	224	340	E0	\
129	201	81	a	161	241	A1	~	193	301	C1	A	225	341	E1	
130	202	82	b	162	242	A2	s	194	302	C2	B	226	342	E2	S
131	203	83	c	163	243	A3	t	195	303	C3	C	227	343	E3	T
132	204	84	d	164	244	A4	u	196	304	C4	D	228	344	E4	U
133	205	85	e	165	245	A5	v	197	305	C5	E	229	345	E5	V
134	206	86	f	166	246	A6	w	198	306	C6	F	230	346	E6	W
135	207	87	g	167	247	A7	x	199	307	C7	G	231	347	E7	X
136	210	88	h	168	250	A8	y	200	310	C8	H	232	350	E8	Y
137	211	89	i	169	251	A9	z	201	311	C9	I	233	351	E9	Z
138	212	8A		170	252	AA		202	312	CA		234	352	EA	
139	213	8B		171	253	AB		203	313	CB		235	353	EB	
140	214	8C		172	254	AC		204	314	CC		236	354	EC	
141	215	8D		173	255	AD	[	205	315	CD		237	355	ED	
142	216	8E		174	256	AE		206	316	CE		238	356	EE	
143	217	8F		175	257	AF		207	317	CF		239	357	EF	
144	220	90		176	260	B0		208	320	D0	}	240	360	F0	0
145	221	91	j	177	261	B1		209	321	D1	J	241	361	F1	1
146	222	92	k	178	262	B2		210	322	D2	K	242	362	F2	2
147	223	93	l	179	263	B3		211	323	D3	L	243	363	F3	3
148	224	94	m	180	264	B4		212	324	D4	M	244	364	F4	4
149	225	95	n	181	265	B5		213	325	D5	N	245	365	F5	5
150	226	96	o	182	266	B6		214	326	D6	O	246	366	F6	6
151	227	97	p	183	267	B7		215	327	D7	P	247	367	F7	7
152	230	98	q	184	270	B8		216	330	D8	Q	248	370	F8	8
153	231	99	r	185	271	B9		217	331	D9	R	249	371	F9	9
154	232	9A	^	186	272	BA		218	332	DA		250	372	FA	
155	233	9B		187	273	BB		219	333	DB		251	373	FB	
156	234	9C		188	274	BC		220	334	DC		252	374	FC	
157	235	9D		189	275	BD	]	221	335	DD		253	375	FD	
158	236	9E		190	276	BE		222	336	DE		254	376	FE	
159	237	9F		191	277	BF		223	337	DF		255	377	FF	

APPENDIX C.   ASCII to EBCDIC

Dec	Oct	Hex	EBCDIC Dec
0	000	00	–
1	001	01	–
2	002	02	–
3	003	03	–
4	004	04	55
5	005	05	45
6	006	06	46
7	007	07	47
8	010	08	22
9	011	09	5
10	012	0A	37
11	013	0B	–
12	014	0C	–
13	015	0D	–
14	016	0E	–
15	017	0F	–
16	020	10	–
17	021	11	–
18	022	12	–
19	023	13	–
20	024	14	60
21	025	15	61
22	026	16	50
23	027	17	38
24	030	18	–
25	031	19	–
26	032	1A	63
27	033	1B	39
28	034	1C	–
29	035	1D	–
30	036	1E	–
31	037	1F	–
32	040	20	64
33	041	21	90
34	042	22	127
35	043	23	123
36	044	24	91
37	045	25	108
38	046	26	80
39	047	27	125
40	050	28	77
41	051	29	93
42	052	2A	92
43	053	2B	78
44	054	2C	107
45	055	2D	96
46	056	2E	75
47	057	2F	97
48	060	30	240
49	061	31	241
50	062	32	242
51	063	33	243
52	064	34	244
53	065	35	245
54	066	36	246
55	067	37	247
56	070	38	248
57	071	39	249
58	072	3A	122
59	073	3B	94
60	074	3C	76
61	075	3D	126
62	076	3E	110
63	077	3F	111
64	100	40	124
65	101	41	193
66	102	42	194
67	103	43	195
68	104	44	196
69	105	45	197
70	106	46	198
71	107	47	199
72	110	48	200
73	111	49	201
74	112	4A	209
75	113	4B	210
76	114	4C	211
77	115	4D	212
78	116	4E	213
79	117	4F	214
80	120	50	215
81	121	51	216
82	122	52	217
83	123	53	226
84	124	54	227
85	125	55	228
86	126	56	229
87	127	57	230
88	130	58	231
89	131	59	232
90	132	5A	233
91	133	5B	173
92	134	5C	224
93	135	5D	189
94	136	5E	95
95	137	5F	109
96	140	60	121
97	141	61	129
98	142	62	130
99	143	63	131
100	144	64	132
101	145	65	133
102	146	66	134
103	147	67	135
104	150	68	136
105	151	69	137
106	152	6A	145
107	153	6B	146
108	154	6C	147
109	155	6D	148
110	156	6E	149
111	157	6F	150
112	160	70	151
113	161	71	152
114	162	72	153
115	163	73	162
116	164	74	163
117	165	75	164
118	166	76	165
119	167	77	166
120	170	78	167
121	171	79	168
122	172	7A	169
123	173	7B	192
124	174	7C	106
125	175	7D	208
126	176	7E	161
127	177	7F	7

Dec	Oct	Hex	EBCDIC Dec
128	200	80	32
129	201	81	33
130	202	82	34
131	203	83	35
132	204	84	36
133	205	85	21
134	206	86	6
135	207	87	23
136	210	88	40
137	211	89	41
138	212	8A	42
139	213	8B	43
140	214	8C	44
141	215	8D	9
142	216	8E	10
143	217	8F	27
144	220	90	48
145	221	91	49
146	222	92	26
147	223	93	51
148	224	94	52
149	225	95	53
150	226	96	54
151	227	97	8
152	230	98	56
153	231	99	59
154	232	9A	60
155	233	9B	61
156	234	9C	4
157	235	9D	20
158	236	9E	62
159	237	9F	225
160	240	A0	65
161	241	A1	66
162	242	A2	67
163	243	A3	68
164	244	A4	69
165	245	A5	70
166	246	A6	71
167	247	A7	72
168	250	A8	73
169	251	A9	81
170	252	AA	82
171	253	AB	83
172	254	AC	84
173	255	AD	85
174	256	AE	86
175	257	AF	87
176	260	B0	88
177	261	B1	89
178	262	B2	98
179	263	B3	99
180	264	B4	100
181	265	B5	101
182	266	B6	102
183	267	B7	103
184	270	B8	104
185	271	B9	105
186	272	BA	112
187	273	BB	113
188	274	BC	114
189	275	BD	115
190	276	BE	116
191	277	BF	117
192	300	C0	118
193	301	C1	119
194	302	C2	120
195	303	C3	128
196	304	C4	138
197	305	C5	139
198	306	C6	140
199	307	C7	141
200	310	C8	142
201	311	C9	143
202	312	CA	144
203	313	CB	154
204	314	CC	155
205	315	CD	156
206	316	CE	157
207	317	CF	158
208	320	D0	159
209	321	D1	160
210	322	D2	170
211	323	D3	171
212	324	D4	172
213	325	D5	74
214	326	D6	174
215	327	D7	175
216	330	D8	176
217	331	D9	177
218	332	DA	178
219	333	DB	179
220	334	DC	180
221	335	DD	181
222	336	DE	182
223	337	DF	183
224	340	E0	184
225	341	E1	185
226	342	E2	186
227	343	E3	187
228	344	E4	188
229	345	E5	79
230	346	E6	190
231	347	E7	191
232	350	E8	202
233	351	E9	203
234	352	EA	204
235	353	EB	205
236	354	EC	206
237	355	ED	207
238	356	EE	218
239	357	EF	219
240	360	F0	220
241	361	F1	221
242	362	F2	222
243	363	F3	223
244	364	F4	234
245	365	F5	235
246	366	F6	236
247	367	F7	237
248	370	F8	238
249	371	F9	239
250	372	FA	–
251	373	FB	–
252	374	FC	–
253	375	FD	–
254	376	FE	–
255	377	FF	–

▸ **Note:**   - indicates straight-thru (same on both)

APPENDIX D.   EBCDIC to ASCII

Dec	Oct	Hex	ASCII Dec	Dec	Oct	Hex	ASCII Dec	Dec	Oct	Hex	ASCII Dec	Dec	Oct	Hex	ASCII Dec
0	000	00	–	32	040	20	128	64	100	40	32	96	140	60	45
1	001	01	–	33	041	21	129	65	101	41	160	97	141	61	47
2	002	02	–	34	042	22	130	66	102	42	161	98	142	62	178
3	003	03	–	35	043	23	131	67	103	43	162	99	143	63	179
4	004	04	156	36	044	24	132	68	104	44	163	100	144	64	180
5	005	05	9	37	045	25	10	69	105	45	164	101	145	65	181
6	006	06	134	38	046	26	23	70	106	46	165	102	146	66	182
7	007	07	127	39	047	27	27	71	107	47	166	103	147	67	183
8	010	08	151	40	050	28	136	72	110	48	167	104	150	68	184
9	011	09	141	41	051	29	137	73	111	49	168	105	151	69	185
10	012	0A	142	42	052	2A	138	74	112	4A	213	106	152	6A	124
11	013	0B	–	43	053	2B	139	75	113	4B	46	107	153	6B	44
12	014	0C	–	44	054	2C	140	76	114	4C	60	108	154	6C	37
13	015	0D	–	45	055	2D	5	77	115	4D	40	109	155	6D	95
14	016	0E	–	46	056	2E	6	78	116	4E	43	110	156	6E	62
15	017	0F	–	47	057	2F	7	79	117	4F	229	111	157	6F	63
16	020	10	–	48	060	30	144	80	120	50	38	112	160	70	186
17	021	11	–	49	061	31	145	81	121	51	169	113	161	71	187
18	022	12	–	50	062	32	22	82	122	52	170	114	162	72	188
19	023	13	–	51	063	33	147	83	123	53	171	115	163	73	189
20	024	14	157	52	064	34	148	84	124	54	172	116	164	74	190
21	025	15	133	53	065	35	149	85	125	55	173	117	165	75	191
22	026	16	8	54	066	36	150	86	126	56	174	118	166	76	192
23	027	17	135	55	067	37	4	87	127	57	175	119	167	77	193
24	030	18	–	56	070	38	152	88	130	58	176	120	170	78	194
25	031	19	–	57	071	39	153	89	131	59	177	121	171	79	96
26	032	1A	146	58	072	3A	154	90	132	5A	33	122	172	7A	58
27	033	1B	143	59	073	3B	155	91	133	5B	36	123	173	7B	35
28	034	1C	–	60	074	3C	20	92	134	5C	42	124	174	7C	64
29	035	1D	–	61	075	3D	21	93	135	5D	41	125	175	7D	39
30	036	1E	–	62	076	3E	158	94	136	5E	59	126	176	7E	61
31	037	1F	–	63	077	3F	26	95	137	5F	94	127	177	7F	34

Dec	Oct	Hex	ASCII Dec	Dec	Oct	Hex	ASCII Dec	Dec	Oct	Hex	ASCII Dec	Dec	Oct	Hex	ASCII Dec
128	200	80	195	160	240	A0	209	192	300	C0	123	224	340	E0	92
129	201	81	97	161	241	A1	126	193	301	C1	65	225	341	E1	159
130	202	82	98	162	242	A2	115	194	302	C2	66	226	342	E2	83
131	203	83	99	163	243	A3	116	195	303	C3	67	227	343	E3	84
132	204	84	100	164	244	A4	117	196	304	C4	68	228	344	E4	85
133	205	85	101	165	245	A5	118	197	305	C5	69	229	345	E5	86
134	206	86	102	166	246	A6	119	198	306	C6	70	230	346	E6	87
135	207	87	103	167	247	A7	120	199	307	C7	71	231	347	E7	88
136	210	88	104	168	250	A8	121	200	310	C8	72	232	350	E8	89
137	211	89	105	169	251	A9	122	201	311	C9	73	233	351	E9	90
138	212	8A	196	170	252	AA	210	202	312	CA	232	234	352	EA	244
139	213	8B	197	171	253	AB	211	203	313	CB	233	235	353	EB	245
140	214	8C	198	172	254	AC	212	204	314	CC	234	236	354	EC	246
141	215	8D	199	173	255	AD	91	205	315	CD	235	237	355	ED	247
142	216	8E	200	174	256	AE	214	206	316	CE	236	238	356	EE	248
143	217	8F	201	175	257	AF	215	207	317	CF	237	239	357	EF	249
144	220	90	202	176	260	B0	216	208	320	D0	125	240	360	F0	48
145	221	91	106	177	261	B1	217	209	321	D1	74	241	361	F1	49
146	222	92	107	178	262	B2	218	210	322	D2	75	242	362	F2	50
147	223	93	108	179	263	B3	219	211	323	D3	76	243	363	F3	51
148	224	94	109	180	264	B4	220	212	324	D4	77	244	364	F4	52
149	225	95	110	181	265	B5	221	213	325	D5	78	245	365	F5	53
150	226	96	111	182	266	B6	222	214	326	D6	79	246	366	F6	54
151	227	97	112	183	267	B7	223	215	327	D7	80	247	367	F7	55
152	230	98	113	184	270	B8	224	216	330	D8	81	248	370	F8	56
153	231	99	114	185	271	B9	225	217	331	D9	82	249	371	F9	57
154	232	9A	203	186	272	BA	226	218	332	DA	238	250	372	FA	–
155	233	9B	204	187	273	BB	227	219	333	DB	239	251	373	FB	–
156	234	9C	205	188	274	BC	228	220	334	DC	240	252	374	FC	–
157	235	9D	206	189	275	BD	93	221	335	DD	241	253	375	FD	–
158	236	9E	207	190	276	BE	230	222	336	DE	242	254	376	FE	–
159	237	9F	208	191	277	BF	231	223	337	DF	243	255	377	FF	–

▸ **Note:** - indicates straight-thru (same on both)

APPENDIX E.   EXCEPTION STATUS CODES

Following is a list of Exception Status codes along with the File Status that will be set, if appropriate. If two File Status values are given, the first is with ANSI COBOL 74 and the second is with ANSI COBOL 85; if only one is given, both return the same value.

On Windows, errors 1 - 31 map directly to Exception Status 1 - 31, while errors 32 - 92 map to Exception Status 288 - 347, i.e., add 256 to Microsoft errors greater than 31.

On UNIX, errno maps to an Exception Status as documented in the second part of this table.

Exception Status	74 File Status	85 File Status	Message
1	30		Invalid operation
2	91	35	File not found
3	96		Path not found
4	91		No more handles available
5	92	37	Access denied
6	92		Invalid handle
7	30		Memory control blocks bad
8	30		Insufficient memory
9	30		Invalid memory control block address
10	30		Invalid environment
11	30		Invalid format
12	30		Invalid access code
13	30		Invalid data
14	30		Insufficient memory to complete this operation
15	96		Invalid drive specifier
16	92		Attempt to remove current directory
17	91		Not the same device
18	91		No more files
19	30	37	Write protected disk
20	30		Unknown hardware unit
21	30		Drive is not ready
22	30		Unknown hardware command
23	30		CRC error in data
24	30		Hardware drive request is bad
25	30		Disk seek error
26	30		Unknown disk media type
27	30		Sector not found
28	30		Printer out of paper
29	30		Write fault
30	30		Read fault
31	30		General failure
32	94		The file already exists
33	94		The file is exclusively opened
34	34		The filesize is too big
35	94	41	Attempt to exclusively open an open file
36	91		The filename is not valid
37	10		End of file
38	98	24	Invalid relative key
39	34		Out of (disk) space
40	91	34	Readline argument is too long
41	91	41	Attempt to open an open file
42	91	42	Attempt to close a closed file
43	92	38	Attempt to open a locked file
44	99		Printer control file is full
45	92		Invalid operation for open mode
46	92		Handle is not open
47	94		Attempt to delete an open file
48	92	34	Record area size too small for record
49	92	44	Record size mismatch on rewrite
50	9A	39	Record too long
51	9A	39	Too many keys requested
52	9A	39	Invalid key packet length
53	9A	39	Key is too long

Exception Status	74 File Status	85 File Status	Message
54	9A	39	Invalid key definition (not in record)
55	9A	39	Record size mismatch on open
56	9A	39	Number of keys mismatch on open
57	9A	39	Key size mismatch on open
58	9A	39	Key offset mismatch on open
59	9A		.NX file version is not valid
60	9A		.XD file version is not valid
61	9E		Out of record locks
62	94		Record is locked
63	23	46	Invalid current record pointer
64	23		Record is deleted
65	22		Record is not deleted
66	21		Not rewriting the current record
67	23		Key not found
68	22		Attempt to write a duplicate key
69	24		.NX file B-tree is full (node depth or full node)
70	21		Not writing in ascending order
71	9B		The .NX file is corrupt
72	9B		The .XD file is corrupt
73	9F		Reliability flag indicates .NX file is corrupt
74	9F		Reliability flag indicates .XD file is corrupt
75	94		Attempt to rename an open file
76	9T		Device timeout
77	30		Device I/O error
78	30		Printer is offline
79	30		Printer is out of paper
80	30		I/O operation aborted by console interrupt
81	91		Device is not available or does not exist
82	9B		The file format is not valid
83	9B		The file does not have the correct revision
84	9B		Record size is zero
85	9B		Record position is too small
86	9B		Record position is not aligned
87	9B		Record position is too big
88	9B		Record position is past EOF
89	9B		Node block number is not zero
90	9B		Node block number is zero
91	9B		Node block number is too big
92	9B		Duplicates are permitted
93	9B		Duplicates are not permitted
94	9B		Key size is zero
95	9B		Node block number is past EOF
96	9B		.XD file size is too small
97	9B		.NX file size is too small
98	9B		Key entry is deleted
99	9B		Record position does not match
100	9B		File version does not match
101	9B		Node block number is inconsistent
102	9B		Node entry count is zero
103	9B		Node entry count is too big
104	9B		Node entry count is the maximum
105	9B		Node level is inconsistent
106	9B		Node key number is inconsistent
107	9B		Node leaf indicator is inconsistent
108	9B		Position is not aligned on a node boundary
109	9B		Relative key value is inconsistent
110	9B		key value is inconsistent
111	00		Reliability flag(s) have been cleared
112	9B		Internal error - invalid use of buffer manager
113	9B		Attempt to release buffer not in use
114	9B		No buffers were available
115	9B		Attempt to destroy buffer still in use
116	9B		The object definition is in use (internal error)
117	97		No more files may be OPENed
118	97		No more OPEN resources are available
119	97		No more SEQUENTIAL files may be OPENed
120	97		No more RELATIVE files may be OPENed

Exception Status	74 File Status	85 File Status	Message
121	97		No more INDEXED files may be OPENed
122	30		Data Carrier Detect (DCD) has been lost
123	30		Requested object definition is not registered (internal error)
124	30		The path does not specify a directory
125	30		I/O aborted by WATCH interrupt
126	30		The terminal has too few lines to WATCH the selected terminal
127	30		The object does not match the expected object type (internal error)
128	30		Console interrupts are disabled
129	30		Aborted by DUMP interrupt
130	97		Object handle or indexd entry is NULL (internal error)
131	9B		No data is available
132	9A		Named item is the wrong type to perform this operation
133	91		The parameter string is not valid for this object
134	91		File's standard header information is bad (.ini)
135	97		Not enough resources to complete request
136	30		Internal system error
137	30		Invalid argument to system call
138	92		File or device must be on the same node or volume
139	02		A duplicate key value has been written
140	02		A duplicate key value has been read
141	30	9B	File standard header is not valid
142	30	9B	File standard header checksum is bad
143	30	9B	File type does not match required type
144	30	9B	File header length, offset, or checksum is bad
145	30	9B	File has wrong byte order
146	9A	39	Key with duplicates specification does not match
147	9A	39	ICISAM file format does not match
148	9A	39	ICISAM file version does not match
149	92	39	The .NX and .XD files are not properly paired
150	9A	39	Delete-is-physical mismatch on open
151	9A	39	Key null value suppression mismatch on open
152	9A	39	Key uppercase conversion specification does not match
153	00	05	File was created
154	00	05	The optional file was not available
155	92	47	Invalid operation for file without input access
156	92	48	Invalid operation for file without output access
157	92	49	Invalid operation for file without I-O access
158	92	43	DELETE or REWRITE not preceded by a successful READ
159	9B		The header information from the .XD and .NX file is not consistent
160	30		A Sort or Merge operation is already active
161	92	10	Optional file was unavailable for sequential READ
162	92	23	Optional file was unavailable for random READ or START
163	30	14	The relative key value exceeds the size of the relative key on READ
164	30	24	The relative key value exceeds the size of the relative key on WRITE
165	9B		Position is not aligned on a shared page boundary
166	22		Attempt to modify an unmodifiable key
167	94		Attempt to rewrite a record which has been modified since it was read
168	94		Attempt to perform an operation which would lead to a deadlock situation
169	9B		Invalid record length value in record header
170	9A	39	Too many key occurs requested
171	9A	39	Too many key suffixes requested
172	9A	39	Too many key alsos requested
173	9A	39	Key occurs/also specification does not match
174	9A	39	Key occurs/also count does not match
175	9A	39	Key occurs span specification does not match
176	9A	39	Key suffix count specification does not match
177	9A	39	Key reverse order specification does not match
178	30		The .XL and .XD files are not properly paired
179	30		The .XL operation is not in sequence
180	30		Invalid combination of network options
181	30		An invalid or corrupted network packet was received
182	30		Data value is not a valid data-type-vale
183	30		Data does not fit in the data area provided
184	9A	39	4GB maximum file size specification does not match
185	92	44	Record size specified exceeds the maximum or is less than the minimum
186	30		*ERROR:
187	30		Conversion error (index register overflow)

Exception Status	74 File Status	85 File Status	Message
188	30		An index is out of range
189	30		The perform count is too large
190	30		The perform stack has overflowed
191	30		Fatal I/O error
192	04		Length of record does not conform to that specified for the file
193	30		The program was terminated by a console interrupt
194	30		**stop run**
195	30		Fatal Runtime System Error
196	30		Fatal Runtime System Error: invalid operation code
197	30		The system is ready. Press Newline to begin LOGON
198	30		The system is currently unavailable
199	30		The program was terminated by another console
200	30		The program is too big
202	30		The program file is not valid
203	30		The program was not found
207	30		The program is already active
208	30		Attempt to call too many programs
209	30		Parameter mismatch in call
212			No more programs are available
213			The program file could not be loaded
215			The program had been disabled
216			I/O aborted by Wakeup Interrupt
219			Invalid task number
220			There are no more entries in the table
221			This operation is not permitted
222			The item is currently in use
223			The item was removed
224			The requested page is not in the file
228			The terminal is not logged on
229			The terminal is not configured into the system
230			The configuration file is not valid
231			Unsupported feature for the current terminal
233			The user has not been granted the requested logon type at this computer
234			The abort request was sent to terminal
235			The message was sent to terminal
236			The maximum number of users are already running
237			The option is not a valid option
239			Process initialization error
240			The option requires an argument
241			The argument is too long to process
242			There are no more options to process
243			Out of processes, system resources, or no data available
244			Shared memory initialization error
245			Shared area revision does not match
246			The shared area is not ready for use
247			Semaphore initialization error
248			No more processes can be run
249			Username:
250			ICEXEC is required
251			Process termination (Quit/Logoff)
252			Program not authorized
253			Process termination (Modem Hangup)
254			The process was terminated by a global timeout
255			Process Termination (Shutdown)
256			Super user privileges required
257			Detaching from login session
258			Detached from login session
259			Icexec was abnormally terminated
260			The (.ini) file section was not found

Exception Status	74 File Status	85 File Status	Message
261			Insufficient memory for Device Control Tables
262			Unable to initialize standard input file
263			Unable to initialize standard output file
264			Unable to initialize standard error file
265			Locking Open/Close
266			Unkown terminal type from terminfo
267			Terminal description keyboard table
268			Unsupported terminal types
269			Screen Control Area
270			Too many directories in path
271			Insufficient memory for pathlist
272			Too many libraries
273			The environment argument is not valid
274			The following environment argument was not provided
275			
276			
279			
280			
281			
282			
284			
285			
286			
287			
288	92		Sharing violation
289	94		Lock violation
290	30		Invalid disk change
291	30		FCB unavailable
292	30		Sharing buffer overflow
294	30		Out of Input
295	34		Insufficient disk space
306	30		Network request not supported
307	30		Remote computer not listed
308	30		Duplicate name on network
309	30		Network name not found
310	30		Network busy
311	30		Network device no longer exists
312	30		Net BIOS command limit exceeded
313	30		Network adapter hardware error
314	30		Incorrect response from network
315	30		Unexpected network error
316	30		Incompatible remote adapter
317	30		Print queue full
318	30		Not enough space for print file
319	30		Print file was deleted
320	30		Network name deleted
321	92	37	Access denied
322	30		Network device type incorrect
323	30		Network name not found
324	30		Network name limit exceeded
325	30		Net BIOS session limit exceeded
326	30		Temporarily paused
327	30		Network request not accepted
328	30		Print or disk redirection is paused
334	30		Not logged in or Network name not valid
336	94		File exists
337	30		
338	30		Cannot make directory entry
339	30		Fail on INT 24
340	30		Too many redirections
341	30		Duplicate redirection

Exception Status	74 File Status	85 File Status	Message
342	30		Invalid password
343	30		Invalid parameter
344	30		Network data fault
345	30		The system cannot start another process at this time
346	30		Required system component not installed
365	30		Connection broken
378	30		The data area passed to the system call is too small
416			Record manager initialization failed
417			Record Manager is inactive
418			Record Manager interface is invalid
419			Record Manager does not implement the required capability
420			Record Manager returned a reserved status code
421			Record Manager returned a generic status code
422			Record Manager returned an undefined status code
432			The terminal is already being WATCHed
433			Cannot watch a pushed terminal
434			Cannot watch a watching terminal
435			A watched terminal cannot watch another
436			Cannot interrupt the terminal to watch
437			Watched terminal has logged off
438			Watched terminal has pushed to CLI. Press Interrupt to discontinue watching
439			Invalid operation for your own terminal
440			Watched terminal terminated itself with an error
441			Watched terminal terminated by interrupt
442			The process is defunct
443			The watched terminals program process has terminated
444			Cannot watch an SP2 or CGI server process
445			Client/server protocol packet or parameter mismatch
448			Operation would block
449			Operation now in progress
450			Operation already in progress
451			Socket operation on non-socket
452			Destination address required
453			Message too long
454			Protocol wrong type for socket
455			Protocol not available
456			Protocol not supported
457			Socket type not supported
458			Operation not supported on socket
459			Protocol family not supported
460			Address family not supported
461			Address already in use
462			Cannot assign requested address
463			Network is down
464			Network is unreachable
465			Network dropped connection on reset
466			Software caused connection abort
467			Connection reset by peer
468			Out of stream resources
469			Socket is already connected
470			Socket not connected
471			Cannot send after socket shutdown
472			Too many connection, cannot splice
473			Connection timed out
474			Connection refused
475			Too many symbolic links in path
476			Filename too long
477			Host is down
478			No route to host
479			Host not found
481			No more stream resources are available

Exception Status	74 File Status	85 File Status	Message
482			The user account already exists
483			The password is too short or fails some other restriction
484			This beta release expired
485			This beta release will run until
486			Open/close semaphore could not be created
487			Open/close setup failed
488			Buffer semaphore could not be created
489			Buffer setup failed
490			Record lock semaphore could not be created
491			Record lock setup failed
492			Logon/logoff semaphore could not be created
493			Logon/logoff setup failed
494			Open semaphore could not be deleted
495			Buffer semaphore could not be deleted
496			Record lock semaphore could not be deleted
497			Logon/logoff semaphore could not be deleted
500			ExitCode 0: Processing completed successfully
501			ExitCode 1: Processing occurred, but had errors
502			ExitCode 2: Processing occurred, but was interrupted or aborted
503			ExitCode 3: Processing occurred, but was halted by a fatal internal error
504			ExitCode 4: Processing failed because of command-line errors
505			ExitCode 5: Processing failed becuaes of an authorization failure
506			ExitCode 6: Processing failed because of program initialization
507			ExitCode 7: Processing did not occur because command-line help was requested
508			ExitCode 8: Processing did not occur because a command-line status request ran
509			ExitCode 9: reserver
510			Unimplemented operating system function
511			Unexpected operating system error

Following is a mapping of UNIX errors (errno) to Exception Status codes.

E2BIG	241	EINVAL	137	ENOTCONN	470
EACCES	5	EIO	77	ENOTDIR	3
EADDRINUSE	461	EISCONN	469	ENOTSOCK	451
EADDRNOTAVAIL	462	EISDIR	5	ENOTTY	5
EADV	315	ELBIN	315	ENOTUNIQ	308
EAFNOSUPPORT	460	ELOOP	475	ENXIO	81
EAGAIN	243	EMFILE	4	EOPNOTSUPP	457
EALREADY	315	EMLINK	340	EOVERFLOW	136
EBADF	6	EMSGSIZE	453	EPERM	5
EBADFD	6	EMULTIHOP	340	EPFNOSUPPORT	459
EBADMSG	344	ENAMETOOLON	476	EPIPE	122
EBUSY	33	ENETDOWN	463	EPROTO	314
ECHILD	1	ENETRESET	465	EPROTONOSUPPORT	456
ECHRNG	6	ENETUNREACH	464	EPROTOTYPE	454
ECOMM	313	ENFILE	18	ERANGE	136
ECONNABORTED	466	ENOBUFS	468	EREMCHG	320
ECONNREFUSED	474	ENODATA	131	EREMOTE	138
ECONNRESET	467	ENODEV	81	EROFS	29
EDEADLK	168	ENOENT	2	ESHUTDOWN	471
EDEADLOCK	168	ENOEXEC	137	ESOCKTNOSUPPORT	457
EDESTADDRREQ	452	ENOLCK	61	ESPIPE	25
EDOM	137	ENOLINK	311	ESRCH	219
EDOTDOT	315	ENOMEM	8	ESRMNT	315
EDQUOT	295	ENOMSG	131	ETIME	76
EEXIST	32	ENONET	307	ETIMEDOUT	473
EFAULT	9	ENOPKG	346	ETOOMANYREFS	472
EFBIG	34	ENOPROTOOPT	455	ETXTBSY	5
EHOSTDOWN	477	ENOSPC	39	EWOULDBLOCK	448
EHOSTUNREACH	478	ENOSR	4	EXDEV	17
EIDRM	223	ENOSTR	5		
EINPROGRESS	315	ENOSYS	511		
EINTR	80	ENOTBLK	11		

any errno not listed generates a 511

INDEX

Support Information Request (SIR)

Envyr Support FAX: (919) 851-4609   Date: __/__/__

Envyr Support E-mail: support@icobol.com

<u>FROM</u>   Originator ID No:_____

Contact: _____   Phone: _____

Company: _____   Fax: _____

Address: _____   Timezone: _____

         _____

City: _____   State/Province: _____

Country: _____   Zip/Mail Code: _____

Interactive COBOL revision: _____ Purchased From: _____

Kind of problem:_____ (Enhancement/Suggestion, Question, Documentation

                    error, Software error)

Frequency: _____ (Frequent, Occasional, Erratic, Reproducible)

Significance: _____ (Low, Medium, High, Urgent)

<u>Host Machine Configuration</u>

Vendor: _____ Model: _____ CPU type: ____

Amount of Memory: _____

Peripherals:

OS Name and Version: _____ _____

Other software in Use with versions:

Rebooted from scratch?   Y   N

  (turned the power off and back on)

Problem/Suggestion: (Describe as fully as possible. If a COBOL problem, a
sample of code that generates the error would be appreciated.)

Attachments: _____ (None, tape, floppy, listings, etc.)

(Please label attachments with company, contents, format, and "# of #" (e.g.,
1 of 2).

From: _____   Stamp

      _____

      _____

Support
Envyr Corporation
#285
4915 Waters Edge Drive
Raleigh,   N.C.   27606
U.S.A.

(Fold Here)

CUSTOMER DOCUMENTATION COMMENT FORM
Envyr Support FAX: (919) 851-4609   Date: __/__/__

Name: _____

Company: _____

Street: _____

City: _____

State: ____   Zip: _____

Manual: Interactive COBOL Utilities Manual
   No: 011-00300-20

Title: _____

About the manual:

Is it easy to read?   Yes   No
Is it easy to understand?   Yes   No
Are the topics logically arranged?   Yes   No
Is the information correct?   Yes   No
Can you easily find what you want?   Yes   No
Does it tell you what you need to know? Yes   No

Comments:

From: _____

_____

_____

Stamp

Documentation
Envyr Corporation
#285
4915 Waters Edge Drive
Raleigh,   N.C.   27606
U.S.A.

(Fold Here)