Installing and Configuring

Interactive COBOL

on Windows®

Revision 3.64

No. 011-00403-08

01/21/2008

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.

Envyr Corporation
4904 Waters Edge Drive, Suite 160
Raleigh, N.C. 27606

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	2.30	- April 1997
Release	2.41	- 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	3.42	- May 2004
Release	3.64	- January 2008

Effective with:
Interactive COBOL Revision 3.64

TRADEMARKS

TABLE OF CONTENTS

APPENDICES

LIST OF FIGURES

LIST OF SCREENS

PREFACE

Scope

This manual provides the information needed to install and configure the Interactive COBOL product on 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)
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)
Provides the complete COBOL syntax supported by Interactive COBOL. Shows 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 Interactive COBOL sp2 User Interface (GUI) Development System.

COBOL FormPrint
Provides 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.

ICOBOL refers to any of the Interactive COBOL products unless otherwise stated.

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

Windows will be used to collectively refer to Windows 2000, Windows XP, Windows Server 2003, Windows Vista versions.

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.

RDOS refers to either RDOS or DG/RDOS.

DG refers to Data General Corporation.

ENHANCEMENTS (Windows)

Interactive COBOL 3.60 added support for the following:

- When using the Windows ThinClient (ICRUNRC) additional Windows builtins are allowed when attached to a UNIX surrogate (ICRUNRS). These are IC_WINDOWS_SHOW_CONSOLE, IC_WINDOW_TITLE, and IC_WINDOWS_MSG_BOX.

Interactive COBOL 3.57 added support for the following:

- ICRUNW and ICRUNRC support Copy/Paste.

- Console configuration has been enhanced with support for *ip-address* and/or *machine-name* device names that allow only the specified thin client connection.

Interactive COBOL 3.56 added support for the following:

- Filenames can contain "(" and ")".

- ICRUNW and ICRUNRC have an enhanced System menu with the addition of Font, Compress, Select printer, Printer font, and About.

Interactive COBOL 3.48 added support for the following:

- ICRUNW and ICRUNRC support the blink attribute by blinking.

Interactive COBOL 3.47 added support for the following:

- @PTS support enhanced via the use of PTS environment variables

Interactive COBOL 3.42 added support for the following:

- ICRUNW supports fontnames used with ICFONT and with the IC_WINDOWS_SETFONT builtin to now have a " Bold" attached to the fontname. I.E. "Courier New Bold" or "Lucida Console Bold". The IC_WINDOWS_SETFONT builtin will also return a " bold" concated to the fontname when bold has been selected. In addition, if non-fixed fonts are used the runtime will try to keep the characters aligned within limits.

Interactive COBOL 3.40 added support for the following:

- Common code has been placed in a single .dll (icsys.dll) that will be installed in the system32 directory. This should make most executables smaller.

- Console configuration has been enhanced with support for special "icrunrs" and "icthins" device names that allow only the specified thin client connection.

Interactive COBOL 3.36 added support for the following:

- ICRUNRC executable supports the Printer-Pass-Thru and Print Screen terminal codes

Interactive COBOL 3.34 added support for the following:

- ICRUNW and ICRUNRC executables select the appropriate Bold and Underline fonts to match the selected font and allowing the Bold and Underline screen attributes to be fully implemented

Interactive COBOL 3.30 added support for the following:

- ThinClient (character) ICRUNRC/ICRUNRS support added

- Configuration limits raised
  * The maximum number of @CONs was raised to 2048.
  * The maximum number of @PCQs, @PRNs, and @SERs was raised to 2048.
  * The maximum buffers was increased to 1024MB

- Configuration files, Terminal description files, and Printer translation files have been re-worked into .ini type files with the new extensions of .cfi, .tdi, and .pti respectively

Interactive COBOL 3.13 added support for the following:

- ThinClient (gui) enhancements

- Ability to change the runtime font under ICRUNW

Interactive COBOL 3.11 added support for the following:

- ICRUNW can be started with a different font/size

Interactive COBOL 3.10 added support for the following:

- ICPERMIT supports licensing via the MAC address of an ethernet card (NIC)

- ThinClient (gui) (ICTHINC/ICTHIS) support for sp2 and FormPrint runtimes

- Integrated Development Environment (ICIDE)

- GUI runtime (ICRUNW)

Interactive COBOL 3.00 added support for the following:

- New Runtime engine with enhanced program size and syntax support.

I. INTRODUCTION

A. Overview

The Interactive COBOL product set is composed of several products: The Interactive COBOL Runtime System, the Interactive COBOL Development System, the Interactive COBOL GUI Development System, the FormPrint Editor, the Interactive COBOL ODBC Driver, the Interactive COBOL Client/Server facility, and the cgiCOBOL runtime.

The Interactive COBOL Runtime is the product which runs standard Interactive COBOL programs along with the needed utilities. The Runtime System is all that is needed to distribute Interactive COBOL applications that do not need development capabilities.

The Interactive COBOL Development System adds the Interactive COBOL compiler(ICOBOL), code dumper (ICDUMP), User Library, Link Kit, along with an Integrated Development Environment (ICIDE) to the Runtime System. The Development System allows COBOL applications to be built and debugged while providing the ability to add user-written routines into a user-built runtime and to write non-COBOL programs that interface to Interactive COBOL Indexed files.

The Interactive COBOL GUI Development Systems provides the sp2 Panel Editor (ICSP2) and other needed files to allow graphical screens to be developed for deployment on Windows. The runtime support for “panels” created with the sp2 Panel Editor is provided in the Windows runtime releases and in UNIX runtime releases when communicating with Windows ThinClients..

The Interactive COBOL FormPrint Editor (ICQPRW) allows printer forms to be designed for use on Windows. It allows the complexities of such forms to be somewhat isolated from a COBOL program, yet still enables a COBOL programmer to have close control over the form directly from a program using standard COBOL code. The runtime support for “panels” created with ICQPRW is provided in the Windows runtime releases and in UNIX runtime releases when communicating with Windows ThinClients..

The Interactive COBOL Client/Server facility (ICNETD) is a server that provides the server side support for both the thick-client and thin-client client/server offerings across a network between UNIX and Windows machines. It provides thick-client support with the ICIOS surrogate and ThinClient support with the ICTHINS (gui-only) and ICRUNRS surrogates.

The cgiCOBOL runtime (ICRUNCGI) enables the use of **ICOBOL** programs as Common Gateway Interface (CGI) scripts, initiated by a web server in response to a browser request. CgiCOBOL programs can be written using standard ANSI COBOL syntax.

This manual will discuss:

1) how to install the Interactive COBOL product,
2) how to configure the execution environment, and
3) how to start COBOL programs.

Interactive COBOL on Windows is a 32-bit native application that requires either Windows 2000, Windows XP, Windows Server 2003, or Windows Vista. Most Interactive COBOL on Windows executables are provided as console applications.

Interactive COBOL licenses are available in models ranging from one to 2049 users.

An Interactive COBOL on Windows license usually includes a parallel or USB protection device that provides a unique serial number to which a license is keyed. (CPUID and MAC address licenses are also available.) The actual license gives the maximum number of active users to be logged on simultaneously to the runtime along with any options or additional products licensed for use by an installation. Licenses can be shared over a network with other Windows machines and when using the CROSS option on a license, licenses can be shared across platforms (Windows to UNIX, UNIX to Windows, UNIX flavor-a to UNIX flavor-b, etc.).

The Interactive COBOL on Windows runtime supports Microsoft and compatible networks allowing file sharing among various Interactive COBOL on MS-DOS and Interactive COBOL on Windows users on multiple nodes (or workstations) on the network using standard Microsoft type locking.

B. How to Read this Manual

Begin with the Installation Chapter (Chapter II), which describes how to install Interactive COBOL on Windows, any changes needed to the operating environment, and tips and techniques to get the most from your system.

Scan the readme file(s) for any new information that is not in the manuals.

Chapter III describes how to use the **ICOBOL** service manager (ICSVCMGR) on Windows.

Proceed with the Licensing (ICPERMIT) Chapter (Chapter IV), which describes how Interactive COBOL is licensed and how to use the Interactive COBOL License Server (ICPERMIT).

Continue the installation process by reading either the ICEDCFW Chapter (Chapter V) or ICCONFIG Chapter (Chapter VI), which describes how to operate the Interactive COBOL configuration utility to prepare a configuration file, printer translation file, and/or Terminal description file for the Interactive COBOL runtime.

Proceed with the ICEXEC Chapter (Chapter VII) which describes how to start and stop the Interactive COBOL Executive process (ICEXEC) which reads the configuration file.

Proceed with the Starting ICRUN Chapter (Chapter VIII), which describes how to start and stop the Interactive COBOL runtime.

Chapters IX and X describes ThinClient and ThinClient (gui) support respectively.

Chapter XI describes some tuning tips for Interactive COBOL.

Chapter XII describes the default ICTERM definitions.

The APPENDIX section contains various information including an ASCII Chart, an RS-232C discussion, Support suggestions, etc., which can be used as reference material.

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 systems. 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. They 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 system 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 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 and operating system. 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 used to set default operating options, or the locations of important files. For example, all Interactive COBOL Windows executable programs look for the environment variable ICROOT as the base directory for finding the system and help files. They also look for command-line options in an environment variable by their own name. Windows provides environment variables through its command processor.

During the Interactive COBOL installation, two pseudo-environment variables are configured into the Windows registry for use by Interactive COBOL programs if they do not find the environment variables maintained by the command processor. These two entries are the ICCODEPATH and ICROOT entries and they are set to point to the installation directory of Interactive COBOL.

Environment variables are maintained in the command processor. Environment variables are setup with a command like:

```
   SET ICROOT=C:\program files\icobol
      or
   SET ICRUN=-iC c:\master\icobol.cfi
```

On Windows, common environment variables can be configured into the Registry. From the **Start** menu, choose **Settings**, **Control Panel**. From **Control Panel**, select **System**, and then pick the **Environment** tab. In this sheet, system or user specific environment variables can be configured.

C.2. Directory Structure

The Interactive COBOL software is installed by default in the "program files" directory with the name "icobol". This directory can be installed wherever is most appropriate or convenient for your system and should be included on your PATH.

The main directory contains all of the Windows executable programs, the *readme* file(s), and supplied COBOL executable programs. One subdirectory is called *help*. The help subdirectory contains help (.hf) files for all the Windows executable programs defined as <command>.hf. Interactive programs have their own subdirectories under help. There may be additional directories with other miscellaneous files, see the appropriate *readme* and/or *packing-list* files for a list of all the actual files. Descriptions that start with (Dev) are part of the development system.

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

- main executables, .dlls, and needed files

Windows executable 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. The help file uses the ICROOT environment variable to find the needed file.

C.3. ICEXEC Control Program

The Interactive COBOL system uses a control program called ICEXEC to coordinate multi-user access to system resources. The following executables **require** the shared area that ICEXEC manages:

icrun, icrunw	icios	(Surrogate started by icnetd)
icruncgi	icrunrs	(Surrogate started by icnetd)
icsmview	icthins	(Surrogate 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 Windows executable 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 specified.

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

... Ellipse 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.

E. Common Switches

E.1. Overall

There are several common switches that appear on all Windows executable 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 the program 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 auditing as:

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 flags.
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, and Username options. On UNIX, 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) On Windows, 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 than the default for the audit file. The default action is the same as usual, truncate the file to zero on startup.

NOTE:

-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)
.CL	Library file
.CO	COBOL Source programs (card format) *(defacto)*
.COB	AOS/VS COBOL text-format source
.CX	COBOL Program file
.ER	Error file *(defacto)*
.FA	File attribute file
.FP	Failsafe protection file
.HF	Interactive COBOL help files
.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)
.RP	Remote protection file (MS-DOS only)
.SD	ICRUN Sort data file (temporary)
.SR	COBOL Source programs (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)
.TMP	Temporary file *(defacto)*
.UDB	U/FOS database
.XD,.NX	Pair of files, ICISAM file (data and index portion)
.XDB	ODBC database definition file (.ini)
.XDT	ODBC table definition file (.ini)
.XL	Log file
.XLG	Generation log file

G. Exit Codes

All Widows executable programs return exit codes that provide an indication of the success or failure of the program. These are returned into ERRORLEVEL on Windows. 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. For example, ICCHECK checked a series of files, and some of them were corrupt.

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 functions.

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 Windows executable 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.

On Windows, common environment variables can be configured into the registry. From the **Start** menu, choose **Settings**, **Control Panel**. From **Control Panel** select **System**, and then pick the **Environment** tab. In this sheet system or user specific environment variables can be configured.

H.2. ICROOT

ICROOT specifies the Interactive COBOL root directory. ICROOT is used to find the various system directories (help, print, term).

The syntax is:

```
ICROOT=dir
```

Where

dir

Specifies the directory where to find the various Interactive COBOL directories. Usually this should be set the current revision directory.

If ICROOT is not set, then the ICROOT that was stored in the registry at startup is used.

H.3. 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.

In a network environment, ICTMPDIR should be used on each workstation to point to a local disk to reduce network traffic.

H.4. ICPERMIT_MACHINE

ICPERMIT_MACHINE specifies the default license server machine and TCP/IP port.

The syntax is:

```
    ICPERMIT_MACHINE=machine[:port]
```
or
```
    ICPERMIT_MACHINE=:port
```

Where

Machine

Specifies an ip-address or a machine-name on which an ICPERMIT license server is running. The default is the current machine (localhost).

Port

Specifies the TCP/IP port which ICPERMIT is using. The default is 7334. The license server and the clients MUST use the same port.

Some of the programs that look for the ICPERMIT_MACHINE environment variable are ICOBOL, ICRUN, ICRUNW, the ICNETD surrogates (ICIOS, ICRUNRS, ICTHINS), ICPERMIT, and any other **ICOBOL** licensed executables.

H.5. Executable Name

All Windows executable utilities support an environment variable of the same name as the utility in upper-case. For example, 'iccheck' 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.

I. Reporting Problems

If you have problems, please first review the manual to recheck your installation and operation. Next check APPENDIX C for a list of general problems to see if any discuss your particular incident. Next review the moments just before the problem occurred to see if you changed anything, i.e., changed the configuration file, added a new device driver, changed attributes, etc.

If your problems still persist fill out a Support Information Request (SIR) form using the one from the end of this manual as a guide, while reviewing the following recommendations and submit the problem. You can also use the online form at www.icobol.com .

1) Completely fill out a copy of the SIR form at the end of this manual. Please include your name, address, and telephone number, FAX number, and E-mail address.

2) Each SIR should document only one problem or suggestion.

3) If the problem is a COBOL problem, a simple test program that reproduces it should be provided. The shorter the better. Include the source and .CX files for the program, along with any data files needed. If this program requires user input (name, password, data-entry, etc.), please provide a script for how to logon and duplicate the problem.

4) If data files are being corrupted then please send `before' and `after' images of the file.

5) If the problem is with a utility send the exact command line of how the utility was invoked.

6) For problems involving devices such as terminals, printers, etc., please send a copy of the current configuration file.

7) Label all media and verify that it is readable and includes everything needed to reproduce the problem.

8) Media should be in COPY format on either a 1.44MB diskette or a CD-ROM. BACKUP format should only be used as a last resort and if so make sure you specify the revision of the BACKUP was made with. CD-ROM is the best for larger amounts of data.

9) Send a copy of the output of ICINFO.

10) If possible, E-mail the SIR to support@icobol.com, fax the SIR to the Support fax number (919-851-4609), or send it by "snail" mail to:

ICOBOL Support
Envyr Corporation
4904 Waters Edge Drive, Suite 160
Raleigh, NC 27606
USA.

Other items that would be helpful to include are:

Is the problem repeatable?

Can you work around the problem?
If so, how?

What is the easiest way to get the problem?

What is the importance of obtaining a solution to the problem?
(Urgent, moderate, low, nuisance)

II. INSTALLATION

A. Introduction

This chapter describes how to install the Interactive COBOL on Windows release along with general guidelines for use on Windows. The Interactive COBOL on Windows download full release (for 3.60 , win360.exe) includes all parts (runtime, development, ICSP2, ICQPRW, ICODBC, ICNETD, ICIDE, link kit, and user library) of the Interactive COBOL release. All or portions can be installed as needed for a particular installation. A separate (smaller) version (win360r.exe) is available that ONLY includes the Runtime and ThinClient client portions.

After installation, either the ICCONFIG or ICEDCFW utility should be run to build custom configuration file(s) (.CFI), any custom terminal description files (.TDI), and any custom printer translation files (.PTI) that are required for execution.

A runtime uses about 10MB while a full install can take upwards of 80MB. The installation program ensures that enough disk space is available to install the selected components.

Installation must be done on every machine that is to use ICOBOL even if the software is to be used from a remote machine as this allows the registry, appropriate shortcuts, and file associations to be set on each machine.

An install for the same revision can be run several times to ADD additional components or to CHANGE any of the selection values that the install program prompts for. An install CANNOT remove pieces that have already been installed.

Batch/scripted installs can be done and more information is shown on page 44 for those installs.

B. Software Installation

B.1. Introduction

This section provides the necessary steps to get the Interactive COBOL on Windows software installed.

Interactive COBOL on Windows is available in several fashions:

A) On CD-ROM that contains the complete product set for a particular release, or
B) In a self-extracting executable for either full or runtime-only release that is downloadable .

For the CD-ROM release, the ICOBOL CD-ROM should be inserted into the cd-rom drive. A Win directory is provided that contains the complete Windows release. Additional directories are provided with documentation, download files, a complete set of readme files, and all the directories for supported UNIX platforms.

If a revision of Interactive COBOL on Windows **before 2.50** is installed, it must be removed before installing newer revisions of Interactive COBOL. This can be done using the **Add/Remove programs** from the **Control Panel** and selecting Interactive COBOL.

On Windows, installation can only be done by Administrators.

NOTE: The installation will use the directory pointed to by the environment variable TEMP. If this temporary directory cannot be accessed, an appropriate message will be issued.

When updating a revision on Windows, all **ICOBOL** services will be stopped and re-started.

The installer checks for a previous revision. If a previous revision that does not have its install information available is detected, the installation prompts you to CANCEL the install and remove the old revision. If the install information is available, (by reading a previous install.txt file) then that information is used to pre-load all the needed prompts in the installation.

Current downloadable installers include:

win360.exe	Full installer, allows all features
win360r.exe	Runtime-only installer
win360g.exe	ThinClient (gui) client installer
win360c.exe	ThinClient (char) client installer
doc360.exe	Documentation installer (Language Reference/Developer's Guide, UNIX Guide, Windows Guide, Utilities, SP2 Development, FormPrint Development)

Where 360 represents the revision of the product.

To do a quick upgrade of an already existing **ICOBOL** install using the same settings as currently installed all that needs to be done is:

```
win360 /a /u
```

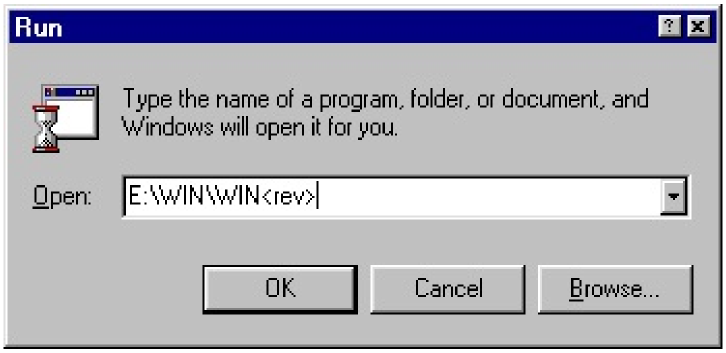
Interactive COBOL on Windows can be provided in several fashions: A) On CD-ROM and B) In a single executable file (that is downloadable). Please select the step that matches your release.

B.2. CD-ROM

For a CD-ROM release:

1. Insert the Interactive COBOL CD-ROM into the CD-ROM drive. The CD-ROM release is contained in a single executable file in the WIN directory of the CD-ROM.

2. From the **Start** menu choose **Run**. In the Open field of the Run dialog, type E:\WIN\WIN360 (or whatever drive letter you inserted the media into other than E: and any different revision number other than 360).



Type the name of a program, folder, or document, and Windows will open it for you.

Open: E:\WIN\WIN<rev>

Proceed to Common Installation below. (Page 30)

B.3. Self-extracting Executable

The syntax for the installation program when using the single self-extracting .exe is shown below:

```
win360.exe [/a [/r|/s] [/u] [/f1<script-path-name>] [/f2<log-path-name>] ]
```

Where

/a Required if any other switches are needed when using the self-extracting .exe.

/r Record mode. Writes a script file (.iss) to be used later for an install with no prompting (silent). Cannot be used with the /s switch. If no /f1 switch is used, the file setup.iss is created in the \WINDOWS (or \WINNT) directory on the boot drive.

/s Silent mode. Reads a script file previously produced by a Record mode. Allows for unattended or batch type installs. During a silent install, no user requested prompts are made. Cannot be used with the /r switch. If no /f2 switch is used, the file setup.log is created in the directory where the .iss file resides.

/f1<*script-path-name*>
Specifies the script file name. Is generally used to read a previously recorded script. Must be a fully qualified name including the .iss extension.

/f2<*log-path-name*>
Specifies the log file name. Is generally used to specify a specific log file for a silent install.

/u Update install. If a valid revision (2.50 or up) is already installed, Update will perform a silent/batch update using the exact same information that is currently installed. When updating a revision on Windows, all services will be stopped, the updated performed, and then the services will be re-started.

Switches must be separated by spaces. The /f1 and /f2 switches cannot have a space separating the switch from the filename.

If not using the self-extracting .exe (i.e., using setup.exe) the /a switch is not allowed or needed.

To build a sample script, execute the install with the record option and take the appropriate selections. After the install is completed, the recorded script can be found in the \WINDOWS directory (\WINNT on Windows 2000) as setup.iss. To start in record mode, enter "/a /r" after the win360.exe argument on the command line. The space IS required. Thus you would enter:

```
 win360.exe /a /r
```

To re-do an install with the exact same responses, a silent install can be done by entering "/a /s /f1<path-name>" after the win360.exe on the command line. The spaces are required and pathname should be the path to the script file. When doing a silent install, a log file (SETUP.LOG) is written to the \WINDOWS directory (\WINNT on Windows 2000) that tells whether the install succeeded. Thus you would enter:

```
win360.exe /a /s /f1c:\test\setup.iss
```

A runtime/thinclient client only version of the installation program is provided as win360r.exe. It only supports either a runtime and/or thinclient client install either local or client. It cannot be used to update an install that has more than runtime or thinclient client selections.

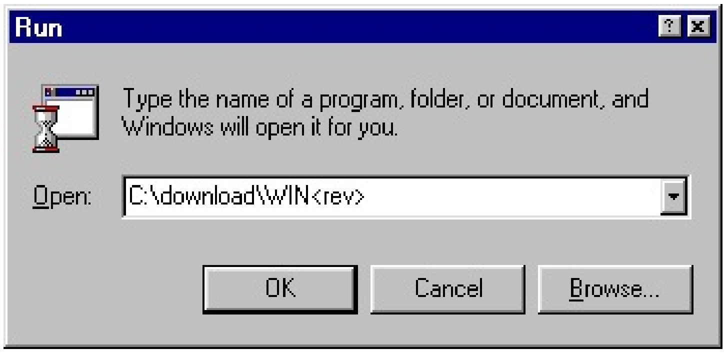
Two separate thinclient client-only installers are also provided as win360c.exe (icrunrc) and win360g.exe (icthinc). Each only supports the indicated thinclient client install. They cannot be used to update an install that has more than that single thinclient.

The splash screen of the install will show the type of installation available.

<u>For a downloaded release</u>:

1. The downloaded release is contained in a single executable file.

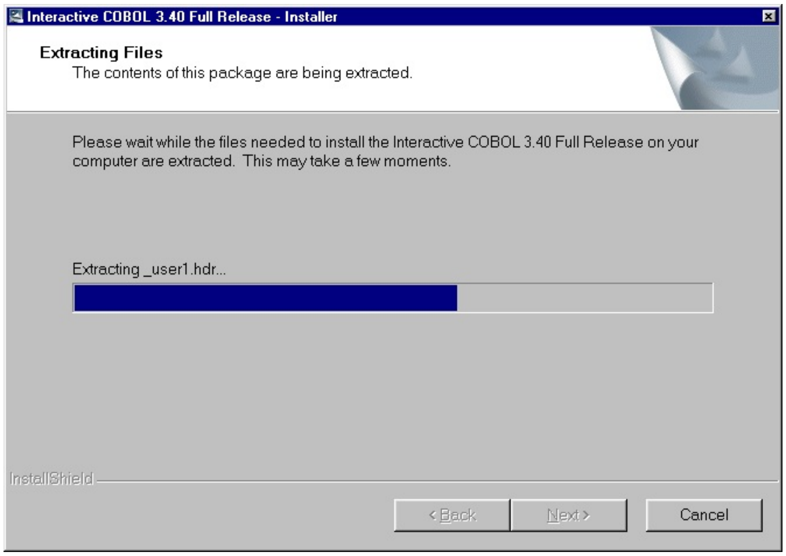
2. From the **Start** menu choose **Run**. In the Open field of the Run dialog, type the complete path to the downloaded executable. For example, C:\download\win360.exe. (or whatever drive letter and directory the downloaded file is in other than C:\download and any different revision number other than 360).



Type the name of a program, folder, or document, and Windows will open it for you.

Open: C:\download\WIN<rev>

The following screen (or one similar) will be shown while the install uncompresses the needed files.



Interactive COBOL 3.40 Full Release - Installer

Extracting Files

The contents of this package are being extracted.

Please wait while the files needed to install the Interactive COBOL 3.40 Full Release on your computer are extracted. This may take a few moments.

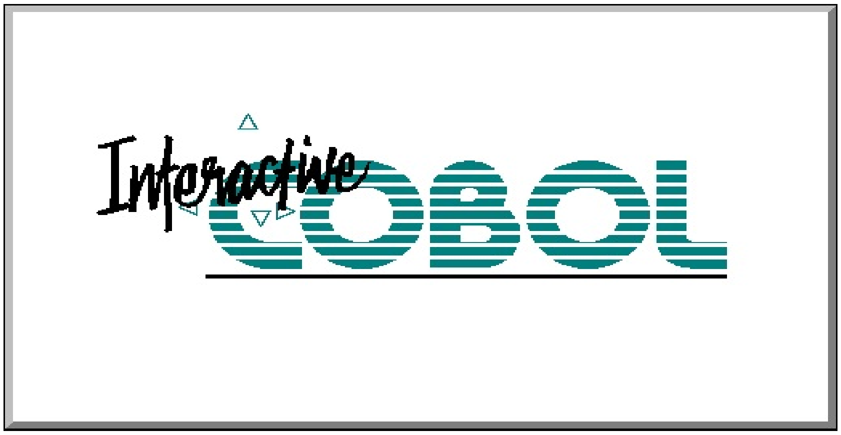
Extracting _user1.hdr...

Proceed to Common Installation below. (Page 30)

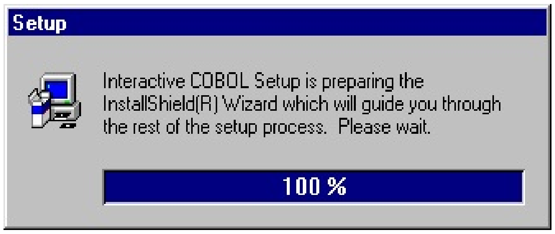
B.4. Common Installation

3. From this point on, the installation will step through a series of questions and answers to allow for a custom installation of Interactive COBOL on Windows on this particular machine. Questions will include installation directory, what components to install, licensing options (remote or use a local license file), whether shortcuts and file associations should be installed, and working directory. At any time the installation can be canceled and no software will be installed or you can back up and redo a particular step.

Following is a typical installation series. The ICOBOL screen will be shown as:

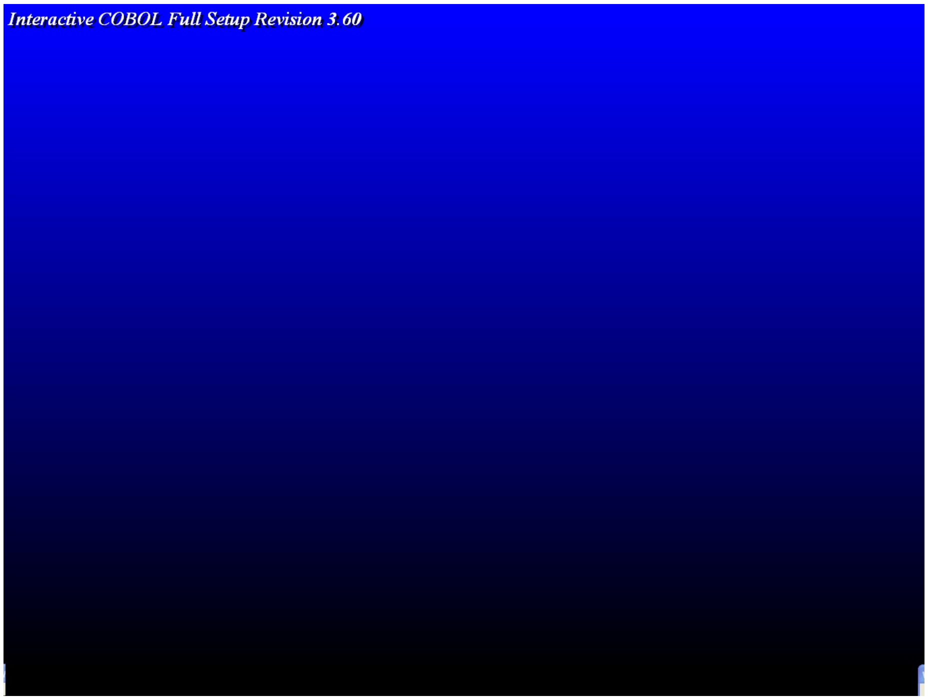


Followed by:

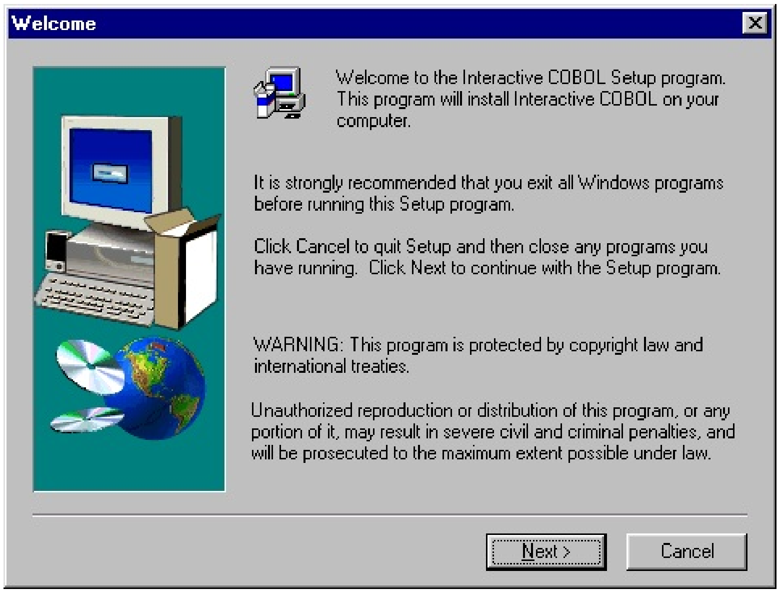


Interactive COBOL Setup is preparing the InstallShield(R) Wizard which will guide you through the rest of the setup process. Please wait.

The splash screen that will be shown throughout the rest of the install will be shown which includes the revision number of **ICOBOL** being installed. If this is not the correct revision then cancel the install.


Interactive COBOL Full Setup Revision 3.60

In addition, each splash screen shows the type of install, Full, Runtime, or ThinClient in addition to the revision.



Welcome to the Interactive COBOL Setup program. This program will install Interactive COBOL on your computer.

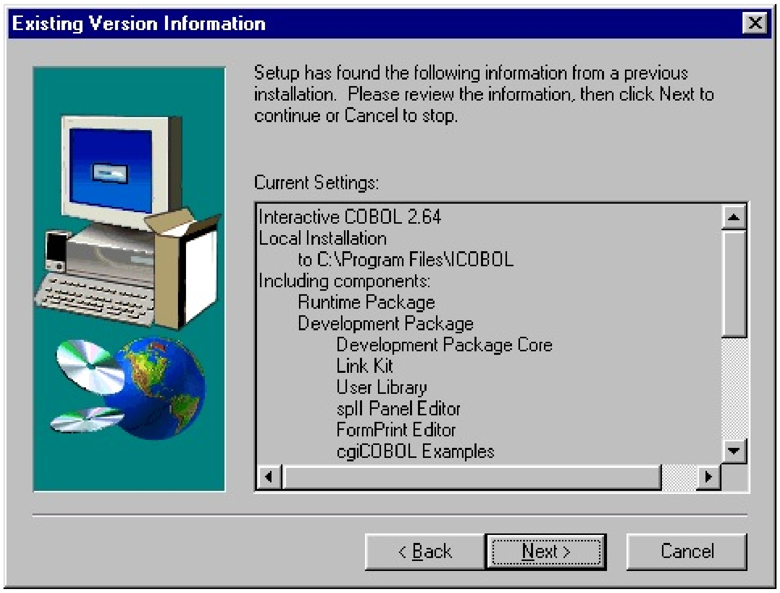
It is strongly recommended that you exit all Windows programs before running this Setup program.

Click Cancel to quit Setup and then close any programs you have running. Click Next to continue with the Setup program.

WARNING: This program is protected by copyright law and international treaties.

Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under law.

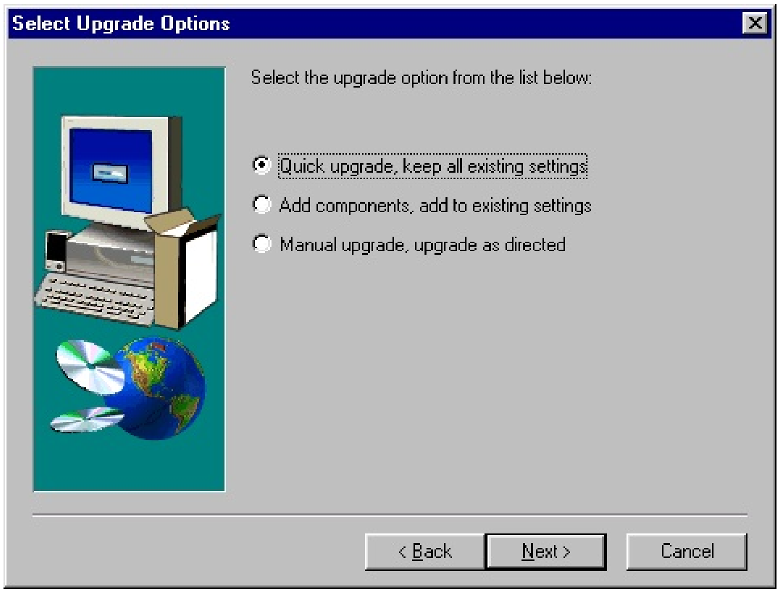
The installer then checks for a previous revision of **ICOBOL**. If no previous **ICOBOL** was found, skip the next two screens and move to the screen about Installation Mode on page 33. If a revision before 2.50 (i.e., 2.3x and 2.4x) was found then it must be uninstalled before installing **ICOBOL** 3 and a prompt to that effect is shown allowing this install to be cancel'ed. If a revision 2.50 or forward is detected, then the following 2 screens are shown allowing an in-place upgrade to be performed.



Setup has found the following information from a previous installation. Please review the information, then click Next to continue or Cancel to stop.

Current Settings:

```
Interactive COBOL 2.64
Local Installation
      to C:\Program Files\ICOBOL
Including components:
      Runtime Package
      Development Package
            Development Package Core
            Link Kit
            User Library
            spII Panel Editor
            FormPrint Editor
            cgiCOBOL Examples
```



Select Upgrade Options

Select the upgrade option from the list below:

● Quick upgrade, keep all existing settings

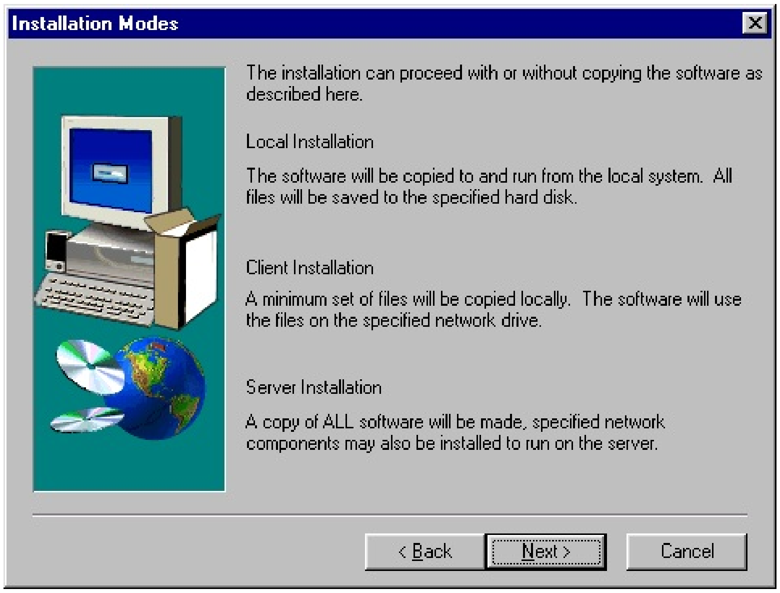
○ Add components, add to existing settings

○ Manual upgrade, upgrade as directed

A Quick upgrade is exactly that, no more questions will be asked and **ICOBOL** will be upgraded with all the current settings. If Quick is selected, proceed to the "Setup Complete" screen on page 42.

An Add or Manual upgrade will continue to prompt for all the individual settings just as a new install.

A screen that gives a brief explanation of the three types of installation (Local, Client, and Server) available is show. More on client/server is shown on page 44.



Installation Modes

The installation can proceed with or without copying the software as described here.

Local Installation

The software will be copied to and run from the local system. All files will be saved to the specified hard disk.

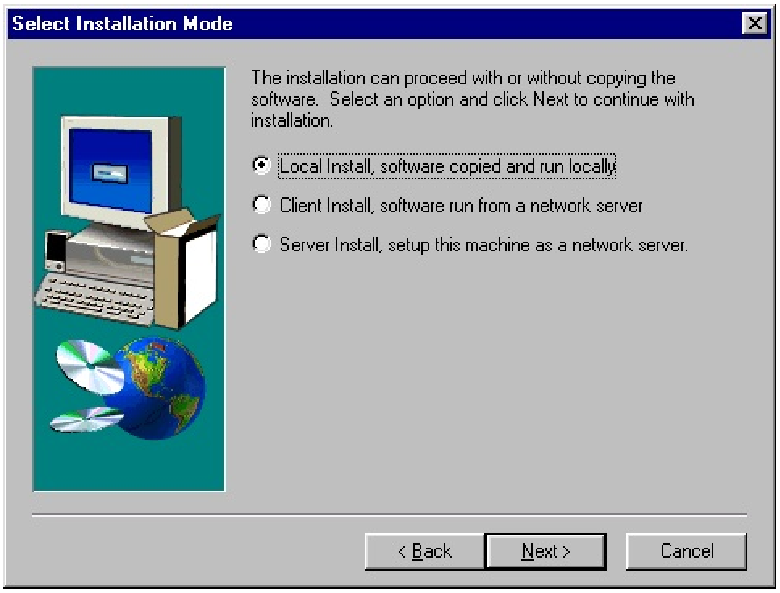
Client Installation

A minimum set of files will be copied locally. The software will use the files on the specified network drive.

Server Installation

A copy of ALL software will be made, specified network components may also be installed to run on the server.

Then the installation prompts as to which type (Local, Client, or Server) to install. Even if software is to be used from a remote machine (Client) some files (uninstall, driver files, and services files) and the needed shortcuts and file associations must be installed on the local machine. A client install requires that the server has been previously installed.



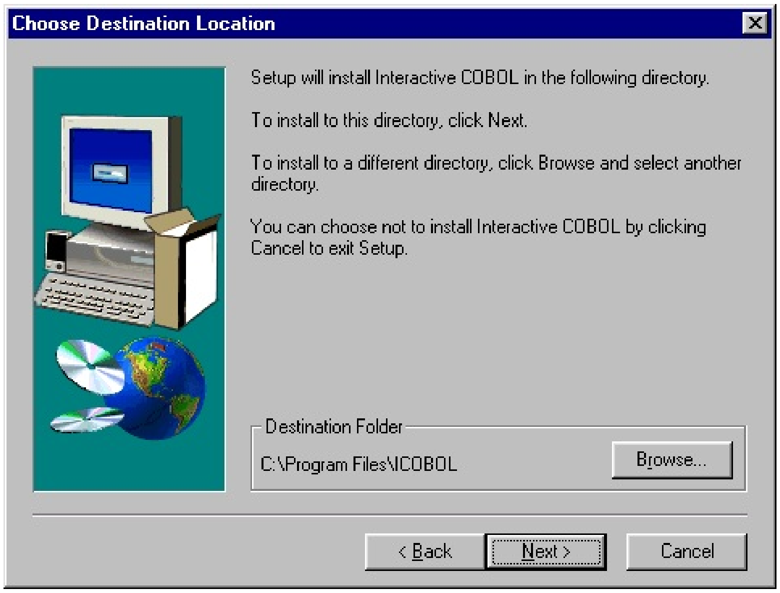
Select Installation Mode

The installation can proceed with or without copying the software. Select an option and click Next to continue with installation.

● Local Install, software copied and run locally

○ Client Install, software run from a network server

○ Server Install, setup this machine as a network server.



Choose Destination Location

Setup will install Interactive COBOL in the following directory.

To install to this directory, click Next.

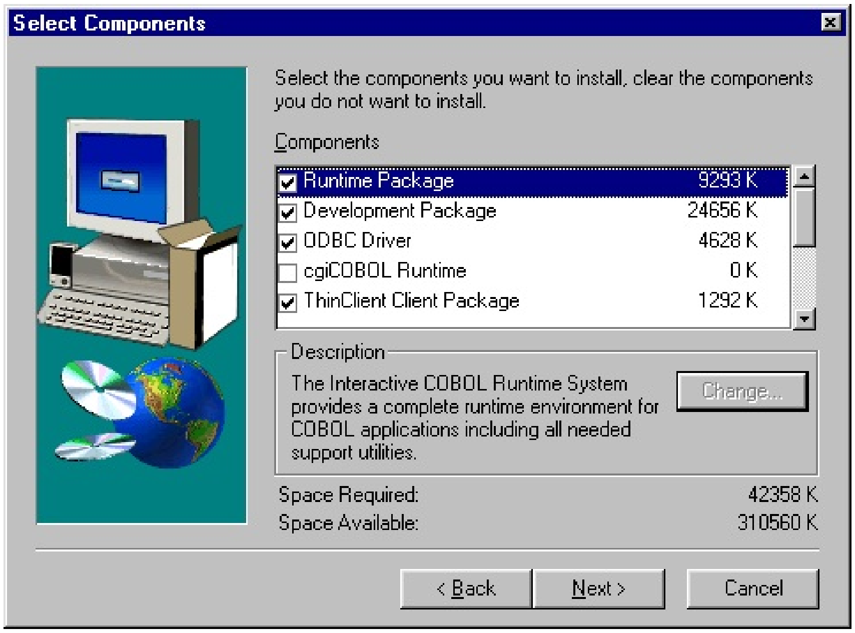
To install to a different directory, click Browse and select another directory.

You can choose not to install Interactive COBOL by clicking Cancel to exit Setup.

Destination Folder

C:\Program Files\ICOBOL

The Destination location is where any needed files are loaded. This location is set to the default ICROOT entry and stored in the registry. If remote software is being used this location is still required for files that are required to be local but the default ICROOT entry will be set to the remote location.



Select Components

Select the components you want to install, clear the components you do not want to install.

Components

☑	Runtime Package	9293 K
☑	Development Package	24656 K
☑	ODBC Driver	4628 K
□	cgiCOBOL Runtime	0 K
☑	ThinClient Client Package	1292 K

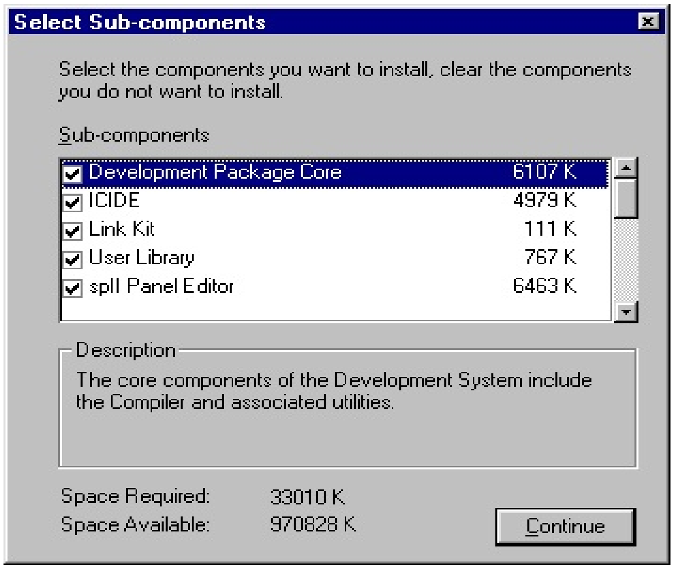
Description

The Interactive COBOL Runtime System provides a complete runtime environment for COBOL applications including all needed support utilities.

Space Required: 42358 K
Space Available: 310560 K

On Windows, the above Select Components screen includes two additional selections for the ICPERMIT License Server and the ICNETD File Server that should be selected if licensing is required and/or if client/server file access is required. If this machine is only to act as a license server, then only the ICPERMIT License Service box needs to be selected. If this machine is to act as a license server and file server then only the two service boxes need to be checked. If this machine is also to run **ICOBOL**, then all the needed boxes should be selected.

Some of the packages have sub-components that can be selectively installed. The Development Package is one of those and it includes Core components, link kit, user library, Sp2, FormPrint, cgiCOBOL examples, and the IDE. If you select the Development Package and then select Change you will see the following, allowing sub-components to be selectively installed. If you are upgrading from a Development release that did NOT include the ICIDE you must enter this screen and manually select the ICIDE sub-component.



Select Sub-components

Select the components you want to install, clear the components you do not want to install.

Sub-components

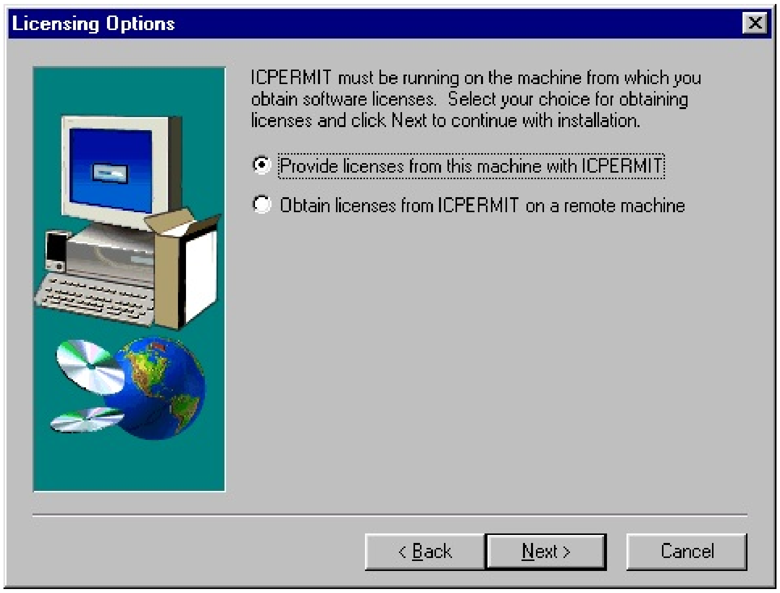
☑	Development Package Core	6107 K
☑	ICIDE	4979 K
☑	Link Kit	111 K
☑	User Library	767 K
☑	spII Panel Editor	6463 K

Description

The core components of the Development System include the Compiler and associated utilities.

Space Required: 33010 K
Space Available: 970828 K

Next licensing options are shown.



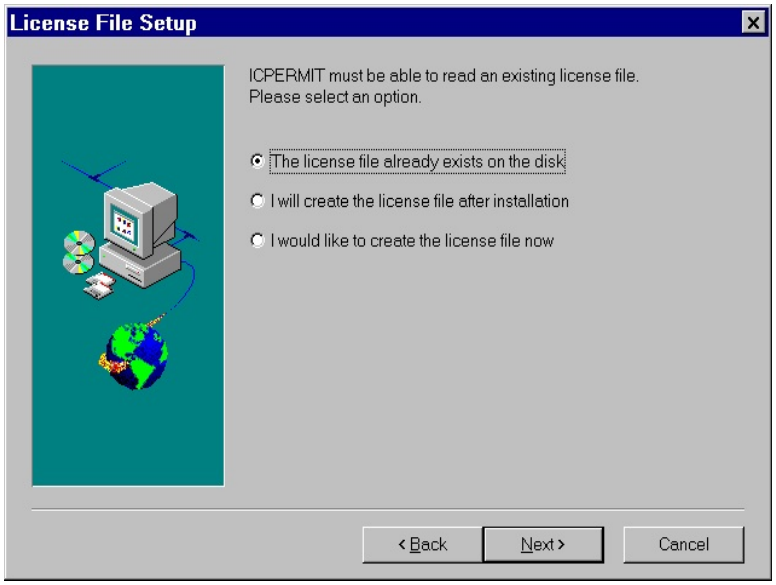
Licensing Options

ICPERMIT must be running on the machine from which you obtain software licenses. Select your choice for obtaining licenses and click Next to continue with installation.

● Provide licenses from this machine with ICPERMIT

○ Obtain licenses from ICPERMIT on a remote machine

If this machine is to run ICPERMIT, then the following two screens will be given.



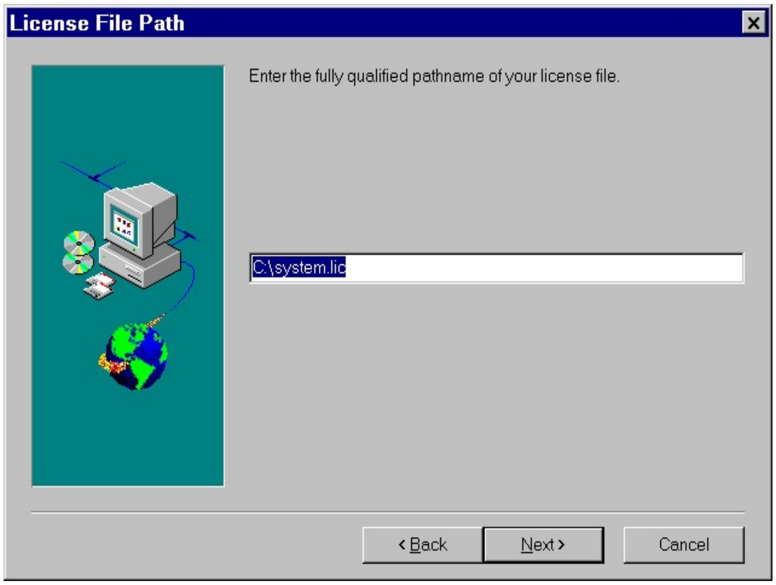
License File Setup

ICPERMIT must be able to read an existing license file. Please select an option.

● The license file already exists on the disk

○ I will create the license file after installation

○ I would like to create the license file now

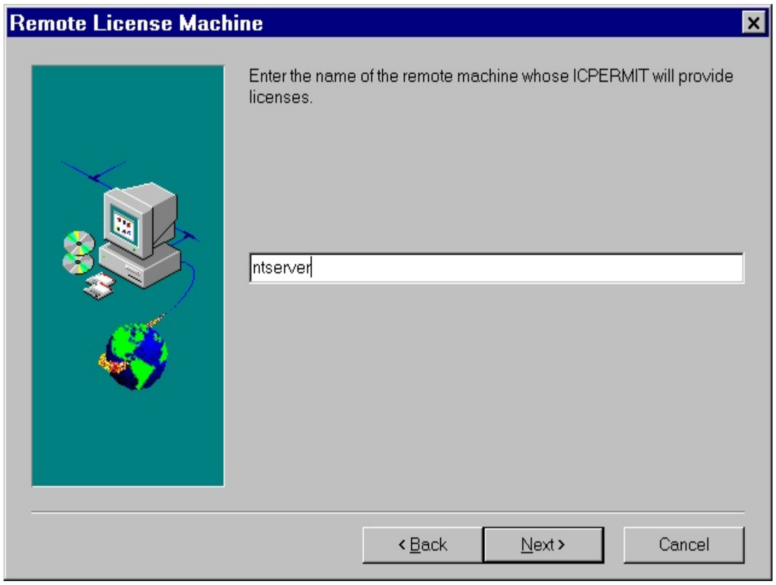


License File Path

Enter the fully qualified pathname of your license file.

C:\system.lic

If remote licensing was selected, the following screen will be given. Enter the remote machine on which an ICPERMIT is to be running. (Do not enter the leading "\\" of a machine-name.) **NOTE:** If you had needed to change the TCP/IP port for the license server then it can be changed here by just entering the *machinename*:*port-address* in this screen. This will not normally be needed.

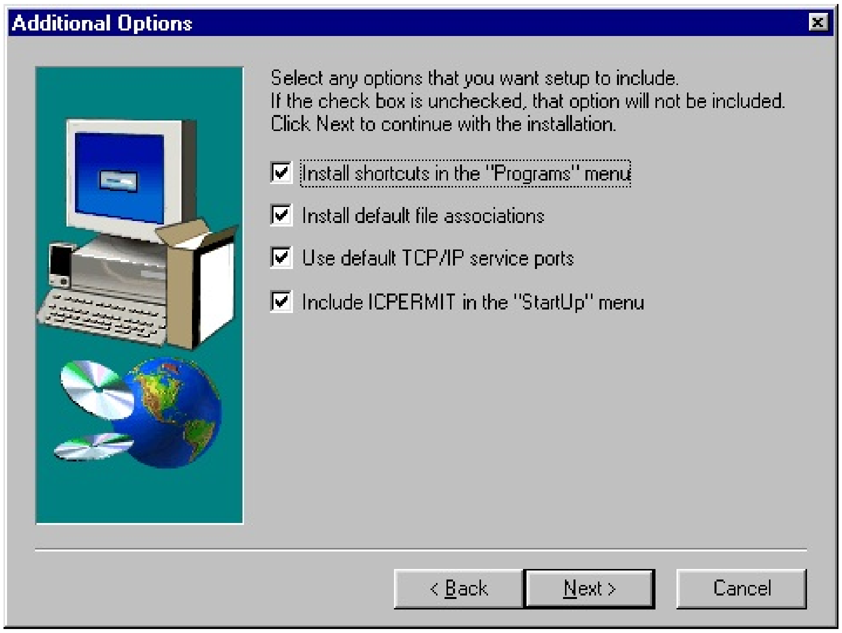


Remote License Machine

Enter the name of the remote machine whose ICPERMIT will provide licenses.

ntserver

Finally any additional options are requested.



Additional Options

Select any options that you want setup to include.
If the check box is unchecked, that option will not be included.
Click Next to continue with the installation.

☑ Install shortcuts in the "Programs" menu

☑ Install default file associations

☑ Use default TCP/IP service ports

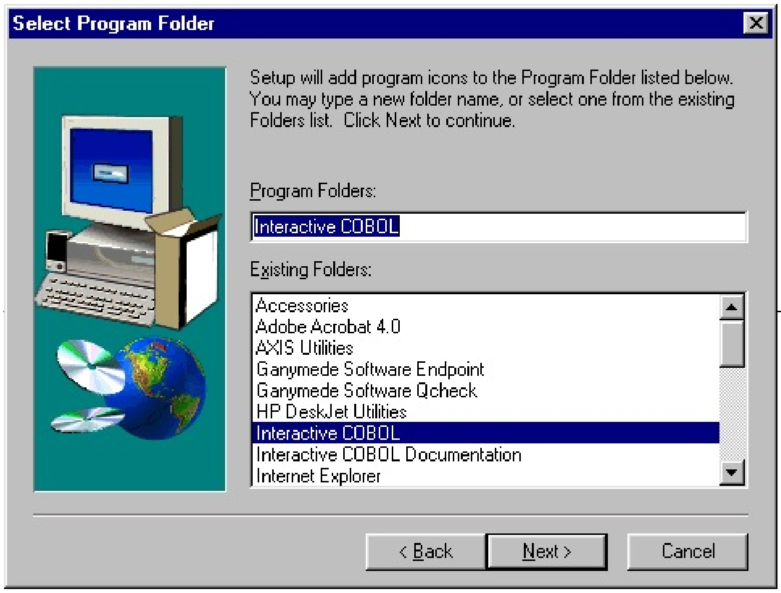
☑ Include ICPERMIT in the "StartUp" menu

The shortcuts that are installed include an **ICOBOL** folder in the programs group which includes shortcuts to the major executables ICEDCFW, ICIDE, ICPERMIT, ICRUNW, ICRUNW (debug), ICRUNRC, ICTHINC, ICQPRW, ICREVW, ICSP2, ODBC Administrator that were loaded, along with two folders containing Console Applications and all the Readme files. See page 43 for a sample screen showing these targets.

The selection “Install default file associations” causes the module icshellx.dll to be installed along with file associations linking common **ICOBOL** file types to certain execution paths. These default file associations can be seen in the Utilities Manual in the ICSHELLX chapter.

If the selection “Use default TCP/IP service ports” is unchecked, then a separate screen will be shown that allows the default ICPERMIT and ICNETD (if installed) service ports to be set to a value other than the default values of 7334 and 7333 respectively. This should only be done by an experienced user to solve a port conflict.

The selection “Create Default server account” is shown. This account is what will be used by client Windows machines to access this server install. See more on client/server on page 44.

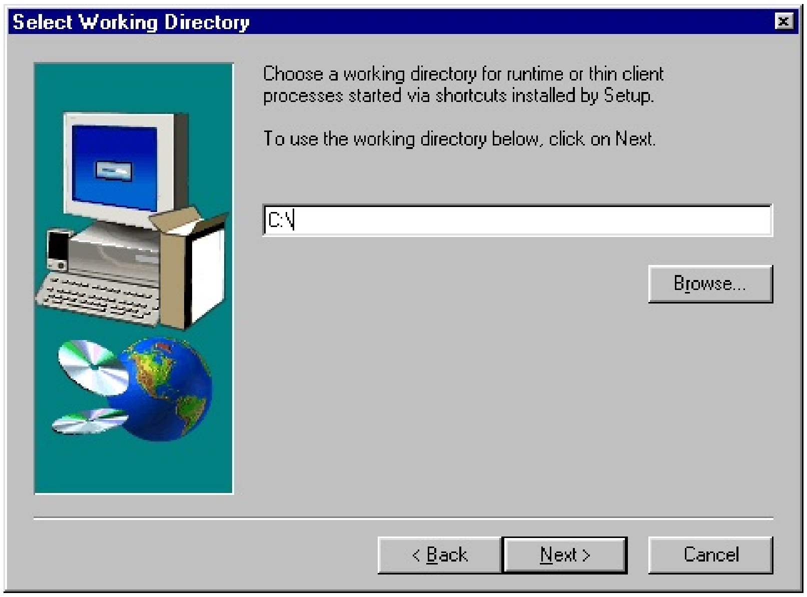


Select Program Folder

Setup will add program icons to the Program Folder listed below. You may type a new folder name, or select one from the existing Folders list. Click Next to continue.

Program Folders:
Interactive COBOL

Existing Folders:
Accessories
Adobe Acrobat 4.0
AXIS Utilities
Ganymede Software Endpoint
Ganymede Software Qcheck
HP DeskJet Utilities
Interactive COBOL
Interactive COBOL Documentation
Internet Explorer


Select Working Directory

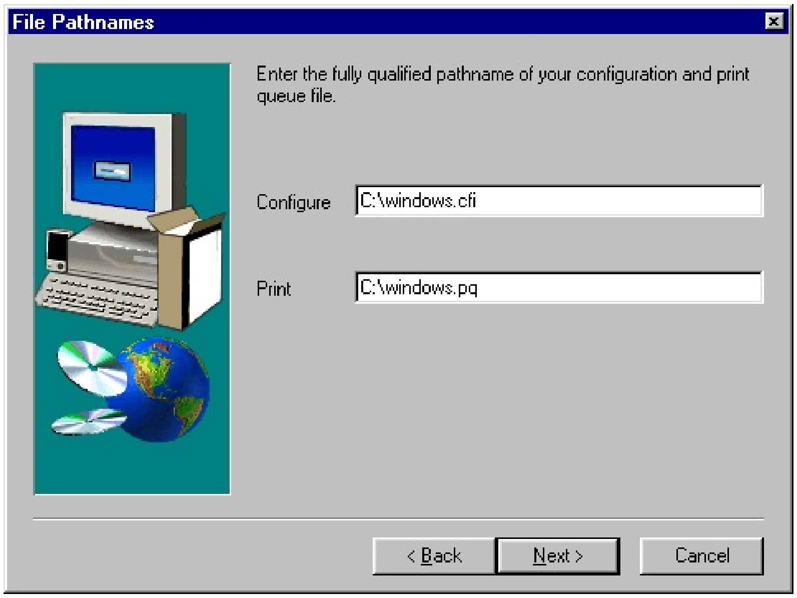
Choose a working directory for runtime or thin client processes started via shortcuts installed by Setup.

To use the working directory below, click on Next.

C:\

This working directory is where all instances of **ICOBOL** executables will start whenever a working directory is not known (as icons, from startup, or services, etc.). In particular, runtimes (icrun) started by ICEXEC for COM ports and runtimes (icthins and icrunrs) started by ICNETD for ThinClient servers will start in this directory. This information is stored in the registry. If the directory does not exist, you will be prompted as to whether to create the directory.

Next the location for the configuration and printer control queue are prompted for:

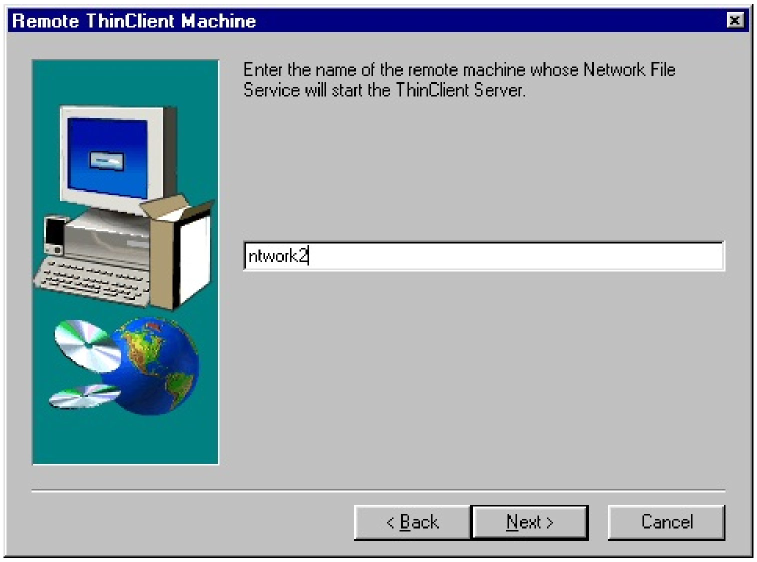

File Pathnames

Enter the fully qualified pathname of your configuration and print queue file.

Configure: C:\windows.cfi

Print: C:\windows.pq

If a ThinClient was selected, then the following screen is show to select the Remote machine for the ThinClient client to connect with.

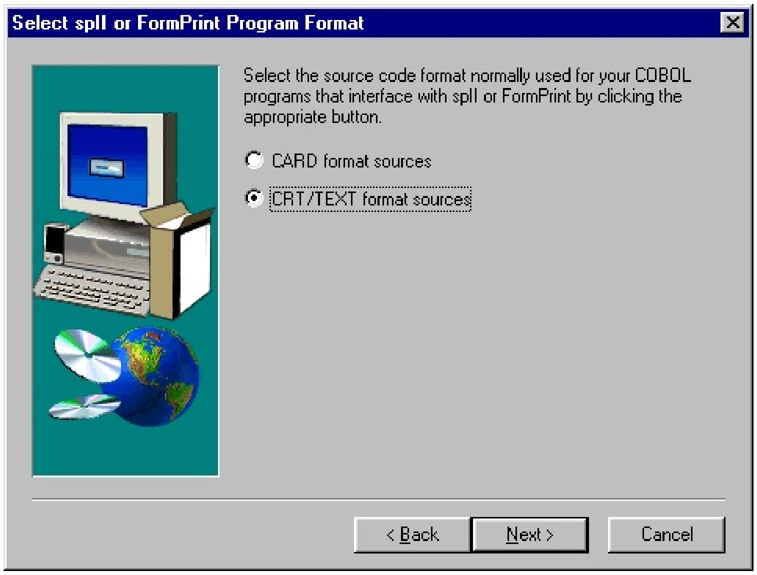


Remote ThinClient Machine

Enter the name of the remote machine whose Network File Service will start the ThinClient Server.

ntwork2

If the Development Package with either the sp2 Panel Editor or the FormPrint editor were selected, then the following screen is shown to allow the default source format.



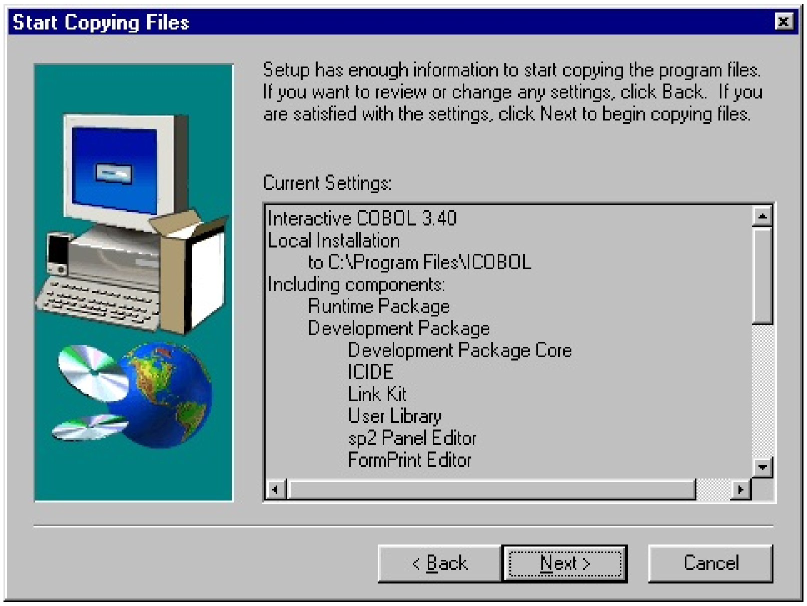
Select spII or FormPrint Program Format

Select the source code format normally used for your COBOL programs that interface with spII or FormPrint by clicking the appropriate button.

○ CARD format sources

● CRT/TEXT format sources

Finally you are ready to start copying files.

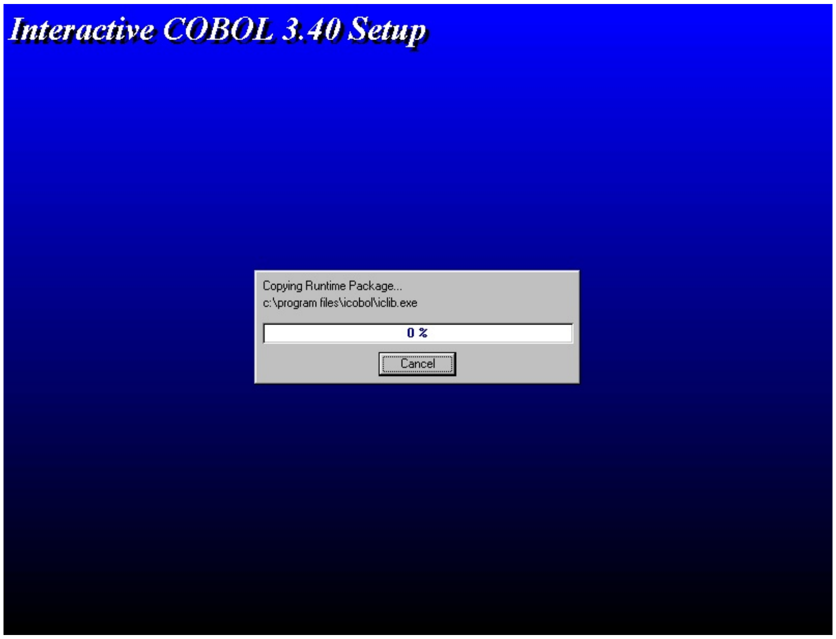


Setup has enough information to start copying the program files. If you want to review or change any settings, click Back. If you are satisfied with the settings, click Next to begin copying files.

Current Settings:

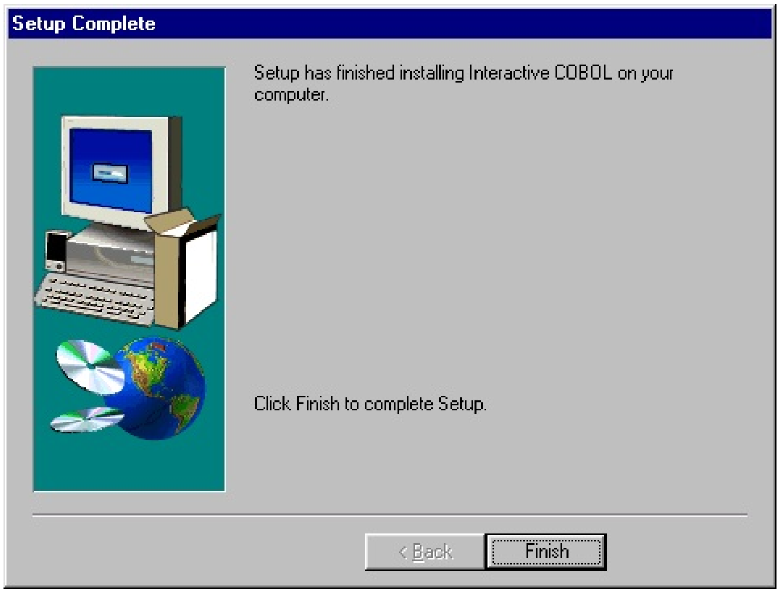
```
Interactive COBOL 3.40
Local Installation
    to C:\Program Files\ICOBOL
Including components:
    Runtime Package
    Development Package
        Development Package Core
        ICIDE
        Link Kit
        User Library
        sp2 Panel Editor
        FormPrint Editor
```

At this point the installation ensures that enough disk space is available to install the selected components. It then copies and installs all the selected components. The following screen is shown while copying the files.



Copying Runtime Package...
c:\program files\icobol\iclib.exe

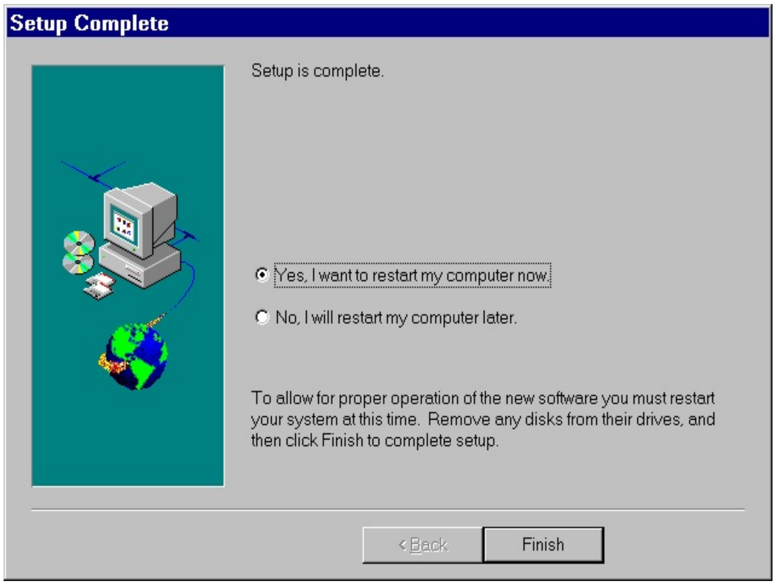
Once the installation is finished, one of the following completion screens will be displayed saying that Interactive COBOL has been successfully installed and may ask you if you wish to restart. If the restart option is given then that says that at some point in the install a file that was in use needed to be updated or a device driver was loaded and a restart will ensure that the correct file is in place. At this point, the Interactive COBOL on Windows software has been installed and an Interactive COBOL folder will be shown with several selections. Restarting at this time will insure that all drivers will be installed and activated. Any of the new **ICOBOL** services that were installed are started without requiring a restart.



Setup Complete

Setup has finished installing Interactive COBOL on your computer.

Click Finish to complete Setup.



Setup Complete

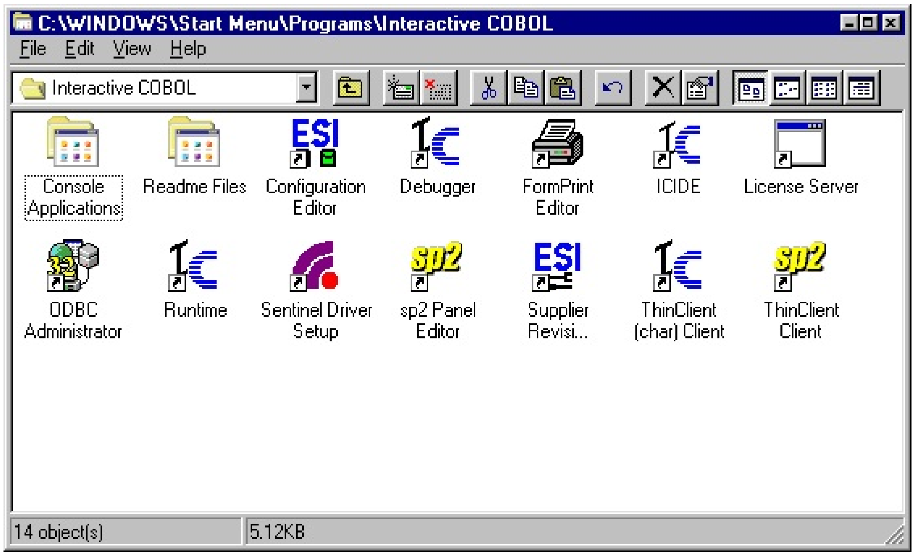
Setup is complete.

● Yes, I want to restart my computer now.

○ No, I will restart my computer later.

To allow for proper operation of the new software you must restart your system at this time. Remove any disks from their drives, and then click Finish to complete setup.

Using the current Interactive COBOL folder, selections can be highlighted to more quickly change any shortcut parameters that need to be changed. Later changes must be done through the **Start** menu, **Settings**, **Taskbar**. Then in the **Taskbar** properties sheet, select the **Start Menu Programs** tab, and its **Advanced**, selection.



If you plan to execute Interactive COBOL executables from a command prompt make sure to add the installation directory to your PATH to find the needed executable files.

Save your media for future reference.

The *Console Applications* folder includes console versions of icconfig, icrun and icrun in debug mode.

The *Readme Files* folder include the **ICOBOL** Readme (readic.txt) that should be read for any new information that has not been put into this manual. In addition there are other readme files that may be useful. These include readme files for cgiCOBOL, sp2 and FormPrint, Installation, and Documentation to name a few.

A sample program, *logon*, has been included on the release along with its source file that shows some examples of Interactive COBOL specific features or other useful features. Some of the features included are: use of terminal Print Pass Through, ability to execute an operating system program, disk space, directory listings, the Get Error Message ability, and some SCREEN HANDLER features to name a few.

The shortcuts setup by the installation program are setup in the “All User’s” profile and you must be Administrator to change these settings.

C. Software Uninstall

To uninstall a previously installed version of Interactive COBOL on Windows use the standard Windows uninstall procedures as described below.

From the **Start** menu, choose **Settings**, **Control Panel**. In the **Control Panel** folder select the **Add/Remove Programs**. In the **Add/Remove Programs** property sheet, scroll through the software that can be removed and select Interactive COBOL. Click on the Add/Remove. An initial screen will be shown do you wish to Uninstall the Sentinel driver and any installed services. Press any key to continue. Next a dialog box will be shown to confirm file deletion. Click Yes to continue. Interactive COBOL will now be removed and a message will be given. Click OK to confirm. Interactive COBOL has now been removed from this machine.

Uninstall removes all the files that the previous installation had installed for a particular version of Interactive COBOL. It will not remove any new files that have been added to the installation directory (or sub-directories). Since new files are not removed, any directory containing a new file is also not removed. It is not recommended to add new files to the installation directory without fully understanding its implications. If a file had been modified from its original it is still removed.

D. Client/Server Install

The Client/Server installation provides the mechanism to store most of the needed **ICOBOL** files on a server and use the software from many clients. This is very helpful in managing software revisions and in controlling total disk usage. It does have a drawback in that the server must be available to run the software and some network bandwidth is consumed by loading programs.

On a server install, all files on the release are copied into the release directory and a client install directory is provided. Only the pieces selected in the server install are actually setup on the server.

On a client install, only a very small subset of files are copied onto the local machine. The software is setup to access the release directory on the server machine. This is usually done via a UNC name like "\\myserver\c\program files\icobol".

Client installs can be performed using the normal Windows install and selecting the Client install option or the setup.exe file can be executed from the client subdirectory in the server release directory.

One note to make is that when doing a client install for Windows clients, the **ICOBOL** services must be installed in a fashion that allows the network to be available before any **ICOBOL** services are started since the executables are being used from a network server. This is done automatically at installation time by using the "+NetworkProvider" Dependency. You are also prompted for a Username/Password that allows the client machine to access files on the server.

The username that is specified must have the following user rights (privileges) on the client:

a) Act as Part of the operating system,
b) Debug programs,
c) Increase quotas,
d) Log on as a service, and
e) Replace a process level token.

On the server, the specified username must have the "Access the computer from the network" user right (privilege). When the Server install was done the installer allowed for this username/password to be setup.

Usually this particular user would not be given "Log on locally" user right. This **ICOBOL** user profile can not be Administrator or localsystem.

Local usernames should be entered as ".\user". Domain usernames should be entered as "domain\user".

NOTE: On Windows 2000, when doing a client install, any executable needed by a service MUST be copied locally.

E. Batch/Scripted Install

The install process can be scripted such that batch installs can be performed.

To build a sample script you must execute the install with the record option and take the appropriate selections. After the install is completed, the recorded script can be found in the \WINDOWS directory (\WINNT on Windows 2000) as setup.iss. To start in record mode, enter "/a /r" after the win360.exe argument on the command line. The space IS required. Thus, you would enter:

```
win360.exe /a /r
```

To re-do an install with the exact same responses, a silent install can be done by entering "/a /s /f1<path-name>" after the win360.exe on the command line. The spaces are required and pathname should be the path to the script file.

When doing a silent install, a log file (SETUP.LOG) is written that tells whether the install succeeded.

During a silent install, no user requested prompts are made.

By using the recording mode and silent installation along with any needed changes to the script file totally batch installs can be done.

LOG File

The SETUP.LOG file contains three sections. The first section, [InstallShield Silent], identifies the version of InstallShield Silent used in the silent installation. It also identifies the file as a log file.

The second section, [Application], identifies the installed application's name and version, and the company name.

The third section, [ResponseResult], contains the result code indicating whether or not the silent installation succeeded. An integer value is assigned to the ResultCode keyname in the [ResponseResult] section. InstallShield places one of the following return values after the ResultCode keyname:

0	Success.
-1	General error.
-2	Invalid mode.
-3	Required data not found in the SETUP.ISS file.
-4	Not enough memory available.
-5	File does not exist.
-6	Cannot write to the response file.
-7	Unable to write to the log file.
-8	Invalid path to the .iss file
-9	Not a valid list type (string or number).
-10	Data type is invalid.
-11	Unknown error during setup.
-12	Dialogs are out of order.
-51	Cannot create the specified folder.
-52	Cannot access the specified file or folder.
-53	Invalid option selected.

SETUP.LOG is the default name for the silent installation log file, and its default location is the directory where the installation's .INS file resides. You can specify a different name and location for the SETUP.LOG using the /f2 switch.

Below is a sample setup.log file

```
[InstallShield Silent]
Version=v5.00.000
File=Log File
[Application]
Name=Interactive Cobol
Version=3.60.000
Company=Egan Systems
Lang=0009
[ResponseResult]
ResultCode=0
```

In addition, an Update switch is available that allows subsequent revisions of **ICOBOL** to update the installed revision using all the same information that was used before. The /U switch can be used at install time to perform this update.

F. Startup Recommendations

When building applications using Interactive COBOL, it is highly recommended to always start the runtime with the Audit switch (-a or -A) such that any errors or warnings can be viewed at a later time if needed.

Another option is to start the runtime with the Quiet switch (-q) such that no runtime generated messages will be displayed on the console screen. Only messages that are in the COBOL application will be displayed. This is especially useful when running a program from the command line, i.e., <icrun -q program>. When running with the Quiet switch (-q), it is even more imperative to have the Audit switch (-a or -A) enabled such that if any errors occur all the associated messages will be displayed somewhere.

Using the Audit switch (-a or -A) and the Information switch (-i) together will provide the maximum amount of information in case a problem does develop.

G. Using serial cards

If multi-port "dumb" serial card(s) are to be used with the runtime, each port on the multi-port card must be installed as separate hardware device. Interactive COBOL does not provide drivers for any serial card, either “dumb” or “smart”.

This can be done by using the Microsoft Communications Port driver or by using a driver supplied by the manufacturer of the multi-port serial card.

The Microsoft Communications Port driver views each port individually so each port on a multi-port card must be installed separately.

If the Microsoft Communications Port driver is to be used follow the instructions below; otherwise follow the instructions provided by the manufacturer of the board.

INSTALLING COMMUNICATIONS PORT DEVICE(S)

1. From the **Start** menu, choose **Settings**, **Control Panel**.

2. From **Control Panel**, double click the **Add New Hardware** icon.

3. Follow the instructions in the Add New Hardware Wizard dialogue boxes that appear:
   a. Click Next, to begin installing your new hardware.
   b. Select No so that Windows will not search for your new hardware, then click Next.
   c. Select Ports (COM & LPT) for the type of hardware to install, then click Next.
   d. Select (Standard port types) for the manufacturer and Communications Port for the model, then click Next.
   e. Click Next, ignoring the resource settings displayed (they are most likely incorrect for your board anyway).
   f. Click Finish.
   g. Click No, so that Windows will not shut down your computer now.

4. Repeat steps 2 and 3 for each port on the board.

5. From **Control Panel**, double click the **System** icon.

6. Select the Device Manager property sheet in the System Properties tabbed dialogue box.

7. Make sure View devices by type is selected at the top of the Device Manager sheet and click the plus sign (+) in front of the Ports (COM & LPT) icon to expand the display to show all of the installed Ports devices.

8. Select the Communications Port (COMn) device installed in step 3 above and click Properties.

9. From the Communications Port (COMn) Properties tabbed dialogue box:
   a. Select the Resources property sheet.
   b. Make sure Use automatic settings is not checked.
   c. Change the Setting based on selection to be Basic configuration 8.
   d. Select Input/Output Range in the Resource type column of the Resource settings list and click Change Setting...
   e. In the Edit Input/Output Range dialogue box, adjust the value to match that of the port on the multi-port board and click OK. Normally, each port of a multi-port board includes a range of eight (8) values (e.g. 100-107) and appears at an Input/Output Range eight (8) values higher than the previous port (e.g. 100, 108, 110, etc...).
   f. Select Interrupt Request in the Resource type column of the Resource settings list and click Change Setting...
   g. In the Edit Interrupt Request dialogue box, adjust the value to match that of the port on the multi-port board and click OK. Normally this is the same for all ports on the multi-port board.
   h. Click OK to exit the Communications Port (COMn) Properties tabbed dialogue box.

11. Click No, so that Windows will not shut down your computer now.

12. Repeat steps 7 through 11 for each Communication Port (COMn) device installed for the multi-port board.

13. Click Close in the System Properties tabbed dialogue box.

14. Click Yes in the System Setting change dialogue box, so that Windows will shut down your computer now.

REMOVING COMMUNICATIONS PORT DEVICE(S)

1. From the **Start** menu, choose **Settings**, **Control Panel**.

2. From the **Control Panel**, double click the **System** icon.

3. Select the Device Manager property sheet in the System Properties tabbed dialogue box.

4. Make sure View devices by type is selected at the top of the Device Manager sheet and click the plus sign (+) in front of the Ports (COM & LPT) icon to expand the display to show all of the installed Ports devices.

5. Select the Communications Port (COMn) device you wish to remove and click Remove.

6. Click OK in the Confirm Device Removal dialogue box.

7. Repeat steps 4 through 6 for each Communication Port (COMn) device installed for the multi-port board to be removed.

III. ICSVCMGR

A. Introduction

ICSVCMGR is the Interactive COBOL Services Manager for Windows. It is used by the installation program to install any **ICOBOL** services (icpermit, icnetd, icexec). It can be used from the command prompt as well. A GUI version is also available. It appears in the Control Panel as **ICOBOL** Services and as a shortcut in the "Interactive COBOL" program group.

B. Command Line

The syntax for ICSVCMGR is:

```
icsvcmgr [-a[:aflag]|-A file|dir[:aflag] [-B path] [-C cmdline] [-D depend]
   [-E error] [-h|-?][-O a|c|d|e|i|q|r|s|t|u|v] [-P password] [-q] [-S a|d|m]
   [-T c|s] [-U username] icexec|icpermit|icnetd
```

Where

-a[:*aflag*] or -A *file*|*dir*[:*aflag*] (Audit)

Enables auditing (default icsvcmgr.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 *path* (Pathname)

Provides the fully qualified pathname to the executable for the indicated service. Required with the create (-O c) and vary (-O v) operations.

-C *cmdline* (Command line)

Provides the command line (options and arguments) to be passed to the service at startup. Use quotes if *cmdline* has embedded spaces. Allowed with the create and vary operations only.

-D *depend* (Dependency)

Provide a dependency for starting a service. A plus(+) in front of a name indicates a group dependency. "+NetworkProvider" is a good choice to insure that the network is running before any **ICOBOL** services are started.

-E c|i|n|s (Error type)

Set the error type for the service. Types are c-critical, i-ignore, n-normal, s-serve. Default normal.

-h | - ? (Help)

Display help text. (All other options are ignored.)

-O a|c|d|e|i|q|r|s|t|u|v (Operation)

Indicates the needed operation to perform. Valid operations are::

a (Add)	add commandline (requires a Command line (-C)) and adds that *cmdline* to the service.
c (Create)	create (requires Pathname (-B)) the service
d (Disable)	disable the service
e (Enable)	enable the service
I (Interrogate)	interrogate the service
q (Query)	query the service
r (Remove)	remove the service
s (Start)	start the service
t (Terminate)	terminate the service
u (Username)	add username (Add a user account to the system, requires -U, -P, and -T)
v (Vary)	vary (requires either Pathname(-B), Command line (-C), or Start type (-S))

-P *password* (Password)

Gives the password for a user account. Required with -O u. Can be "" for no password.

-q (Quiet)

Enables quiet operation.

-S a|d|m (Start type)

Specifies the start type for the service (a-automatic, d-disabled, or m-manual). Automatic is the default. Allowed with the Create and Vary operations only.

-T c|s (account Type)
  Specifies the account type, either c-client or s-server to create. Required with -O u.
-U *user* (Username)
  Provides the username of the account to be added with the appropriate privileges. Required with -O u. No argument (icexec, icnetd, or icpermit) is required.

The "-B path" and "-S adm" options are only valid with the create and vary operations.

The "-C cmdline" option is only valid with the add, create, and vary operations.

The "-P password", "-T type", and "U user" options are only valid with the add username operation. operations.

ICSVCMGR can be used from the command line to manipulate the needed services after installation.

The Start and Terminate operations (-O s and -O t) start and stop the indicated service.

The Query and Interrogate operations (-O q and -O i) give status information for the indicated service.

The Create and Remove operations (-O c and -O r) can be used to totally create or remove the indicated service.

The Vary operation (-O v) can be used to change any of the options (path, cmdline, and/or start type) of an already installed service. This is probably the most useful option since it allows for the command line to be changed as needed for a particular installation.

The Add a user operation (-O u) adds the specified user account with the given username and password and sets the appropriate privileges for either client or server access. If the user account already exists then only the appropriate privilege is added. Requires -U, -P, and -T. **No service argument is allowed or needed.!**

You must have the appropriate permissions from Windows to manipulate services.

At installation time, the following ICSVCMGR calls are made if the service has been selected to be installed. (Assumes executables in the default directory, a working directory set to "C:\test200" that includes the .cfi and .pq files, and a system directory of C:\WINDOWS.)

```
icsvcmgr -B "c:\program files\icobol\icpermit.exe"
   -C "-A:b c:\windows -L c:\test200\system.lic -N w" -O c icpermit


icsvcmgr -B "c:\program files\icobol\icexec.exe"
   -C "-A:b c:\windows -C c:\test200\system.cfi -P c:\test200\system.pq" -O c
   icexec


icsvcmgr -B "c:\program files\icobol\icnetd.exe" -C "-A:b c:\windows" -O c icnetd
```

ICPERMIT, ICEXEC, and ICNETD all start in the default services directory which is usually C:\WINDOWS\SYSTEM32. This note is only useful in that for the default Audit file (-a) that file is created in that directory.

On local or server installs where all files are local to the machine the system account “LocalSystem” is used to start services.

On systems where some files are remote, a special user account must be used to allow access to the remote file as “LocalSystem” cannot access files on a remote system.

Services should not use mapped drives since they are not valid in all startup cases. Use UNC names for all remote files.

If a service, for example ICEXEC, has been setup to use files locally and then a file like the configuration file is moved to a network filename then you would have to do the following to re-configure the Service Manager to start ICEXEC in a different order:

```
   icsvcmgr -O v -D +NetworkProvider -U username -P password icexec
```

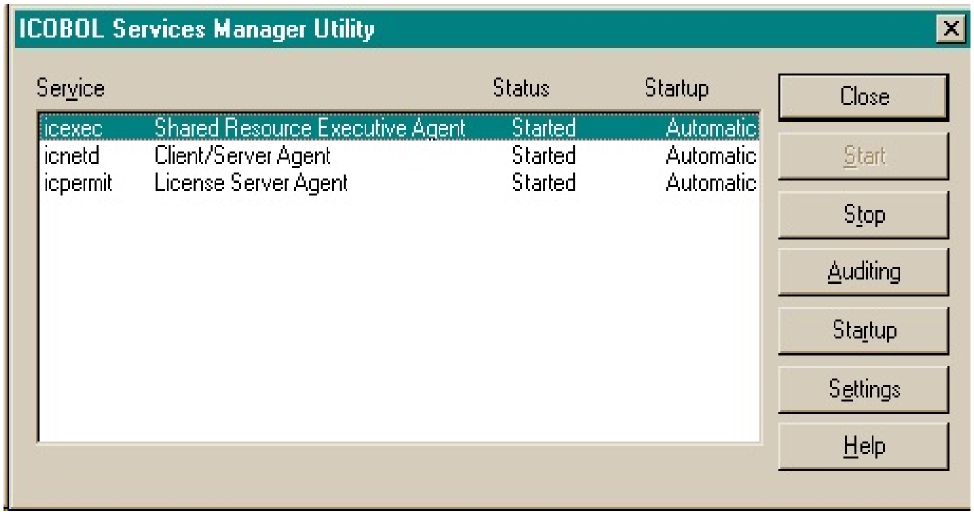
If password is empty you must specify "" (two double-quotes). Remember if no domain is being used the username must be ".\username".

If an **ICOBOL** Service fails to start look at both the Windows event log and the particular log file (.lg) for that service to determine the cause.

C. GUI Service Manager

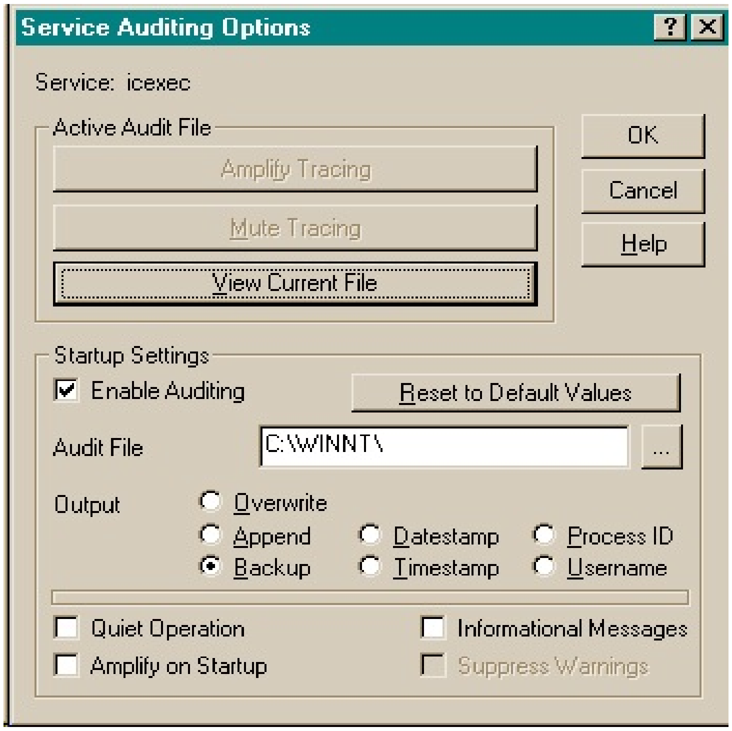
The GUI version of ICSVCMGR is available from the Interactive COBOL programs selection or from the Windows Control Panel. The GUI ICSVCMGR can be used to change any selection and to start and stop the **ICOBOL** services. It CANNOT be used to install or uninstall a service.

The main screen is shown below. From it you can select the appropriate service (only those installed will show) and you can do any of the actions shown on the right. If its grayed out, then the selection is not available.



Service		Status	Startup
icexec	Shared Resource Executive Agent	Started	Automatic
icnetd	Client/Server Agent	Started	Automatic
icpermit	License Server Agent	Started	Automatic

Selecting **Auditing** will show the following:



Service Auditing Options

Service: icexec

Active Audit File

Startup Settings

☑ Enable Auditing

Audit File C:\WINNT\

Output ○ Overwrite ○ Append ● Backup ○ Datestamp ○ Timestamp ○ Process ID ○ Username

□ Quiet Operation

□ Amplify on Startup

□ Informational Messages

□ Suppress Warnings

The name of the service is shown at the top. Actions dealing with the currently active audit file are listed immediately below that in the “Active Audit File” group box. Options dealing with audit file settings are shown in the “Startup Settings” group box on the bottom half of the dialog. The “Startup Settings” will display the settings as found in the Windows service database. These are the settings that will be in force the NEXT time the service is started. Normally, unless changed by someone, these will be the same as the active running service. Use the “View Current File” button to see the settings for an active service. Auditing options include:

Amplify Tracing

Select the “Amplify Tracing” button to send a message to an active running service to “amplify” it’s auditing functions. This action will tell the service to log more information in the audit file. This function is only available to running services and is not supported by the icexec service. If successful, a message saying the “Tracing successfully amplified.” is displayed.

This option is the equivalent of using the “-O a” from the command line versions of the service programs.

```
   icpermit|icnetd -O a
```

Mute Tracing

Select the “Mute Tracing” button to send a message to an active running service to “mute” it’s auditing functions. This action will tell the service to log less information in the audit file. This function is only available to running services and is not supported by the icexec service. If successful, a message saying the “Tracing successfully muted.” is displayed.

This option is the equivalent of using the “-O m” from the command line versions of the service programs.

```
   icpermit|icnetd -O m
```

View Current File

Select the “View Current File” button to view the current audit log file. If the service is still active, this will be the file that is currently being written. If the service is inactive, this will be the last audit file written. If the audit file is being logged with the datestamp, timestamp, Process ID, or username options and the file can not be uniquely identified, an “Audit File” box will be displayed to allow the user to select the specific audit file to open. The snapshot of the file will be opened using the Windows “notepad” editor. From there, the user may view or print the contents of the file. The user will need to close and reopen the file to refresh the contents shown on the screen.

Reset to Default Values

The “Reset to Default Values” button will set the audit file settings to the defined default settings. These are the same settings that would be in effect if you had taken all the default selections during the installation of the software. This button only affects the audit file settings, other settings on the command line are not modified.

This option is the equivalent of using the “-A %WINDIR%:b” from the command line versions of the service programs.

```
    icexec|icpermit|icnetd -A C:\WINDOWS:b
```

Enable Auditing

The “Enable Auditing” button will turn on auditing for the service. When NOT enabled, the “Audit File Pathname” and “Output Flags” options are grayed. In conjunction with the “Audit File Pathname” and “Output Flags” options, this option will determine where the audit file is located.

This option is the equivalent of using either the “-A” or “-a” from the command line versions of the service programs.

```
    icexec|icpermit|icnetd -A path[:aflag]
```

if an “Audit File Pathname” is specified or

```
    icexec|icpermit|icnetd -a[:aflag]
```

if “Audit File Pathname” is left blank

Where
  aflag is determined by the “Output Flags” option

Audit File Pathname

The “Audit File Pathname” edit window will allow the user to set the location of the audit log file. If left blank the audit file will default to the directory where the service is started, normally the “C:\WINDOWS\system32" directory. The user may enter either a pathname or a fully qualified filename. If just a pathname is used, the filename portion will default to the service name. Audit log files will use the “.lg” extension unless specified differently here.

This option is the equivalent of using either the “-A” or “-a” from the command line versions of the service programs.

```
    icexec|icpermit|icnetd -A path[:aflag]
```

if an “Audit File Pathname” is specified or

```
    icexec|icpermit|icnetd -a[:aflag]
```

if "Audit File Pathname" is left blank

Where
aflag is determined by the "Output Flags" option

For example, using the ICEXEC service:

Audit File Pathname	Command Line Equivalent	Audit Log Filename
<blank>	"-a"	"C:\WINDOWS\system32\icexec.lg"
C:\WINDOWS	"-A C:\WINDOWS"	"C:\WINDOWS\icexec.lg"
C:\Logdir\logname.txt	"-A C:\Logdir\logname.txt"	"C:\Logdir\logname.txt"

Output Flags

The "Output Flags" option allows the user to specify the action taken by the service upon startup.

This option is the equivalent of using the ":flag" on either "-A" or "-a" settings from the command line versions of the service programs.

```
    icexec|icpermit|icnetd -A path[:aflag]
```

if an "Audit File Pathname" is specified or

```
    icexec|icpermit|icnetd -a[:aflag]
```

if "Audit File Pathname" is left blank

Where
aflag is determined by the "Output Flags" option

Overwrite	No flag is specified. If a previous log file (.lg) exists, it will be overwritten.
Append	Do not truncate the file, just append to the current file.
Backup	If a previous log file (.lg) exists, rename it to *.lgb and then open a new .lg file.
Datestamp	Add date in the form of _YYYYMMDD before the .lg extension. (YYYY-year, MM-month, DD-day of the month)
Timestamp	Add time in the form of _YYYYMMDDHHmmsshh before the .lg extension. (HH-hour, mm-minute, ss-second, hh-hundredths of seconds.)
Process ID	Add process id (pid) in the form of _NNNN before the .lg extension
Username	Add username in the form _name before the .lg extension.

Quiet Operation

The "Quiet Operation" option allows the user to specify that the service is to run in QUIET mode. This normally means NO output will be sent to either STDOUT or STDERR. Tracing information is still sent to the audit log file if it is enabled.

This option is the equivalent of using the "-q" from the command line versions of the service programs.

```
    icexec|icpermit|icnetd -q
```

Amplify on Startup

The "Amplify on Startup" option allows the user to specify that the service is to start operating with the amplified tracing switch set ON.

This option is the equivalent of using the "-t" from the command line versions of the service programs.

```
   icexec|icpermit|icnetd -t
```

Informational Messages

The "Informational Messages" option allows the user to specify that the service is to start operating with the informational message switch set ON. When set, information type messages as well as warnings and errors are issued by the service.

This option is the equivalent of using the "-i" from the command line versions of the service programs.

```
   icexec|icpermit -i
```

This option is not supported by the icnetd service.

Suppress Warnings

The "Suppress Warnings" option allows the user to specify that the service is to start operating with the suppress warning switch set ON. When set, warning messages are NOT issued by the service.

This option is the equivalent of using the "-N w" from the command line versions of the service programs.

```
   icpermit -N w
```

This option is only supported by the icpermit service.

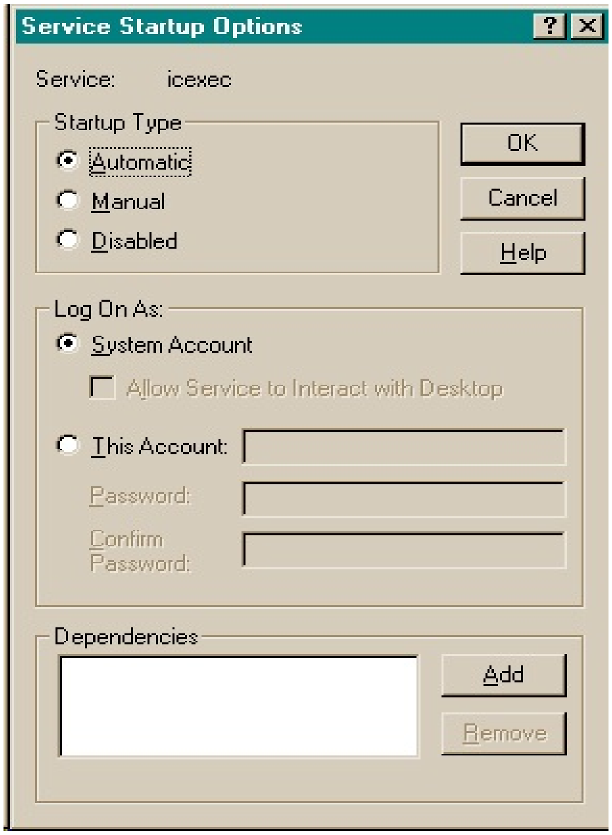
OK Button

The "OK" button will save any changes made by the user to the audit settings and write them to the Windows service manager database. If successful, the user will be returned to the initial status display window.

Cancel Button

The "Cancel" button will allow the user to return to the initial status display window without saving any changes. When unsure of your actions on a particular window, you should use the "Cancel" button (or the escape key) to exit the window.

Selecting **Startup** will show the following:



Service Startup Options

Service: icexec

Startup Type

● Automatic
○ Manual
○ Disabled

Log On As:

● System Account

□ Allow Service to Interact with Desktop

○ This Account:

Password:

Confirm Password:

Dependencies

The name of the service is shown at the top. The "Service Startup Options" dialog allows the user to specify Windows service options. These options are common to all Windows services and the user should refer to the Windows documentation for a more complete description of these options.

Startup Type

The Startup Type defines when a service is started by the Windows service manager. Three startup types are defined:

Automatic.	Service is started at boot time.
Manual.	Service is not started at boot time but can be manually started later.
Disabled.	Service is not started at boot time and is disabled by the service manager.

Interactive COBOL services are usually started automatically at boot time.

This option is the equivalent of using the "-S" from the command line version of ICSVCMGR.

```
icsvcmgr -S a|d|m icexec|icpermit|icnetd
```

Log On As

The log on information is used to define the Windows user account used when a service is started. Normally the default system account "LocalSystem" account is used to start the service. If necessary, a different account may be used.

This option is the equivalent of using the "-U" and "-P" from the command line versions of ICSVCMGR.

```
   icsvcmgr -U username -P password icexec|icpermit|icnetd
```

Dependencies

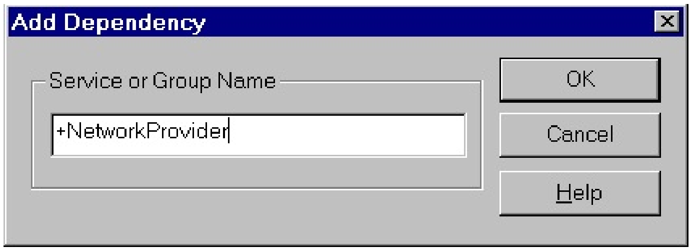
In certain instances, the order a service is started at boot time is important. The Dependencies setup allows a user to maintain a list of services (or service groups) that must be started before the specified service is started by the Windows service manager.

This option is the equivalent of using the "-D" from the command line versions of ICSVCMGR.

```
   icsvcmgr -D dependency  icexec|icpermit|icnetd
```

Adding Dependencies

To add a dependency, select the "Add" button to bring up the "Add Dependency" window. Enter the service name and hit the "OK" button to add the dependency. The service added should now be shown on the "Service Startup Options" dialog. Service groups are signified by prefixing the name with a "+" sign.



Service or Group Name: +NetworkProvider

Removing Dependencies

To remove a dependency, highlight the dependency to remove and hit the "Remove" button.

OK Button

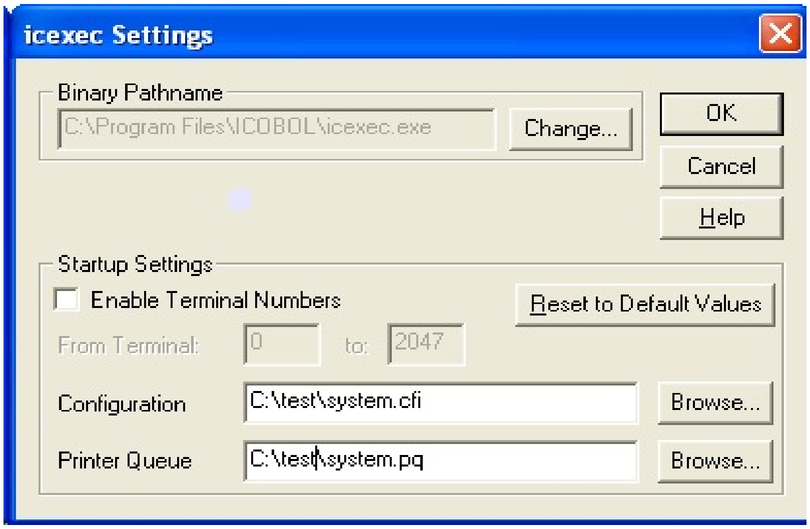
The "OK" button will save any changes made by the user to the settings and write them to the Windows service manager database. If successful, the user will be returned to the initial status display window.

Cancel Button

The "Cancel" button will allow the user to return to the initial status display window without saving any changes. When unsure of your actions on a particular window, you should use the "Cancel" button (or the escape key) to exit the window.

Selecting **Settings** will show the following for each of the indicated services:

For ICEXEC:



Binary Pathname: C:\Program Files\ICOBOL\icexec.exe

Startup Settings

☐ Enable Terminal Numbers

From Terminal: 0 to: 2047

Configuration: C:\test\system.cfi

Printer Queue: C:\test\system.pq

Binary Pathname

The “Binary Pathname” option displays the pathname to the executable file that is the specified service. This pathname is normally set during the installation and is not changed. It can be changed by selecting the “Change...” button and using the Open dialog window that comes up to select a new pathname.

This option is the equivalent of using the “-B” from the command line versions of ICSVCMGR.

```
icsvcmgr -B pathname icexec
```

Reset to Default Values

The “Reset to Default Values” button will set the icexec service settings to the defined default settings. These are the same settings that would be in effect if you had taken all the default selections during the installation of the software.

This option is the equivalent of using the following:

```
icexec -C %workingdir%\system.cfi -P %workingdir%\system.pq
```

Where
workingdir was defined at installation time

Terminal Numbers

The user may specify a range of terminal numbers to be used by the icexec service. To specify a range, the user must first select the “Enable Terminal Numbers” check box, then enter the proper range. The numbers entered here correspond to the console numbers @CONn to @CONm.

This option is the equivalent of using the “-T n:m” from the command line version of the service program.

```
icexec -T 0:1024
```

Configuration File

The user may specify either a directory name or a fully qualified filename for the configuration file. If just the directory name is entered, the service will attach the name “system.cfi” to the directory when opening the configuration file. The user may select the “Browse” button to bring up a browse window that can be used to select the proper directory path or file on the disk. If this field is left blank then the service will use it’s default value for the configuration file.

This option is the equivalent of using the “-C dir|file” from the command line version of the service program.

```
   icexec -C configfile
```

Printer Queue File

The user may specify either a directory name or a fully qualified filename for the printer queue file. If just the directory name is entered, the service will attach the name “system.pq” to the directory when opening the printer queue file. The user may select the “Browse” button to bring up a browse window that can be used to select the proper directory path or file on the disk. If this field is left blank then the service will use it’s default value for the printer queue file.

This option is the equivalent of using the “-P dir|file” from the command line version of the service program.

```
   icexec -P printerqueuefile
```

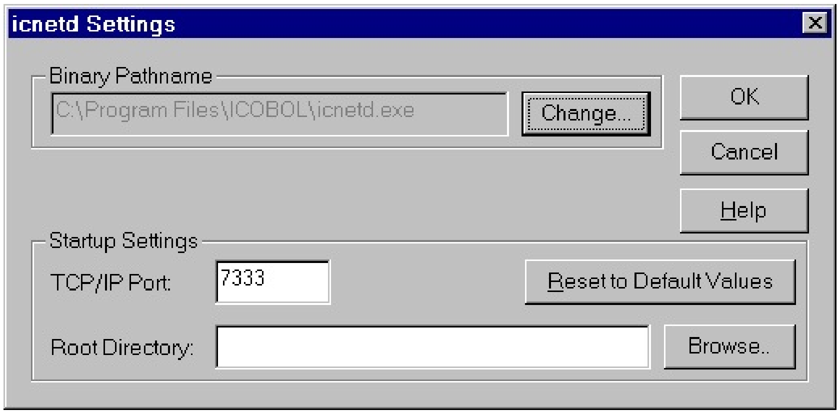
OK Button

The “OK” button will save any changes made by the user to the icexec settings and write them to the Windows service manager database. If successful, the user will be returned to the initial status display window.

Cancel Button

The “Cancel” button will allow the user to return to the initial status display window without saving any changes. When unsure of your actions on a particular window, you should use the “Cancel” button (or the escape key) to exit the window.

For ICNETD:



icnetd Settings

Binary Pathname: C:\Program Files\ICOBOL\icnetd.exe

Startup Settings

TCP/IP Port: 7333

Root Directory:

Binary Pathname

The “Binary Pathname” option displays the pathname to the executable file that is the specified service. This pathname is normally set during the installation and is not changed. It can be changed by selecting the “Change...” button and using the Open dialog window that comes up to select a new pathname.

This option is the equivalent of using the “-B” from the command line versions of ICSVCMGR.

```
icsvcmgr -B pathname icnetd
```

Reset to Default Values

The “Reset to Default Values” button will set the icnetd service settings to the defined default settings. These are the same settings that would be in effect if you had taken all the default selections during the installation of the software.

This option is the equivalent of using the following:

```
icnetd
```

TCP/IP Port

The user may specify the “listening” TCP/IP port for the icnetd service. This would be used in case there is a conflict with the default port normally used by the service (7333).

This option is the equivalent of using the following:

```
icnetd -M :port
```

Root Directory (for icios (thickclients))

The user may specify the pathname for the effective root directory of the service. When clients access files through icnetd’s i/o surrogate, icios, the file pathnames are relative to this root directory. The user may select the “Browse” button to bring up a browse window that can be used to select the proper path on the disk. If this field is left blank, then the service will use it’s default value for the root directory. This option is NOT used by the ThinClient servers.

This option is the equivalent of using the following:

```
    icnetd -R rootdir
```

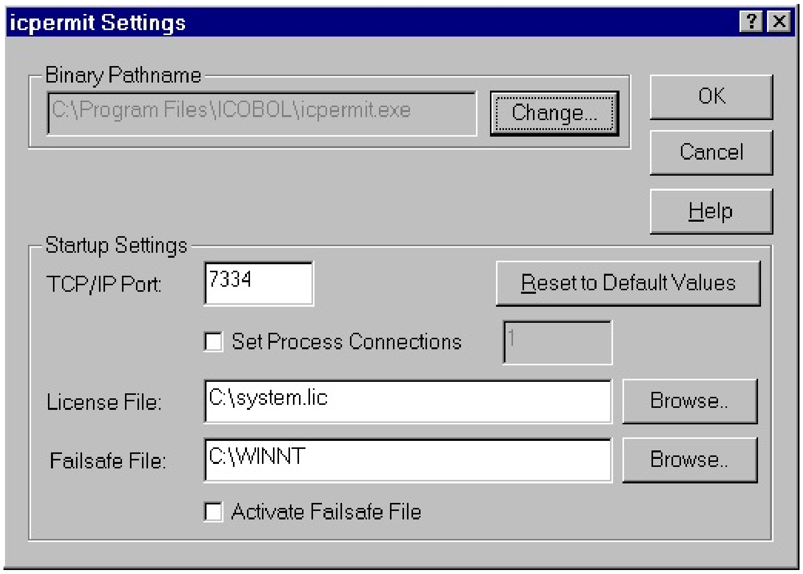
OK Button

The “OK” button will save any changes made by the user to the icnetd settings and write them to the Windows service manager database. If successful, the user will be returned to the initial status display window.

Cancel Button

The “Cancel” button will allow the user to return to the initial status display window without saving any changes. When unsure of your actions on a particular window, you should use the “Cancel” button (or the escape key) to exit the window.

For ICPERMIT:



icpermit Settings

Binary Pathname

C:\Program Files\ICOBOL\icpermit.exe

Startup Settings

TCP/IP Port: 7334

☐ Set Process Connections 1

License File: C:\system.lic

Failsafe File: C:\WINNT

☐ Activate Failsafe File

Binary Pathname

The “Binary Pathname” option displays the pathname to the executable file that is the specified service. This pathname is normally set during the installation and is not changed. It can be changed by selecting the “Change...” button and using the Open dialog window that comes up to select a new pathname.

This option is the equivalent of using the “-B” from the command line versions of ICSVCMGR.

```
    icsvcmgr -B pathname icpermit
```

Reset to Default Values

The “Reset to Default Values” button will set the icpermit service settings to the defined default settings. These are the same settings that would be in effect if you had taken all the default selections during the installation of the software.

This option is the equivalent of using the following:

```
    icpermit -L %windisk%\system.lic -F %windir%
```

Where
windisk is the root directory of the disk containing Windows (usually C:\)
windir is the main Windows directory (usually C:\WINDOWS)

TCP/IP Port

The user may specify the “listening” TCP/IP port for the icpermit service. This would be used in case there is a conflict with the default port normally used by the service (7334).

This option is the equivalent of using the following:

```
icpermit -M :port
```

Set Process Connections

The user may specify the number of process connections allowed for the icpermit service. This would be used in the case where the default count computed from the license file is insufficient.

This option is the equivalent of using the following:

```
icpermit -C processcount
```

Where
processcount is a positive integer less than 65,535

License File

The user may specify the license description file that the icpermit service will use when servicing license authorization requests. The user must specify a fully qualified filename for the license description file. The user may select the “Browse” button to bring up a browse window that can be used to select the proper file on the disk. If this field is left blank then the service will use it’s default value for the license description file.

This option is the equivalent of using the following:

```
icpermit -L licensefile
```

Failsafe File

The user may specify the directory name for the location of the failsafe file, system.fp. The user may select the “Browse” button to bring up a browse window that can be used to select the proper path on the disk. If this field is left blank, then the service will use it’s default value for the failsafe file.

This option is the equivalent of using the following:

```
icpermit -F failsafedir
```

Activate Failsafe File

The user may select this box to use the existing failsafe security file instead of the protection device.

This option is the equivalent of using the following:

```
   icpermit -f
```

OK Button

The “OK” button will save any changes made by the user to the icpermit settings and write them to the Windows service manager database. If successful, the user will be returned to the initial status display window.

Cancel Button

The “Cancel” button will allow the user to return to the initial status display window without saving any changes. When unsure of your actions on a particular window, you should use the “Cancel” button (or the escape key) to exit the window.

Help can always be used to provide additional information for an item.

IV. LICENSING (ICPERMIT)

A. Introduction

This chapter discusses the license description file and how to use ICPERMIT, the license server, to provide authorization information to any Interactive COBOL processes that requires licensing.

Interactive COBOL on Windows requires a valid license to be present for various programs to be executed. ICPERMIT is the program that reads a license description file (default system.lic) and authorizes the specified product(s) for the given user count. In most cases on Windows, either a parallel protection device or a USB protection device, with the indicated serial number, must also be present. If remote licensing is being used, ICPERMIT cannot be running on the local machine but must be available over a network from a central server. Two additional license types are offered: CPUID uses the serial number embedded in certain Intel processors and MAC uses the Media Access Control address of a network (usually ethernet) card.

ICPERMIT allows licenses to be shared over a TCP/IP-based network. With the CROSS option on a license, a license can be shared among machines of different types (Windows, UNIX-SCO, UNIX-HP, etc.). Without the CROSS option on a license, licenses can only be shared among machines of the same type (Windows, or the same flavor of **ICOBOL**/Unix.

B. License Description File

The license description file provides the unique information needed to license an individual site for the various combinations of product(s) and users. In most cases it must be used with a unique protection device (either parallel or USB (Windows) or serial (UNIX)). CPUID and MAC address licenses are also available.

If the license is provided on a PRODUCT ACTIVATION KEY CARD, a license description file can be created by using a standard text editor and entering the license (LIC) and authorization key (KEY) information provided on the card.

If the license has been provided as an e-mail attachment then copy/move the license file to the needed directory. If the license is provided on media then copy the license file to the appropriate disk.

PRODUCT ACTIVATION KEY CARD

The PRODUCT ACTIVATION KEY CARD contains the license and key information to authorize licensed products. Each KEY CARD shows one or more licensed products in the following fashion:

```
* For PARALLEL PROTECTION DEVICE 00000766

* ICOBOL Runtime License with BTRIEVE
LIC 01-PARALLEL-00000766-ICRUN-30-A-WIN-00033
KEY 2fgy87k1ms-8u1op1mn98-jhtrewsa-j8h6frd5s4-cxwzbnmk87-98up0kmngf
```

Where

Line one is a comment showing the license type (CPUID, MAC, PARALLEL, SERIAL, USB, etc.) with its Envyr serial number.

Line two is a comment showing the licensed product and any additional licensing option(s) for this license.

Line three (starting with LIC) is the text description of the license showing the ICPERMIT revision (01), license type (PARALLEL), serial number (00000766), licensed product (ICRUN), revision (30), options (A), operating system (WIN), and user count (00033).

Line four (final line) (starting with KEY) is an encrypted version of the license that can spill across multiple lines as needed.

Lines two through four can be repeated for each particular licensed product.

NOTE: The letters "i", "l", "q", and "v" are not used to prevent confusion with the number 1 or the letters "g" and "u".

Lines in the license description file that start with a `*' are comment entries only. Comment entries and blank lines are ignored.

If the license type is PARALLEL, SERIAL, or USB then this license must be paired up with the parallel, serial, or USB protection device with the same serial number to allow authorization. If the license type is CPUID, then this license will only run on a particular hardware configuration with its unique hardware identifier. If the license type is MAC, then this license will only run on a particular hardware configuration with its unique MAC network address as provided by the NIC card.

After building a license description file, run ICPERMIT with the Check switch (-c) to insure that the file was entered correctly.

Licensed products for each type of license currently include:

All runtimes (ICRUN, ICRUNCGI, etc.) use **ICOBOL** Runtime Licenses,
(Licensed options on Windows include: BTRIEVE, WATCH, ...)
Runtimes that use SP2 and/or FormPrint use **ICOBOL** SP2 Runtime Licenses,
Runtimes that use **ISQL** statements use **ICSQL** Runtime Licenses,
ICNETD I/O surrogates (icios) use **ICOBOL** Network Server Licenses,
ICNETD ThinClient (gui) surrogates (icthins) use both **ICOBOL** SP2 Runtime Licenses and **ICOBOL** Runtime Licenses,
ICNETD ThinClient surrogates (icrunrs) use both **ICOBOL** Remote Runtime Licenses and **ICOBOL** Runtime Licenses and possiblity **ICOBOL** SP2 Runtime licenses,
ICOBOL and ICIDE uses **ICOBOL** Development Licenses,
ICSP2 uses **ICOBOL** SP2 Development Licenses,
ICQPRW uses **ICOBOL** FormPrint Development Licenses,
User-Programs built using the User Library use **ICOBOL** Application Interface Licenses,
ODBC-enabled programs using the **ICOBOL** ODBC Driver use **ICOBOL** ODBC Driver Licenses.

C. Parallel or USB Protection

Parallel protection devices are supported for Windows platforms. Note that the appropriate hardware connection must also be available to connect the protection device.

If your license description file specifies "For PARALLEL PROTECTION DEVICE xxxxxxxx" then a parallel protection device must be installed. If your license description file specifies "For USB PROTECTION DEVICE xxxxxxxx" then a USB protection device must be installed.

Parallel protection devices must be placed on an IBM compatible parallel printer port that ICPERMIT can access through the standard parallel printer locations of 3BCh, 378h, or 278h. A parallel port on a USB port will NOT work. The parallel protection device causes no interference to an attached printer or any character transmitted to the printer. The parallel protection device provides a unique identification value to which an individual license description file is keyed.

USB protection devices must be installed in a powered USB port. They should be installed only **AFTER ICOBOL** has been installed with local licensing selected as the USB Rainbow driver files will be provided to the operating system at that time. Upon initial installation of the USB protection device, the operating system will detect the device and load the needed driver.

D. **CPUID or MAC Protection**

If your license description file specifies "For CPUID xxxx-xxxx-xxxx-xxxx” then the cpu in the machine must have an enabled cpu identification number that matches the given CPUID string.

If your license description file specifies "For MAC xxxx-xxxx-xxxx-xxxx” then a network card in the machine must have a MAC address with the given value.

ICINFO can be used to show CPUID and/or MAC addresses.

E. **ICPERMIT**

E.1. Syntax

The syntax for ICPERMIT is:

```
icpermit [-a[:aflag]|-A file|dir[:aflag] [-c] [-C count] [-f] [-F faildir]
    [-h|-?] [-i] [-L licedesc] [-M machine[:port]] [-N w]
    [-O a|c|d|e|i|l|m|p|r|s|t] [-q] [-s] [-t]
```

Where

-a[:*aflag*] or -A *file|dir*[:*aflag*] (Audit)

Enables auditing (default icpermit.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 (Check-only)

Check a license description file for validity and query any specified device, do not install. (Check can be used even when another ICPERMIT is running.)

-C *count* (Connections)

Specifies the maximum number of connections from the license server at any one time. The default is the maximum number of licenses times 3 plus 1. At times this number may be too small.

-f (Failsafe)

Use an existing failsafe security file instead of the protection device.

-F *faildir* (Failsafe location)

Specifies where to locate the failsafe security file. The failsafe file, system.fp, is located in this directory if given, otherwise the current directory is used.

-h|-? (Help)

Display help text.

-L *licedesc* (License description file)

Specifies the license description filename. The default is system.lic.

-M *machine*[:*port*]] (Machine)

Specifies the license server machine and optional TCP port. The default is the current machine (localhost) and port 7334. *Machine* can be an ip-address or a machine-name. No leading “\\” should be specified.

-N w (No)

Specifies a No option: Valid options are: w=no warnings.

-O a|c|d|e|i|l|m|p|r|s|t (Operation)

Specifies an operation to perform:

a (Amplify)	Amplify (turn on) tracing on the running icpermit
c (Check)	Check if an icpermit is running,
d (Disable)	Disable any new license connections,
e (Enable)	Re-enable new license connections,
i (Info)	Display license information,
l (Local)	Enable local operation,
m (Mute)	Mute (turn off) tracing,
p (Post)	The current connection/license information to be written to the log file,
r (Remote)	Enable remote operation (requires -M),
s (Start)	Start icpermit

t (Terminate)	Terminate icpermit

-q (Quiet)
Enables quiet operation.
-s (Service)
Enables start as a service.
-t (Tracing enabled)
Start with tracing enabled.

ICPERMIT 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, ICPERMIT is installed as a service using the ICSVCMGR utility. More on ICSVCMGR can be found on page 49. ICSVCMGR can be used to change the default command line if needed. ICPERMIT starts in the system directory (usually C:\WINDOWS).

Note: The Check Operation (-O c) and the Check mode (-c) do not require administrator privilege.

E.2. Description

ICPERMIT can run in one of several modes:

Check mode	(-c)	validates a license description file and device
License mode	(neither -c or -O)	provides license authorization
Operation mode	(-O)	performs an ICPERMIT operation

Check mode (-c)

Check mode instructs ICPERMIT to validate the given license description file and then query the protection device to confirm the license. ICPERMIT in check mode does not interfere with an already running ICPERMIT or whether local or remote licensing is in effect. Does not require administrator privilege.

Below is the output from a successful check.

```
> icpermit -c
icpermit Revision 3.60 (Windows)
Copyright (C) 1987-2007, Envyr Corporation.  All rights reserved.
License check was successful
icpermit is finished
>
```

If successful, an ERRORLEVEL of 0 is returned, otherwise a 1 is returned.

License mode

License mode instructs ICPERMIT to authorize processes that require Interactive COBOL licenses. ICPERMIT uses the information in the license description file to provide any required authorizations. Requires administrator privilege.

When ICPERMIT in license mode starts it takes the following steps:

1) Processes and validates any command line switches.
2) The audit file is opened. If it does not exist, it is created.
3) Opens a TCP/IP socket on the specified port to communicate with licensed products.
4) Reads the license description file and provides authorization based on that information.
   a) If the Failsafe switch (-f) was given the failsafe file is used to authorize users until a new protection device can be installed. This authorization is only temporary and will expire after the indicated period.

b) For a protection device, ICPERMIT interrogates the device as specified by the license to validate the license provided and authorize users.

c) If authorized by step b above, a new failsafe file (system.fp) is written to the appropriate location for subsequent use with the Failsafe switch (-f) if the protection device fails for some reason.

5) Waits to authorize users.

If at any point in the above steps ICPERMIT detects an error, it displays the appropriate message and terminates.

Multiple invocations of licensed products (the runtime, compiler, sp2 development, sp2 runtime, etc.) can be started on the master console and only one license count is used for each license type.

ICPERMIT in license mode can only be started as a service. The installation program sets up the ICPERMIT service to start in the working directory and with the license file information provided to the install script. The audit log defaults to icpermit.lg in the system directory (usually C:\WINDOWS). To check on services, from the **Start** menu, choose **Settings**, **Control Panel**. In the **Control Panel** folder, select **Services**. While in **Services**, ICPERMIT can be stopped and restarted.

When given a termination, if no licenses are in use, ICPERMIT will terminate immediately, otherwise it will pause for about 2 minutes before terminating. ICPERMIT does NOT terminate processes that are running.

Below is the output from a successful ICPERMIT in license mode on Windows 9x.

```
> icpermit
icpermit Revision 3.60 (Windows)
Copyright (C) 1987-2007, Envyr Corporation.  All rights reserved.
Nov-17-2006 05:38:51 icpermit (4293957149): Info: The license server is ready
```

For local connections, ICPERMIT continuously checks to insure that the process that requested a license is still running. If it no longer detects the process, a message will be generated, and the license will be removed from use to be re-used. This check is not done for remote licenses. If a remote user terminates abnormally, that license will remain in use until ICPERMIT receives a terminate request via the TCP/IP keepalive mechanism.

Operation mode (-O a|c|d|e|i|l|m|p|r|s|t)

Operation mode instructs ICPERMIT to perform some operation. All operation modes, except Check (-O c) require administrator privilege. In most cases, it communicates to a currently running ICPERMIT through the ICPERMIT TCP/IP port. Valid options include: (a) amplify (turn on) tracing, (c)heck if already running; (d)isable licenses; (e)nable licenses; give (i)nfo; (l) enable local licensing, (m) mute (turn off) tracing; (p) post ; (r) enable remote licensing, (s)start the icpermit service, and (t) terminate icpermit. If no ICPERMIT is running (for amplify, check, disable, enable, mute, post, or terminate) an "Unable to connect to the license server socket" will be given.

Operation amplify (-O a) sends an amplify command to an already running ICPERMIT to cause it to turn on tracing. If successful the following will be given:

```
Tracing of requests has been amplified.
```

Operation check (-O c) queries either the local or remote ICPERMIT to see if it is running. If successful, the pid and machine on which ICPERMIT is running is given. If successful, one of the following will be given:

```
ICPERMIR is running on pid 4294757653 of computer localhost, using port 7334.
```

Or

```
ICPERMIT is running on pid 150 of computer ntserver, using port 7334.
```

Operation disable (-O d) sends a disable command to ICPERMIT to disable providing any new license authorizations. Licenses that are currently authorized will continue to operate but no new licenses will be granted. This pauses the ICPERMIT service. If successful the following will be given:

```
The software licenses have been disabled.
```

Operation enable (-O e) sends an enable command to ICPERMIT to re-enable any licenses that had previously been disabled. This continues the ICPERMIT service. If successful the following will be given:

```
The software licenses have been re-enabled.
```

Operation info (-O i) queries ICPERMIT to provide current licensing information including licenses available, in use, and the most used. If successful the following type of message will be given:

```
License 1:  (enabled) ICOBOL Runtime Revision 3.xx and before
  Serial# 21000063  OS: Windows
  Authorized: 0 (max 0) of 129    References: 0
License 2:  (enabled) ICOBOL Development Revision 3.xx and before
  Serial# 21000063  OS: Windows
  Authorized: 0 (max 0) of 129    References: 0
```

Operation local (-O l) sets a registry entry that ICPERMIT will be running locally.

Operation mute (-O m) sends a mute command to ICPERMIT to cause it to turn off tracing. If successful the following will be given:

```
Tracing of requests has been muted.
```

Operation post (-O p) sends the post command to ICPERMIT to instruct it to output to its audit log the current license connection information.

Operation remote (-O r) sets a registry entry that ICPERMIT will be running on a remote machine. No local ICPERMIT will be permitted to run. Requires -M *machine* switch to specify the remote machine (option port).

Operation start (-O s) specifies that the ICPERMIT service should be started.

Operation terminate (-O t) specifies that the ICPERMIT service should be terminated.

With the use of TCP/IP, ICPERMIT can be used over the Internet by allowing the ICPERMIT port to be accepted (through a firewall) by the Internet server site. Then remote users can use licenses from a single "master" license server at any time the Internet is available from their machine.

Licenses can be shared over a network by machines of the same type by default. By having the optional “CROSS” license option added to a specific license, licenses can be shared over a network by machines of different flavors.

On a system shutdown, the system shutdown will send the appropriate message to ICPERMIT to cause it to start shutting down.

On shutdown, ICPERMIT will log its connection/license information to the log file.

When started with a valid protection device, ICPERMIT creates a failsafe security file, system.fp. The failsafe security file can be used to start ICPERMIT for up to 3 weeks after the failsafe security file was last written. (While ICPERMIT is running, the filesafe file is updated every 24 hours.) This insures that if the protection device fails, Interactive COBOL can still be used until you have time to get a replacement. From initial activation, the failsafe security file is good for 10 days. Within that timeframe, you should insure that your protection device has been replaced.

To use a failsafe file either use the Services Panel and start the ICPERMIT service manually and add the -f switch or use ICSVCMGR to change the ICPERMIT service to add the -f switch.

E.3. Errors

If at any point in starting an error is detected, the appropriate message is displayed and ICPERMIT terminates. All errors return a non-zero ERRORLEVEL.

Some example errors and how to fix them appear below.

If the license description file is bad in some way, the following will be generated:

```
Error:  Invalid or mis-matched license information: Line 4
Error:  No licenses are authorized.
```

Check that the license description file has been entered correctly on the given LIC and KEY lines associated with the Line number shown.

If no protection device can be found, the following will be generated:

```
Error:  Device is not available or does not exist:  Software protection device
Error:  No licenses are authorized
```

Attach the correct protection device or change the license file.

If an incorrect parallel device is found, the following will be generated:

```
Error:  License description does not match protection device information:  Software
    protection device
Error:  No licenses are authorized
```

Attach the correct protection device or change the license file.

If an ICPERMIT operation (-O) is done trying to communicate with an ICPERMIT that is NOT running the following will be generated:

```
Error:  Unable to connect to the license server socket.
```

Various errors above this error may indicate particular reasons why no connection could be made, including errors like: connection refused, remote computer is not available, etc.

On Windows, if you get an error like:

```
    Oct-28-2003 11:31:28.30 icpermit (4294847509):     Error:   Unexpected operating system
error (oserr=10022): Creating primary listening socket
    Error:   Could not create socket: Unexpected operating system error (oserr=10022)
```

Then TCP/IP is not installed and must be installed.

E.4. Notes

Licensed products do NOT know how they are licensed. They use the TCP/IP port created by an ICPERMIT to request authorizations.

Below are *errnum* values for errors that can be returned when communicating with the Sentinel driver for either parallel or USB protection devices.

1	Invalid function code
2	Invalid Packet
3	Unit not found
4	Access denied
5	Invalid Memory address
6	Invalid access code
7	Port is busy
8	Write not ready
9	No port found
10	Already zero
11	Driver open error
12	Driver not installed
13	IO Communications error
15	Packet too small
16	Invalid parameter
17	Memory access error
18	Version not supported
19	Os not supported
20	Query too long
30	Driver is busy

F. ICPERMIT Termination

To terminate ICPERMIT, either use the Terminate Operation (-O t) of ICPERMIT, use ICSVCMGR, or standard system services and select STOP .

When ICPERMIT terminates it will first prevent any new Interactive COBOL processes from starting.

Please be patient while ICPERMIT is terminating.

On a system shutdown, the system shutdown script/executable will send the appropriate signals to ICPERMIT to cause it to start shutting down.

On shutdown, ICPERMIT will log its connection/license information to the log file.

V. ICEDCFW

A. Introduction

ICEDCFW is the Windows GUI version of the configuration utility. ICEDCFW and/or ICCONFIG can be used to edit configuration files (.cfi), terminal description files (.tdi), and printer translation files (.pti). Each of these files is a .ini based text file with the appropriate sections and definitions..

ICEDCFW can only be run on the master console as it always assumes the master console type pcwindow.

NOTE: Versions of ICOBOL before 3.30 used configuration files, terminal description files, and printer translation files with the extensions of .cf, .td, and .pt files that were binary files. This format has been discontinued. The ICREVUP utility (documented in the readic.txt file) has been provided to up-rev these files to their .ini-based counterparts. If you need to be able to configure these older files then you must keep an older copy of the configuration utility around to provide that ability. The current configuration utilities will ONLY support the new .ini-based format.

Check boxes are used for Yes/No type responses. Checking the box is a Yes. An un-checked box is a No.

While editing a file, the OK selection at the bottom of the screen can also be used to save and close a file.

While editing a file, the CANCEL selection at the bottom of the screen can be used to close a file ignoring any changes.

While editing a file, the APPLY selection at the bottom of the screen can be used to save the file. It will be gray (inactive) until a change has been made.

If no Save is done, then any changes made are NOT saved. If you try to exit and no Save has been done since the configuration was last modified, you are prompted whether a Save should be done.

Multiple files of the same or different types can be opened and edited at the same time .

B. Startup

To start ICEDCFW the syntax is:

```
icedcfw [-a[:aflag]|-A file|dir[:aflag]] [-h|-?] [-q] [file]
```

Where

-a[:*aflag*] or -A *file*|*dir*[:*aflag*] (Audit)
    Enables auditing (default icedcfw.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.

file
    Specifies the actual filename to edit. If no extension is given, the `.cfi' extension is added. To edit a terminal description file or a printer translation file, the filename with the correct extension must be given.

If no argument is provided, ICEDCFW starts with the last file that was being edited.

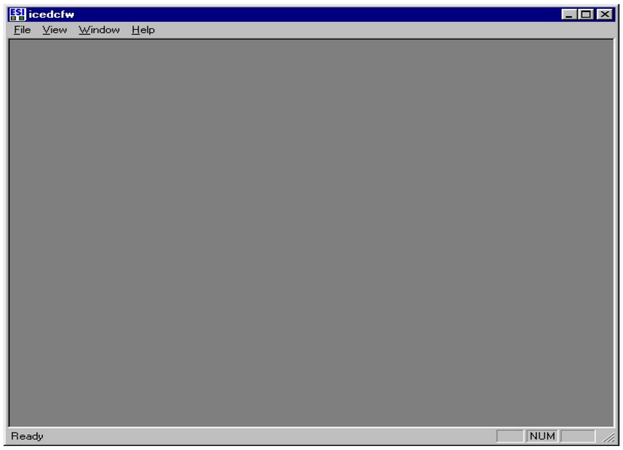
If the given configuration file exists and cannot be read at startup, an error is displayed and ICEDCFW terminates. The error should be fixed before re-running ICEDCFW. If the given configuration file does not exist at startup, a warning is displayed that the file was not found but ICEDCFW continues with no file.

For example the line:

```
icedcfw samplecf.cfi
```

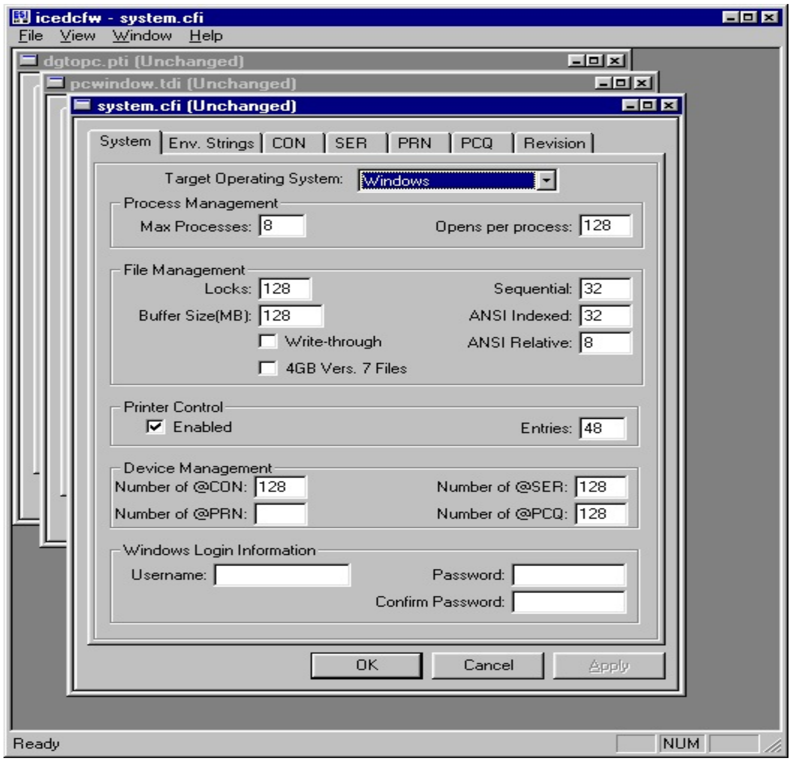
would retrieve the configuration file samplecf.cfi.

SCREEN 1 shows the MDI Window displayed when ICEDCFW starts with no file open. The View Tab allows Status Bar and Set Font for printing.



SCREEN 1. ICEDCFW no file

Multiple files of the same or different types can be opened and edited at the same time as shown in the following screen:



Target Operating System: Windows

Process Management
Max Processes: 8
Opens per process: 128

File Management
Locks: 128
Sequential: 32
Buffer Size(MB): 128
ANSI Indexed: 32
☐ Write-through
ANSI Relative: 8
☐ 4GB Vers. 7 Files

Printer Control
☑ Enabled
Entries: 48

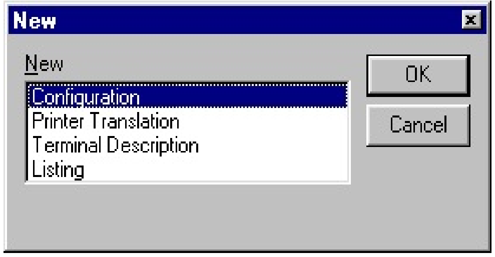
Device Management
Number of @CON: 128
Number of @SER: 128
Number of @PRN:
Number of @PCQ: 128

Windows Login Information
Username:
Password:
Confirm Password:

SCREEN 2. ICEDCFW multiple files

C. File tab

The **File** tab provides the ability to open a New file of any type. Selecting New will show a selection box like below:



that allows the type of file to be created.

It addition there are selections for Exit, Open which will bring up a Open dialog box, and once a file is open, there are selections for Save, Save As, Close, Create Listing. Print, and Print Preview. There is also a list of some of the most recently edited files that can be opened directly.

D. View tab

The **View** tab provides the ability to enable/disable the Status bar at the bottom and most importantly it provides a Font selection for the font to use with any listing. Only fixed fonts can be used and the default is a Courier.

E. Windows tab

The **Windows** tab provides the ability select how you want multiple files to be viewed. (Cascade, Tile, Arranged) along with selecting the current file.

F. Help tab

The **Help** tab gives an About selection that displays the revision of ICOBOL currently in use.

G. System Configuration (.cfi)

G.1. Overview

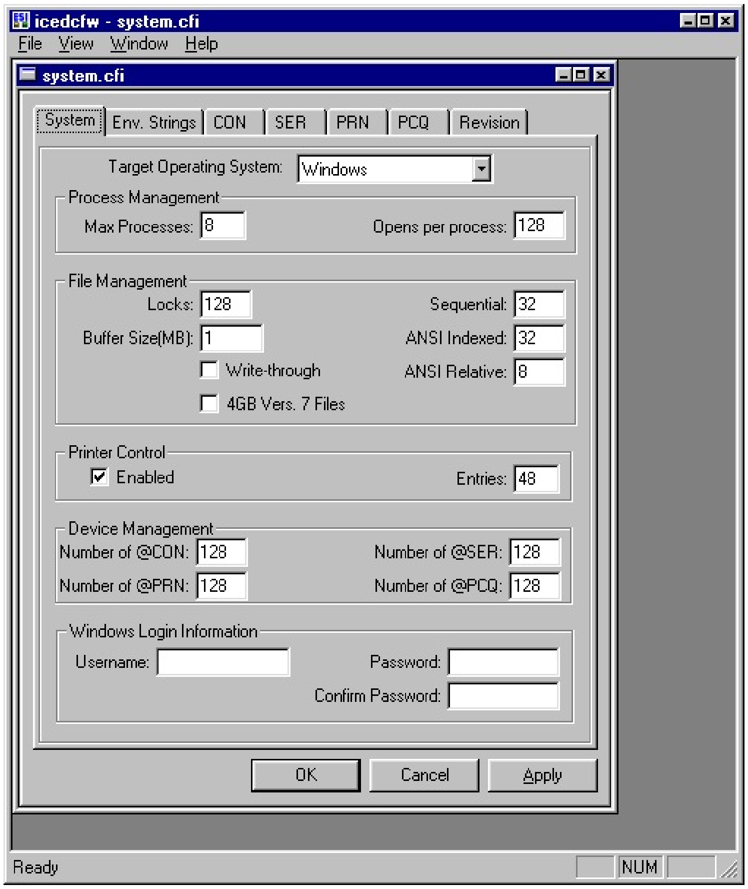
Initially opening a configuration file or selecting the **System** tab will show SCREEN 3. Each portion of the configuration file has a separate tab to allow that section to be configured.

G.2. **System** tab

These System Parameters define how the Interactive COBOL runtime allocates various resources. Choose the appropriate values by typing a valid value for each parameter. If there is an error, the menu is re-displayed with the cursor on the offending entry.

Terminal Status in Interactive COBOL will display the actual values that are current for some of these parameters along with the actual in use count for the files, record locks, and several other values.

System Information in Interactive COBOL will display the in use, maximum used, and configured values for many of these parameters during a particular invocation of the runtime.



Target Operating System: Windows

Process Management

Max Processes: 8   Opens per process: 128

File Management

Locks: 128   Sequential: 32

Buffer Size(MB): 1   ANSI Indexed: 32

□ Write-through   ANSI Relative: 8

□ 4GB Vers. 7 Files

Printer Control

☑ Enabled   Entries: 48

Device Management

Number of @CON: 128   Number of @SER: 128

Number of @PRN: 128   Number of @PCQ: 128

Windows Login Information

Username: Password:

Confirm Password:

SCREEN 3. ICEDCFW SYSTEM PARAMETERS

Target Operating System will show the target operating system for which this configuration is being setup for. For a new configuration file it defaults to "Windows".

Under Process Management.

Processes, is the maximum number of Interactive COBOL processes including ICEXEC, runtimes, and ICNETD surrogates that can execute simultaneously when ICEXEC is running. Valid entries are from 1 to 9999; 8 is the default.

OPENs is the maximum number of simultaneous file opens that any Interactive COBOL runtime invocation will allow to be opened (i.e., OPEN FD's in a COBOL program). The number of available handles is decreased every time a file is logically opened by any program in the run unit. Valid entries are from 100 to 2048; 128 is the default.

Under File Management.

These apply to ICEXEC as a whole .

Locks is the number of simultaneous record locks allowed . Valid entries are from 0 to 32767; 128 is the default.

SEQUENTIAL is the number of unique sequential files that can be simultaneously opened . Valid entries are from 0 to 4096; 32 is the default.

Buffer Size (MB) is the amount of memory, in MegaBytes, that will be allocated for buffers. A certain minimum buffer size must be provided that is calculated as "maximum number of processes * 2 * 2KB". Valid entries are from 1 to 1024; 1 is the default. With no ICEXEC, more than 1MB is generally not needed since most files are opened in network mode such that all writes must flush the data to the network immediately. If large files are opened exclusively for modification then increasing this parameter may be useful. With ICEXEC, this number should be such that each runtime has at least 30-40KB.

ANSI INDEXED is the number of unique indexed files that can be simultaneously opened . Valid entries are from 0 to 8192; 32 is the default.

Write-through set to Yes instructs the Interactive COBOL runtime to write all modified pages to the operating system on any operation that modifies data that resides on the disk. Valid entries are Yes or No; No is the default.

ANSI RELATIVE is the number of unique relative files that can be simultaneously opened. Valid entries are from 0 to 4096; 8 is the default.

Enable 4GB ICISAM version 7 files specifies whether to create version 7 ICISAM files with the ability to have 4GB files. If not enabled (No), only 2GB files are allowed. Valid entries are yes and no; No is the default.

<u>Under Printer Control.</u>

Enabled allows for the Printer Control Utility to be enabled (Yes) or disabled (No). If set to No, files normally placed in the printer control file (system.pq) are not placed there and the IC_PRINT_STAT builtin will return an error. Valid entries are Yes or No; Yes is the default.

Entries is the maximum number of entries allowed in the printer control file at once. Once this number is reached, all new files will get a File Status 99 when a new file is being OPEN'ed which would create a new entry in the printer control file. Valid entries are from 48 to 1024 entries; 48 is the default.

<u>Under Device Management</u>

Number of @CON is the number of consoles to allowed to be configured in this configuration. Valid entries are from 1 to 2048. 128 is the default

Number of @SER is the number of serials to allowed to be configured in this configuration. Valid entries are from 0 to 2048. 128 is the default

Number of @PRN is the number of printers to allowed to be configured in this configuration. Valid entries are from 0 to 2048. 128 is the default

Number of @PCQ is the number of printer queues to allowed to be configured in this configuration. Valid entries are from 0 to 2048. 128 is the default

<u>Logon Information</u>

This username/password will be used as the default profile for serial line logons via ICEXEC and the username prompt has been disabled.

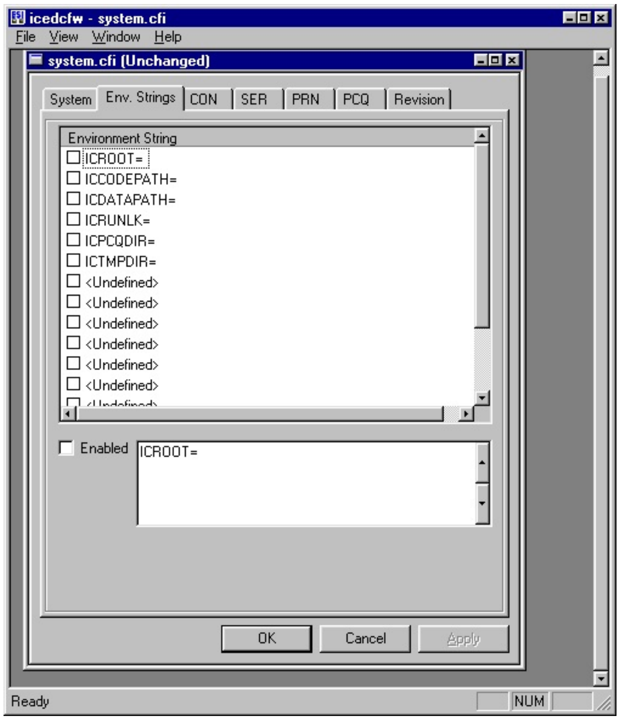
Username specifies the default username.

Password specifies the default password for the above username.

Confirm Password confirms the above password.

G.3. **Environment Strings** tab

The **Env. Strings** tab shows SCREEN 4 which allows common environment entries to be specified. Entries defined in this section will be available to the runtime for all consoles. These common environment entries can be overridden by setting the same entry in the actual environment provided by the operating system.



Environment String

- □ ICROOT=
- □ ICCODEPATH=
- □ ICDATAPATH=
- □ ICRUNLK=
- □ ICPCQDIR=
- □ ICTMPDIR=
- □ <Undefined>
- □ <Undefined>
- □ <Undefined>
- □ <Undefined>
- □ <Undefined>
- □ <Undefined>
- □ <Undefined>

□ Enabled ICROOT=

SCREEN 4. ICEDCFW ENVIRONMENT STRINGS

More on environment entries can be found on page 135, 153.

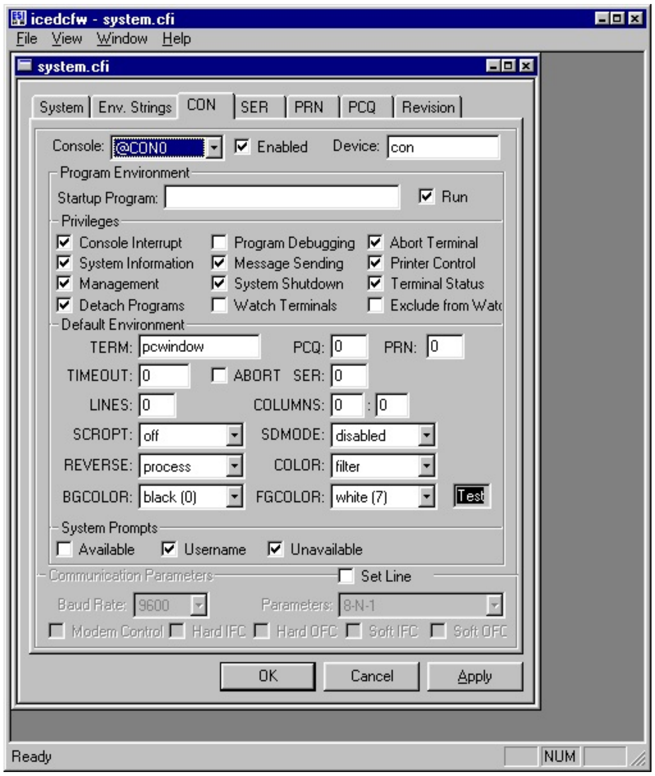
Up to 15 default environment strings can be stored in the configuration file.

An environment string entry allows up to 255 characters.

G.4. **Consoles and Programs** tab

Selecting the **CON** tab allows the configuration information for all the logical console lines (@CON0 and up) in the Interactive COBOL system to be defined as shown in SCREEN 5.

This menu allows a console to be enabled or disabled and an actual hardware device to be assigned to it.



Console: @CON0 ☑ Enabled Device: con

Program Environment
Startup Program: ☑ Run

Privileges
☑ Console Interrupt ☐ Program Debugging ☑ Abort Terminal
☑ System Information ☑ Message Sending ☑ Printer Control
☑ Management ☑ System Shutdown ☑ Terminal Status
☑ Detach Programs ☐ Watch Terminals ☐ Exclude from Watc

Default Environment
TERM: pcwindow PCQ: 0 PRN: 0
TIMEOUT: 0 ☐ ABORT SER: 0
LINES: 0 COLUMNS: 0 : 0
SCROPT: off SDMODE: disabled
REVERSE: process COLOR: filter
BGCOLOR: black (0) FGCOLOR: white (7) Test

System Prompts
☐ Available ☑ Username ☑ Unavailable

Communication Parameters ☐ Set Line
Baud Rate: 9600 Parameters: 8-N-1
☐ Modem Control ☐ Hard IFC ☐ Hard OFC ☐ Soft IFC ☐ Soft OFC

SCREEN 5. ICEDCFW CONSOLE DEVICE

Where:

The first entry *Console* is the console to be configured. Valid numbers are 0 to the highest supported console.

Enabled set to Yes allows this console to execute Interactive COBOL.

Device, can be any of the following:

1) “con” allows the console to be available from the Master Console session. (ICTERM=pcwindow). In most cases, many consoles should be configured this way to allow multiple invocations.

2) “icthins” allows the console to be available for ThinClient (gui) programs only.

3) “icrunrs” allows the console to be available for ThinClient programs only.

4) “nul” allows the console to be available for detached programs or for ThinClient (gui) programs.

5) “cgi”, allows the console to be available for cgi programs using icruncgi. You should have enough “cgi” consoles to support the maximum simultaneous number of cgi connections you expect to have.

6) (blank) allows the console to be available for telnet/rlogin/ThinClient sessions.

7) any valid serial device. If a COM port above 9 is specified it must be specified as “\\.\COM10", etc. Only

one instance of a serial device can be specified for a console that will allow COBOL programs to be run.

8) a valid *"machine-name*" or *"ip-address*" allows the console to be available for a ThinClient connection from that particular remote machine.

The *machine-name* and/or *ip-address* allows for a specific machine to have a specific console number or numbers.

Multiple consoles can be set to the same device (with the exception of a serial device) to allow multiple runtime sessions to be started on that device. Within the same device name, consoles are used on a first-come first-served basis. A particular console can be selected by using the Terminal number switch (-T *n*).

Run specifies whether this console allows a COBOL program to be run on it. Valid selections are Yes or No. The default is Yes for all consoles. Console lines with the Run program option enabled are called program lines. If the Run Program option is set to No this console device is treated like an @SER serial device and all the remaining settings in this menu are ignored.

Startup program is the initial COBOL program to run when ICRUN initializes this console. Valid selections are no entry or any valid COBOL program with up to 30 characters. The default is no entry causing the COBOL program LOGON to be run.

Privileges

These options provide for individual control over many system management functions about whether a COBOL program on this console has access to the particular feature. Valid selections are Yes or No. Console 0 defaults to all these settings enabled. All other consoles default to have Abort, Printer control management, and Shutdown disabled.

Console interrupt determines whether to allow the user to abort the currently running COBOL program with the Ctrl-C sequence and to halt the program with the Ctrl-Break sequence. If set to No, the Ctrl-C character is passed on up to the program as data and the Ctrl-Break sequence is ignored. When set to Yes, Ctrl-C is trapped by the runtime system and generates an abort to the currently running program, Ctrl-Break is trapped by the runtime system and generates a halt to the currently running program.

Program Debugging determines whether to allow Interactive COBOL debugging to be run on this line.

Abort Terminal determines whether this feature is allowed for COBOL programs running on this console.

System Information determines whether this feature is allowed for COBOL programs running on this console.

Message Sending determines whether this feature is allowed for COBOL programs running on this console.

Printer Control determines whether this feature is allowed for COBOL programs running on this console.

Management determines whether this console is allowed to perform all operations on the printer control file while in the Printer Control utility. If this privilege is granted, this console may perform any operation on any file while in the printer control utility provided the user has access to the file from the operating system.

System Shutdown is ignored.

Terminal Status determines whether this feature is allowed for COBOL programs running on this console.

Detach Programs determines whether to allow this program to detach programs and run Host programs.

Watch other terminals determines whether the Watch facility should be allowed for COBOL programs running on this console.

Exclude from Watch determines whether this program can be Watched or Controller by another.

Default Environment

This section specifies terminal specific information for a particular console line. There are corresponding environment entries available that override these entries in present.

ICTERM specifies the terminal description entry to be used for this console. The default is pcwindow for all consoles. Valid ICTERM selections are valid terminal description entries with corresponding .TDI files. For an enabled console, this menu cannot be exited without some selection specified for ICTERM.

PCQ sets the generic printer control queue (@PCQ) to @PCQn based on the entered number. Valid selections are 0 through 2047, the default is 0. If set to an invalid queue, an error will occur on the OPEN.

PRN sets the generic printer device (@PRN) to @PRNn based on the entered number. Valid selections are 0 through 2047, the default is 0. If set to an invalid printer, an error will occur on the OPEN.

ICTIMEOUT sets a default global timeout value in seconds for all ACCEPTs and STOP literals on this console. If no key has been pressed in the specified time interval, the ACCEPT returns with the ESCAPE code set to 99. Valid selections are 0 through 6300, the default is 0 meaning no timeout, i.e., wait forever.

ICABORT specifies whether to abort the console (i.e., log it off) if an ACCEPT times out due to the global timeout setting (ICTIMEOUT). Valid selections are off or on, the default is off.

SER sets the generic serial device (@SER) to @SERn based on the entered number. Valid selections are 0 through 2047, the default is 0. If set to an invalid serial device, an error will occur on the OPEN.

ICLINES and *ICCOLUMNS* set the number of lines and columns (including min and max) that will be allowed on this console. Valid selections are 0 through 255. The default of 0 says use the values specified in the terminal description entry. These values indicate to Interactive COBOL where the screen wraps (ICCOLUMNS) and scrolls (ICLINES). If set incorrectly, screens may not display properly.

ICSCROPT specifies how the SCREEN OPTIMIZER is to perform. Valid selections are off, partial, full, and mute. The default is full. All enabled consoles have at least one screen area reserved. For a 24x80 column screen, a single screen image consumes about 8KB.

Off says to transmit character codes as they are written by the program.

Partial enables the SCREEN OPTIMIZER to use a simple method of reducing the amount of characters sent to the terminal.

Full enables the SCREEN OPTIMIZER to keep two images of the current screen in memory and provides a complex method of comparing the data in the two images to reduce the amount of characters sent to the terminal to only those characters that would change the screen display at the end of an operation. In this mode two additional memory areas are allocated to hold this image.

Mute forces Interactive COBOL to not send any implied codes to the terminal either at startup or termination. Only when executing a program instruction are codes sent to the terminal.

Ctrl-U from the keyboard while in an input from the console, can be used to refresh the screen.

Usually partial and/or full will provide improved screen performance. Full is preferred with the debugger.

ICSDMODE specifies whether to enable the SCREEN HANDLER and if so in what mode. Valid selections are disabled, underline (0), reverse (1), and drawlines (2). The default is disabled. Drawlines uses the characters for line drawing specified in the terminal description file for the particular terminal.

ICREVERSE specifies how to interpret reverse codes from a COBOL program. Valid selections are filter, ignore, and process. The default is process. Reverse codes are Ctrl-B and Ctrl-V along with the two-byte sequences <036>D and <036>E.

Filter says to watch for reverse codes from the program and to NOT send them to the terminal, since it does not support reverse.

Ignore says that the user wants total control of the screen and may be sending binary reverse data to the screen and that Interactive COBOL should ignore all reverse codes (i.e., do not look for reverse codes). If running in this mode, the SCREEN OPTIMIZER cannot correctly repaint a user's screen that includes reverse codes.

Process says to interpret reverse codes from the program and send the appropriate sequences to the terminal.

ICCOLOR specifies how to interpret color codes from a COBOL program. Valid selections are filter, ignore, and process. The default is filter.

Filter says to watch for color codes from the program and to NOT send them to the terminal, since it does not support color.

Ignore says that the user wants total control of the screen and may be sending binary color data to the screen and that Interactive COBOL should ignore all color codes (i.e., do not look for color codes). If running in this mode, the SCREEN OPTIMIZER cannot correctly repaint a user's screen that includes colors.

Process says to interpret color codes from the program and send the appropriate sequences to the terminal. When set to Process the initial background and foreground colors are set by Interactive COBOL at startup.

ICBGCOLOR sets the initial background color to the indicated value when running with ICCOLOR set to Process. Valid selections are black (0), blue (1), green (2), cyan (3), red (4), magenta (5), brown (6), and white (7). The default is black (0).

ICFGCOLOR sets the initial foreground color to the indicated value when running with Color support (ICCOLOR) set to Process. Valid selections are black (0), blue (1), green (2), cyan (3), red (4), magenta (5), brown (6), and white (7). The default is white (7).

The Test box is used to show the current ICBGCOLOR and ICFGCOLOR settings output.

System Prompts - Messages

These selections are ONLY used by ICEXEC when supporting serial consoles. These selections allow the Windows product to more closely match the **ICOBOL** for MS-DOS multi-user product.

Available specifies whether to prompt with a “System is ready” message. The default is no.

Username specifies whether to prompt for username/password. The default is yes. If disabled a username/password must be found somewhere to allow ICEXEC to log the user on. This is done with the default username/password pair stored in the System Strings section of the configuration.

Unavailable specifies whether to show a “System is unavailable” message when ICEXEC shutdowns.

Communication Parameters

***S**et Line* instructs the runtime whether to use the following port device settings. If set to No, the following settings are not used and an open of the device will use whatever system defaults there are. If set to Yes, the runtime will attempt to set the device settings to the given values.

Baud Rate and *Parameters* set the appropriate selection to the given value. These values must match those of the attached device or else undefined results will occur. Generally the baud rate should be set to the fastest possible setting that the runtime, port, wiring, and output device can support.

Modem Control tells Interactive COBOL how to handle modem control signal Data Carrier Detect (DCD) for a particular device. If Mdm Ctl is set to No, DCD is ignored.

Modem Control set to Yes for program lines (i.e., an @CON logical device is directed to them with the Run Program option set to Yes) causes the initial banner message to be held until Data Carrier Detect (DCD) is asserted, at which time the message will be displayed and the remote user can logon. If the line was set to auto run, the initial program will start when DCD is asserted. If DCD is lowered while a program is active, the program is aborted and the terminal is logged off.

Modem Control set to Yes for serial lines will cause an OPEN to wait until DCD is asserted before returning. If DCD is lowered after the OPEN, the next I/O operation to that line will receive an error and the appropriate action will be taken.

Hard OFC (HOFC) tells Interactive COBOL how to handle the Clear To Send (CTS) signal. HOFC set to No says to ignore CTS. HOFC set to Yes says to monitor CTS to decide if the attached device is ready to receive output. If not, it waits until CTS is asserted. Data Set Ready (DSR) must also be asserted for characters to be sent.

Hard IFS (HIFC) tells Interactive COBOL how to handle the Request to send (RTS) signal. HIFC set to No says to leave RTS high after an open. HIFC set to Yes says to raise and lower RTS to instruct the attached device if it can send data.

HOFC and HIFC should both be enabled when dealing with high-speed modems as SOFC and SIFC are not reliable in these cases.

Soft OFC (SOFC) tells Interactive COBOL whether to watch for a Ctrl-S (XOFF) coming from the output device to tell Interactive COBOL to stop sending characters and then wait for a Ctrl-Q (XON) to continue transmitting. SOFC set to No says to ignore XON and XOFF codes while Yes says to watch for an XOFF and stop transmitting until an XON is seen.

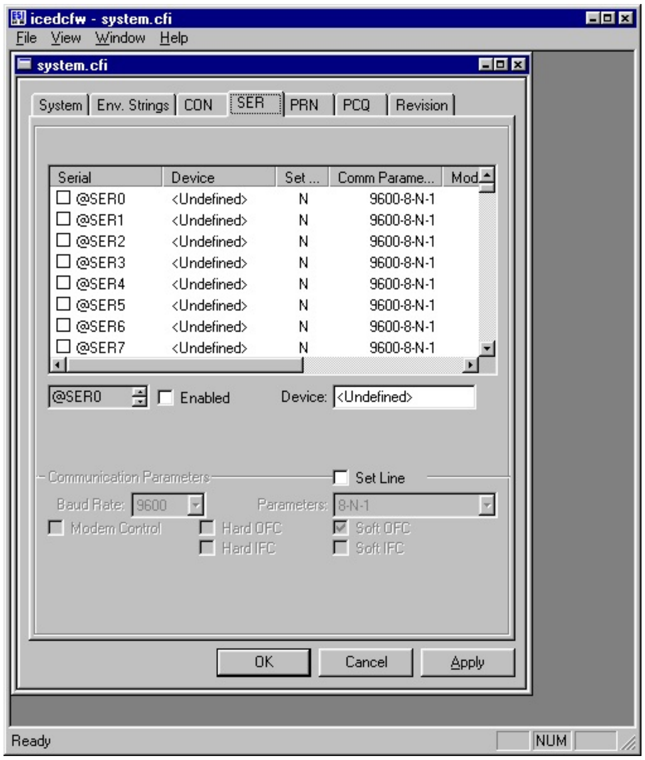
Soft IFC (SIFC) tells Interactive COBOL whether to send Ctrl-S (XOFF) and Ctrl-Q (XON) to the input device to tell it to stop transmitting characters and start transmitting characters respectively. This should NEVER be used for a DG terminal or emulator because the Ctrl-S and Ctrl-Q codes are valid attribute control codes and are NOT recognized as XOFF and XON.

Console lines that are not enabled are ignored. Console lines for which the hardware is not present give a warning at startup and an error on OPEN.

G.5. **Serial Lines** tab

Selecting the **SER** tab allows the configuration information for all the logical serial lines (@SER0-@SER2047) to be defined as shown in SCREEN 6. A serial device differs from a console device in that it can only be used for I/O. Console devices can be used to run programs or for I/O.

This menu defines logical serial devices to point to a particular hardware device.



Serial	Device	Set ...	Comm Parame...	Mod
□ @SER0	<Undefined>	N	9600-8-N-1	
□ @SER1	<Undefined>	N	9600-8-N-1	
□ @SER2	<Undefined>	N	9600-8-N-1	
□ @SER3	<Undefined>	N	9600-8-N-1	
□ @SER4	<Undefined>	N	9600-8-N-1	
□ @SER5	<Undefined>	N	9600-8-N-1	
□ @SER6	<Undefined>	N	9600-8-N-1	
□ @SER7	<Undefined>	N	9600-8-N-1	

@SER0 □ Enabled Device: <Undefined>

Communication Parameters □ Set Line

Baud Rate: 9600 Parameters: 8-N-1

□ Modem Control □ Hard OFC ☑ Soft OFC

□ Hard IFC □ Soft IFC

SCREEN 6. ICEDCFW SERIAL DEVICE

Where:

Enabled set to Yes allows this serial device to be used.

Device can be any of the hardware character devices except for parallel printers that support both input and output operations. Two or more serial devices (@SERn) can point to the same hardware device.

***S**et Line* instructs the runtime whether to use the following port device settings. If set to No, the default the following settings are not used and an open of the device will use whatever system defaults there are. If set to Yes, the runtime will attempt to set the device settings to the given values.

Baud Rate and *Parameters* set the appropriate selection to the given value. These values must match those of the attached device or else undefined results will occur. Generally the baud rate should be set to the fastest possible setting that the runtime, port, wiring, and attached device can support.

Modem Control (modem control) tells Interactive COBOL how to handle modem control signal Data Carrier Detect (DCD) for a particular device. If Modem Control is set to No, DCD is ignored.

Modem Control set to Yes for serial lines will cause an OPEN to wait until DCD is asserted before returning. If DCD is lowered after the OPEN, the next I/O operation to that line will receive an error and the appropriate action will be taken.

Hard OFC (HOFC) tells Interactive COBOL how to handle the Clear To Send (CTS) signal. HOFC set to No says to ignore CTS. HOFC set to Yes says to monitor CTS to decide if the attached device is ready to receive output. If not, it waits until CTS is asserted. Data Set Ready (DSR) must also be asserted for characters to be sent.

Hard IFC (HIFC) tells Interactive COBOL how to handle the Request to send (RTS) signal. HIFC set to No says to leave RTS high after an open. HIFC set to Yes says to raise and lower RTS to instruct the attached device if it can send data.

HOFC and HIFC should both be enabled when dealing with high-speed modems as SOFC and SIFC are not reliable in these cases.

Soft OFC (SOFC) tells Interactive COBOL whether to watch for a Ctrl-S (XOFF) coming from the output device to tell Interactive COBOL to stop sending characters and then wait for a Ctrl-Q (XON) to continue transmitting. SOFC set to No says to ignore XON and XOFF codes while Yes says to watch for an XOFF and stop transmitting until an XON is seen.

Soft IFC (SIFC) tells Interactive COBOL whether to send Ctrl-S (XOFF) and Ctrl-Q (XON) to the input device to tell it to stop transmitting characters and start transmitting characters respectively. This should NEVER be used for a DG terminal or emulator because the Ctrl-S and Ctrl-Q codes are valid attribute control codes and are NOT recognized as XOFF and XON.

A hardware device configured as a console device enabled to run programs cannot also be used as a serial device.

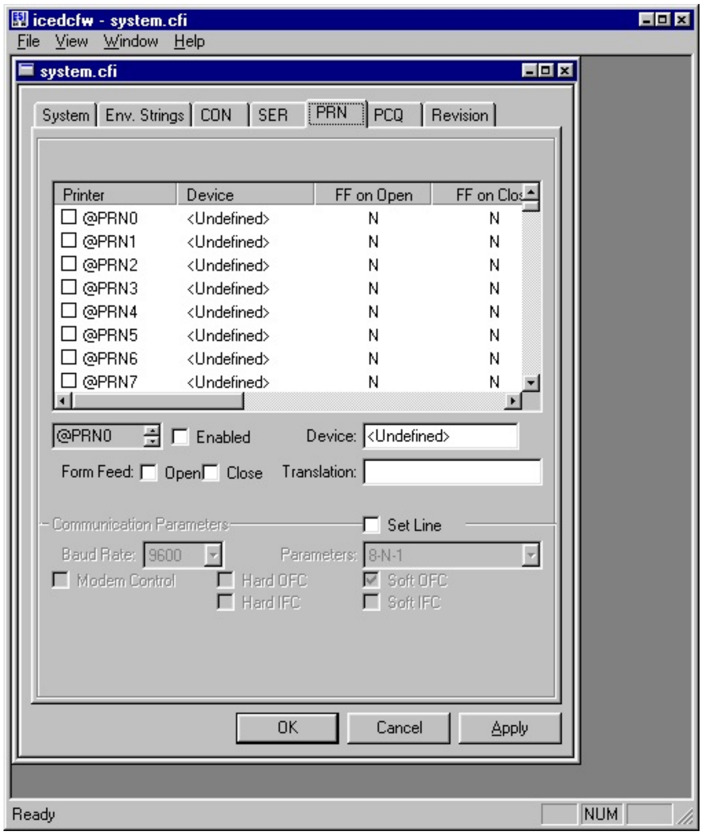
Serial devices that are not enabled or for which the hardware device is not available are ignored and give an error on OPEN.

When opening serial devices and Set Line is No, the runtime uses the last setting for that serial device to set up the default parameters (baud, parity, number of data bits, etc.). The MODE command can be used to perform these settings if needed. This MODE will be remembered until another setting is stored. In addition, extended open options can be used to set the needed values.

G.6. **Printers** tab

Selecting the **PRN** tab allows the configuration information for all the logical printers (@PRN0-@PRN2047) to be defined as shown in SCREEN 7.

This menu defines certain characteristics for each logical printer to be specified.



Printer	Device	FF on Open	FF on Clos
□ @PRN0	<Undefined>	N	N
□ @PRN1	<Undefined>	N	N
□ @PRN2	<Undefined>	N	N
□ @PRN3	<Undefined>	N	N
□ @PRN4	<Undefined>	N	N
□ @PRN5	<Undefined>	N	N
□ @PRN6	<Undefined>	N	N
□ @PRN7	<Undefined>	N	N

@PRN0 □ Enabled Device: <Undefined>

Form Feed: □ Open □ Close Translation:

Communication Parameters □ Set Line

Baud Rate: 9600 Parameters: 8-N-1

□ Modem Control □ Hard OFC ☑ Soft OFC

□ Hard IFC □ Soft IFC

SCREEN 7. ICEDCFW PRINTER DEVICE

Where:

Enabled set to Yes allows this printer device to be used.

Device directs the Interactive COBOL runtime where to send the print data for a particular logical printer. Possible selections are any of the hardware character devices (COMx, MDMx, LPTx, etc)that are not in use as terminal lines or blank for None. Two or more printers (@PRNn) can point to the same hardware device. This would be useful if different printer options are needed.

FF on OPEN and *FF on CLOSE* instructs the runtime whether to send a Form-Feed to the printer when the appropriate statement is executed on a particular printer.

Translation specifies a printer translation file to be used when printing. If nothing is specified, each character is printed as given. Printer translation files are opened and read when the runtime system is started.

***S**et Line* instructs the runtime whether to use the following port device settings. If set to No, the following settings are not used and an open of the device will use whatever system defaults there are. If set to Yes, the runtime will attempt to set the device settings to the given values. These device settings are ignored for parallel ports.

Baud Rate and *Parameters* set the appropriate selection to the given value. These values must match those of the attached device or else undefined results will occur. Generally the baud rate should be set to the fastest possible setting that the runtime, port, wiring, and output device can support.

Modem Control tells Interactive COBOL how to handle modem control signal Data Carrier Detect (DCD) for a particular device. If Modem Control is set to No, DCD is ignored.

Modem Control set to Yes for serial lines will cause an OPEN to wait until DCD is asserted before returning. If DCD is lowered after the OPEN, the next I/O operation to that line will receive an error and the appropriate action will be taken.

Hard OFC (HOFC) tells Interactive COBOL how to handle the Clear To Send (CTS) signal. HOFC set to No says to ignore CTS. HOFC set to Yes says to monitor CTS to decide if the attached device is ready to receive output. If not, it waits until CTS is asserted. Data Set Ready (DSR) must also be asserted for characters to be sent.

Hard IFC (HIFC) tells Interactive COBOL how to handle the Request to send (RTS) signal. HIFC set to No says to leave RTS high after an open. HIFC set to Yes says to raise and lower RTS to instruct the attached device if it can send data.

HOFC and HIFC should both be enabled when dealing with high-speed modems as SOFC and SIFC are not reliable in these cases.

Soft OFC (SOFC) tells Interactive COBOL whether to watch for a Ctrl-S (XOFF) coming from the output device to tell Interactive COBOL to stop sending characters and then wait for a Ctrl-Q (XON) to continue transmitting. SOFC set to No says to ignore XON and XOFF codes while Yes says to watch for an XOFF and stop transmitting until an XON is seen.

Soft IFC (SIFC) tells Interactive COBOL whether to send Ctrl-S (XOFF) and Ctrl-Q (XON) to the input device to tell it to stop transmitting characters and start transmitting characters respectively. This should NEVER be used for a DG terminal or emulator because the Ctrl-S and Ctrl-Q codes are valid attribute control codes and are NOT recognized as XOFF and XON.

A hardware device configured as a console device enabled to run programs cannot also be used as a printer device.

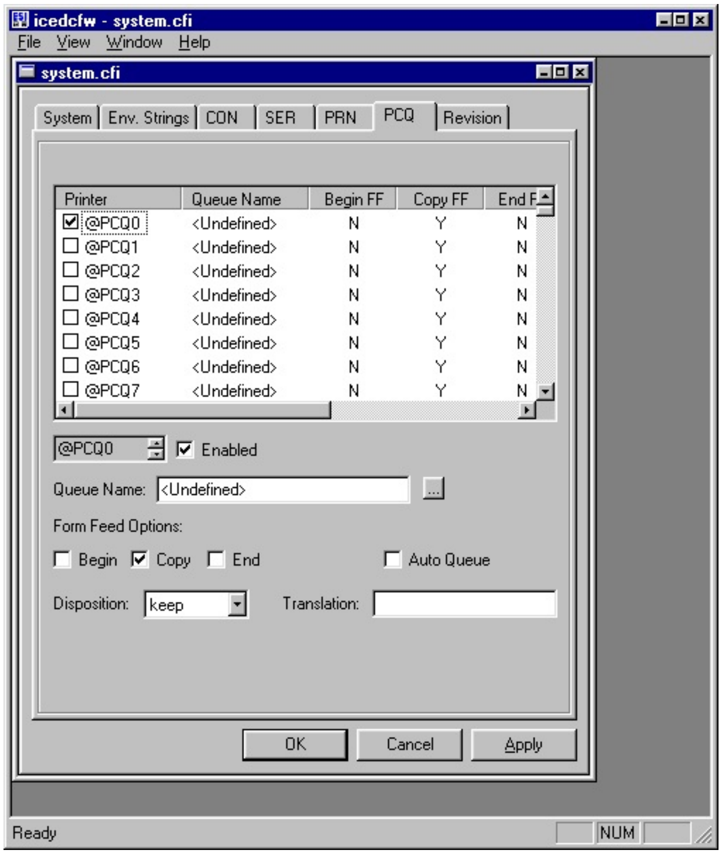
Printers that are not enabled or for which the hardware device is not available, are ignored and give an error on OPEN.

PRN devices are direct linkages from the Interactive COBOL runtime to the device. When used with the COBOL EXCLUSIVE option, the Interactive COBOL runtime prevents other Interactive COBOL processes from opening that device while it is in use.

G.7. **Printer Queues** tab

Selecting the **PCQ** tab allows the configuration information for all the logical printer control queues (@PCQ0-@PCQ2047) to be defined as shown in SCREEN 8.

This menu defines certain characteristics for each logical printer control queue to be specified.



Printer	Queue Name	Begin FF	Copy FF	End F
☑ @PCQ0	<Undefined>	N	Y	N
□ @PCQ1	<Undefined>	N	Y	N
□ @PCQ2	<Undefined>	N	Y	N
□ @PCQ3	<Undefined>	N	Y	N
□ @PCQ4	<Undefined>	N	Y	N
□ @PCQ5	<Undefined>	N	Y	N
□ @PCQ6	<Undefined>	N	Y	N
□ @PCQ7	<Undefined>	N	Y	N

@PCQ0 ☑ Enabled

Queue Name: <Undefined>

Form Feed Options:

□ Begin ☑ Copy □ End □ Auto Queue

Disposition: keep Translation:

SCREEN 8. ICEDCFW PRINTER QUEUE DEVICE

Where:

Enabled set to Yes allows this printer control queue device to be used. When the Interactive COBOL runtime starts, it checks with the Windows printer subsystem for each enabled printer control queue.

Queue Name directs Interactive COBOL where to send the data for a particular logical printer control queue. Any printer defined to the Windows Printer subsystem can be specified or blank. On Windows 9x, use the name as given in the **Printer Control Panel** (from the **Start** menu, choose **Settings**, **Printers**). On Windows NT/2000, use the printer name as given in the **Printer Control Panel** (from the **Start** menu, choose **Settings**, **Printers**) for local printers and use the network port name for redirected printers. Blank can be used to select the default printer. The ... box can be used to pull-down all the current Windows printers on this machine. ICINFO can also be used to see the available Windows printers on a particular machine.

The *Form Feed* options instructs the Printer Control Utility whether to add additional Form Feeds at certain points when printing to the print device.

Auto Queue instructs the Printer Control utility whether to automatically queue a file to its default print queue when the file has been closed.

Translation specifies a printer translation file to be used when printing. If nothing is specified, each character is printed as given. Printer translation files are opened and read when the runtime is started.

Disposition provides the Printer Control utility with the default option for a particular print file of whether to Keep, Remove, or Delete the particular file or entry after it has been printed.

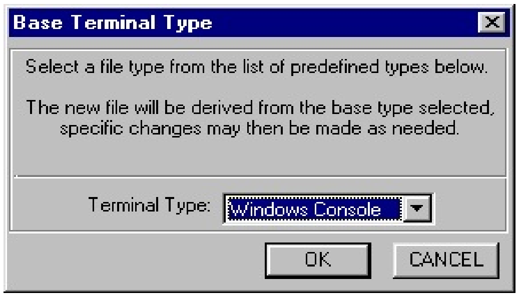
Printer control queues that are not enabled are ignored and give an error on OPEN.

Printer control queues (@PCQs) are indirect linkages from Interactive COBOL to a particular device through Interactive COBOL print spooler. The data is spooled to a disk file and when the file is closed under COBOL the disk file is submitted to the appropriate print spooler to be printed on the indicated print device. If a printer control queue is opened directly, e.g., "@PCQ25", the runtime will write the data directly to the Windows print spooler without creating a temporary file in most cases. In cases where it does need to create a temporary file, the file will NOT be placed in the .pq file or in the ICPCQDIR directory and will be sent to the printer spooler on CLOSE.

H. Terminal Descriptions (.tdi)

H.1. Overview

To create a new terminal description file, select the **File** tab, New, Terminal Description will present a Base Terminal Description as shown below.



Base Terminal Type

Select a file type from the list of predefined types below.

The new file will be derived from the base type selected, specific changes may then be made as needed.

Terminal Type: Windows Console

SCREEN 9. ICEDCFW BASE TERMINAL TYPE

The Base Terminal Type screen allows a base terminal type to be selected from a list of available terminals. These can be selected via a pull-down menu. This base set includes the following:

ANSI class:	AIX Console, ANSI, AT&T 605, AT386 Console, SCO Console, SUN Console, XENIX Console, XTERM Console, 386IX Console	
DG class:	DG D200+	
Disk class:	File	
IBM class:	IBM 3151	
Freedom class:	FREEDOM-One	
VTxxx class:	VT100, VT220, VT220PC, Linux	
Wyse 50 class:	WYSE 50	
Wyse class:	WYSE 60	
PC Window class:	PCWINDOW, PCWINDOWMONO	(On Windows only)
Terminfo class:	TERMINFO	(On UNIX only)

The sets of classes shown above are the default output codes used for each of the particular base terminal descriptions. I.E., all the terminal descriptions in the ANSI class use ANSI codes, those in the DG class use DG control codes, etc. . When in the Configure Display screens these classes are shown at the top as Format.

More information about these base terminal descriptions can be found in the ICTERM Chapter starting on page 173.

Once a base terminal type is selected the file will be named, opened, and positioned to the Keyboard selection as shown in the next section.

A comment can be placed into a terminal description file from any selection. The Comment field allows up to thirty

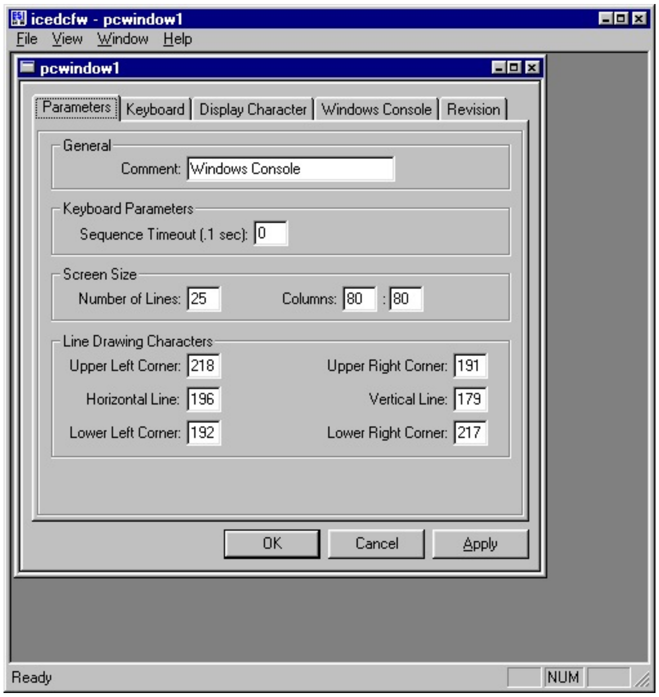
(30) characters to be entered for this terminal description. This option is especially useful if you change a base terminal description to provide some documentation of the change. For example, if you add a HotKey to a DG description you may change the comment to "DG F1 Hot".

H.2. **Parameters** tab

Selecting the Parameters tab gives the PARAMETER CONFIGURATION as seen in SCREEN 10.

The current name of the terminal being configured is displayed at the top of the screen along with its format. The format can be any of the valid display classes mentioned previously.

The PARAMETER CONFIGURATION provides the basic defaults for this terminal description for the number of lines and columns and what characters to use for the line drawing character set. The lines and columns values can be overridden by the ICLINES and/or ICCOLUMNS environment entries either in the actual environment or in the PROGRAM ENVIRONMENT screen specified previously in the configuration file.



General

Comment: Windows Console

Keyboard Parameters

Sequence Timeout (.1 sec): 0

Screen Size

Number of Lines: 25 Columns: 80 : 80

Line Drawing Characters

Upper Left Corner: 218 Upper Right Corner: 191

Horizontal Line: 196 Vertical Line: 179

Lower Left Corner: 192 Lower Right Corner: 217

SCREEN 10. ICEDCFW TERMINAL PARAMETER

The sequence timeout is the number of tenths of seconds to wait for the intercharacter gap between function keys sequences. The default is 5 for most terminals. Some remote connections, using telnet or rlogin, may need to have this number adjusted upward to correctly handle function key sequences.

The line drawing codes are the decimal codes for the ASCII character to be used for each appropriate part of a box. The default values shown are what this terminal supports. If the terminal does not support a real line drawing character set, the default values of 43, 45, and 124 (decimal) which are `+', `-', and `|', are used.

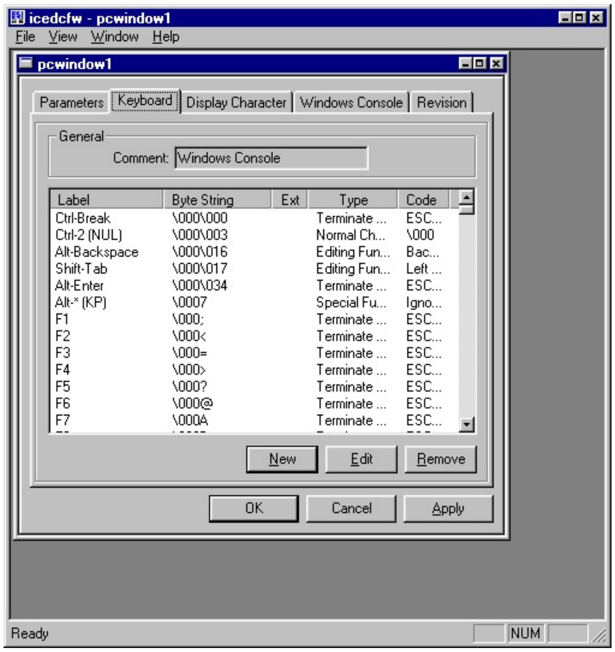
These line drawing codes are used by the SCREEN HANDLER when ICSDMODE has been set to drawlines.

H.3. **Keyboard** tab

The **Keyboard** tab presents the screen as shown in SCREEN 11. The KEYBOARD CONFIGURATION instructs Interactive COBOL what to do for each possible input sequence from this keyboard.

The current name of the terminal being configured is displayed at the top along with its format. The format can be either ASCII or terminfo. In ASCII mode, this table instructs Interactive COBOL how to handle each of the entered ASCII sequences on input from the keyboard. In terminfo mode, this table instructs Interactive COBOL how to handle each of the entered terminfo capname sequences on input from the keyboard.

This table can have up to 512 unique input sequences for a particular terminal description.



Comment: Windows Console

Label	Byte String	Ext	Type	Code
Ctrl-Break	\000\000		Terminate ...	ESC...
Ctrl-2 (NUL)	\000\003		Normal Ch...	\000
Alt-Backspace	\000\016		Editing Fun...	Bac...
Shift-Tab	\000\017		Editing Fun...	Left ...
Alt-Enter	\000\034		Terminate ...	ESC...
Alt-* (KP)	\0007		Special Fu...	Igno...
F1	\000;		Terminate ...	ESC...
F2	\000<		Terminate ...	ESC...
F3	\000=		Terminate ...	ESC...
F4	\000>		Terminate ...	ESC...
F5	\000?		Terminate ...	ESC...
F6	\000@		Terminate ...	ESC...
F7	\000A		Terminate ...	ESC...

SCREEN 11. ICEDCFW TERMINAL DESCRIPTION (ICTERM) – KEYBOARD

To allow a particular terminal description to be available for Interactive COBOL a terminal description file (.TDI) must be created for that particular ICTERM entry. Each of the base terminal descriptions can be used to create a default description file. This can be done by using option 1 (Select Base Terminal) followed by option 7 (Save terminal description file) for each of the possible base terminal selections.

The first column of the table gives the legend or label of the key and any shift or ctrl key that must be pressed to get this key. Interactive COBOL does not use this column in any way, it is useful only as a label.

Column 2 gives the actual codes (series of 8-bit bytes) that this key generates when pressed on the keyboard. Interactive COBOL normally watches for these code(s) in a timing-sensitive fashion and, if seen, will handle as specified by the next field.

When entering data in column 2 the following can be used:

\a enters a bell (Ctrl-G)
\b enters a backspace (Ctrl-H)
\dnnn enters nnn in decimal
\e enters an ESC
\f enters a form-feed (Ctrl-L)
\k allows a terminfo Capname to be
entered when using terminfo
\n enters a linefeed (Ctrl-J)
\r enters a carriage-return (Ctrl-M)

\t enters a tab (Ctrl-I)
\v enters a vertical tab (Ctrl-K)
\xnn enters nn in hex
\z allows for timing-insensitive keystrokes to be entered
\\, \^, \", \', and \? enters a single `\', `^', `"', `'', and `?' respectively
^@ through _ enters the control code (\000 - \037)
\nnn enters nnn in octal
any printable character (`!' - `~') is entered as itself

When displayed on the screen or in the listing, all printable characters (`!' - `~') will be displayed as is, while all others will be shown in octal.

The next columns allow the type to be entered; within that type, how to interpret the key (code), and for next, previous, and terminating types whether to erase to the right of the cursor position. Right-arrow or left-arrow is then used to cycle forward or backward through the valid choices through these ferris-wheel fields.

The up-arrow, down-arrow, F1 (left), F2 (right), F3 (page-up), and F4 (page-down) keys will position to the field to change. F5 can be used to copy a row down.

This table is always stored in sorted order based on the input sequence defined in column 2.

Valid Types with their Codes are defined in the following paragraphs. The numbers in parentheses after the Types and Editing codes are the values returned in the IC_GET_KEY builtin function for the appropriate keystroke.

Normal Character (1) - The runtime system will treat keycodes of this type as normal 8-bit ASCII characters. The Code column contains the actual character code to be used by the runtime system. The value can be 0 to 255 (decimal).

Editing Function (2) - This type of keycode instructs the runtime system to take the action as described in the Code column. The possible actions in the Code column are:

clear to end-of-field (7)
end of field (9)
left word (11)
left TAB stop (13)
left a character (1)
backspace (3)
insert mode ON/Off (5)
sound bell (15)

beginning of field (8)
right word (10)
destructive TAB (12)
right TAB stop (14)
right a character (2)
delete a character (4)
clear field (6)
back delete (16)

The Back delete (16) action code does a collapsing backspace (i.e., it deletes the previous character and closes up the line).

TAB settings are set at every fourth character position from the beginning of the field, i.e., 1, 5, 9, ...

Terminate Field (3) - This type of keycode causes the runtime system to accept the current field and set the ESCAPE KEY value to the value given in the Code column. The ESCAPE KEY value can be 00 to 99, but remember that value 99 is used for timeouts.

Previous Field (4) - This type of keycode instructs the runtime system to move to the previous field in a screen. If the current field is not the first field in a screen, the field is accepted and the screen is positioned to the preceding field. If the current field is the first (or only) field in a screen, the result depends on the ESCAPE KEY value associated with the key given in the Code column. If the ESCAPE KEY value is 00 (default), the system will beep, and the screen will remain positioned at the first field. If the ESCAPE KEY value is not 00, the field will be accepted, and the screen will exit with the specified ESCAPE KEY value. The ESCAPE KEY value can be 00 to 99, but remember that the system returns 99 for timeouts.

Next Field (5) - This type of keycode instructs the runtime system to move to the next field in a screen. If the current field is not the last field in a screen, the field is accepted and the screen is positioned to the next field. If the current field is the last (or only) field in a screen the field will be accepted and the screen will exit with the specified ESCAPE KEY value given in the Code column. When using the default value it will act just like a newline had been hit. The ESCAPE KEY value can be 00 to 99, but remember that the system returns 99 for timeouts.

Use of the *Previous Field* and *Next Field* functions on the up-arrow and down-arrow keys with unique ESCAPE KEY values will allow applications to tie several screens together and control the flow from one screen to the other. The default setting of up-arrow is Previous field-ESCAPE KEY 00. The default setting of down-arrow is Next field-ESCAPE KEY 00.

Hot Key Function (6) - Allows for a particular hotkey program to be called whenever this key is entered. The code column allows a value from 00 to 99 to be set such that a COBOL CALL "hotkey<nn>" will be called with the given value replacing the <nn>. There must be a COBOL program available and executable with this name or else a beep will be given.

Previous Row (7) - This type of key code instructs the runtime system to move to the "best fitting" field on a previous row in a screen. If the current field is not in the topmost row of the screen it is accepted and the cursor is positioned to the "best fitting" field. If the current field is in the topmost row of the screen, the result depends on the ESCAPE KEY value associated with the key in Code Column. If the ESCAPE KEY value is 00, the screen will remain positioned in the current field and the bell will sound. If the ESCAPE KEY value is not 00, the field will be accepted and the screen will exit with the specified ESCAPE KEY value. The "best fitting" field is defined to be a field in a preceding row which has the same column position (1st choice), a higher column position (2nd choice) or a lower column position (last choice) than the current field. In any case, the field selected will be the first screen row proceeding the current one which contains ANY fields.

Next Row (8) - This type of key code instructs the runtime system to move to the "best fitting" field on a subsequent row in a screen. If the current field is not in the bottommost row of the screen it is accepted and the cursor is positioned to the "best fitting" field. If the current field is in the bottommost row of the screen, the result depends on the ESCAPE KEY value associated with the key in Code Column. If the ESCAPE KEY value is 00, the screen will remain positioned in the current field and the bell will sound. If the ESCAPE KEY value is not 00, the field will be accepted and the screen will exit with the specified ESCAPE KEY value. The "best fitting" field is defined to be a field in a subsequent row which has the same column position (1st choice), a lower column position (2nd choice) or a higher column position (last choice) than the current field. In any case, the field selected will be the first screen row following the current one which contains ANY fields.

Special Function (0) - Is a set of special internal actions to be taken by the runtime upon receipt of this keystroke. *Special Function* keys do not return in IC_GET_KEY. The actions are defined by the following Codes:

> *Illegal Character* - The runtime system will beep when it receives a keystroke of this type.
>
> *Ignored Character* - The runtime system ignores keycodes of this type.
>
> *Refresh Screen* - The runtime system will clear the current screen and totally refresh the screen from its internal image.
>
> *Enter minus* - This runtime system enters a minus character key followed by a *Terminate Field* with an ESCAPE KEY 0, as two separate keystrokes.

The Erase column is only valid for *Terminate Field*, *Next Field*, *Previous Field*, *Next Row*, and *Previous Row* types. If Erase is set to No (the default), the runtime accepts the current field as currently displayed. If set to Yes, all characters to the right of the cursor in the current field are discarded. It is equivalent to first entering the clear to end-of-field key followed by the same Terminate, Next, or Previous key without the Erase option.

When configuring for UNIX and using the terminfo base setting, Terminfo Capname codes should be entered by preceding the Capname with a backslash (\). For example, *\kcud1* would be entered for cursor down.

When configuring for Windows and using the PCWINDOW base setting, an extended keyboard code can be entered by preceding a null with a backslash (\) followed by the appropriate code. For example, Shift-TAB would be entered as *\000\017*.

Ext

The timing-insensitive option (*Ext*) can be used to configure multi-character keystroke sequences for those terminals that do not support the needed number of function keys. The lead-in character for a timing-insensitive sequence must not have been previously defined as a timing-sensitive character, otherwise an error is given. An example of how timing-insensitive keys can be entered is given below:

> Let's say you want the Ctrl-R character to be the lead-in character for your timing-insensitive codes and you

wish to use Ctrl-R followed by the `1' key for function key F1, followed by a `2' for F2, and so on up to `9', Ctrl-R followed by an `a' for F10, `b' for F11, . . . , `f' for F15 and the shifted states of the second character to get the shifted function keys. I.E., Ctrl-R followed by `!' for Shift-F1, and Ctrl-R followed by `A' for Shift-F10 etc. .

Now, to allow the Ctrl-R character to be the lead-in for the functions keys F1-F15 in the normal and shift states and allow the Ctrl-R Ctrl-R keystrokes to be interpreted as a single Ctrl-R, change the default Ctrl-R Byte String entry from "\022" to "\z\022\022".

Now for each needed function key add the appropriate line. For the first case it would be:

In column 1 (Label), give an appropriate label:
MyF1

In column 2 (Byte String), give the string:
\0221

In column 3 (Ext), enable timeing-insensitice:
Y

In column 4 (Type), give the type code as:
Terminate Field

In column 5 (Code), give the appropriate function key code to be returned:
ESCAPE 2

In column 6 (Erase), give the appropriate value on whether to erase to the right of the cursor:
N

Now continue these responses row by row to add the needed keys as given below:

\0222	to generate F2
.	.
\0229	to generate F9
\022a	to generate F10
.	.
\022f	to generate F15
\022!	to generate Shift-F1
.	.
\022(	to generate Shift-F9
\022A	to generate Shift-F10
.	.
\022F	to generate Shift-F15

Additional timing-insensitive keys can be added by following the above example.

H.4. **Display Characters** tab

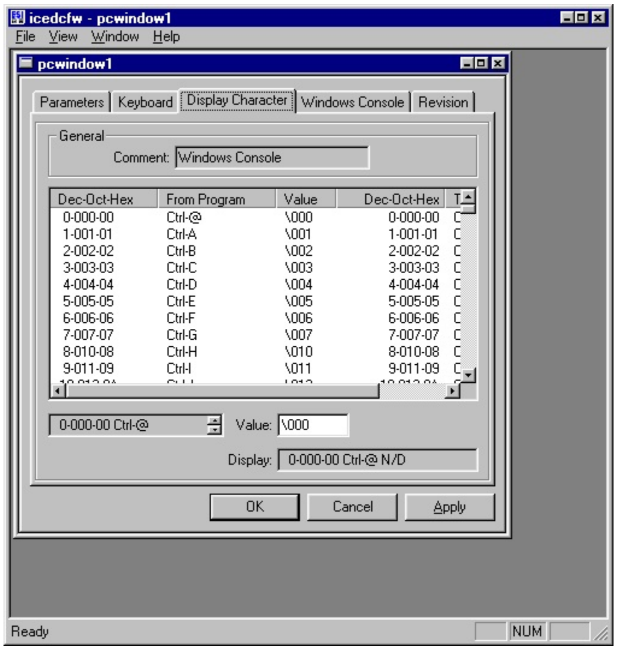
Selecting the **Display Characters** tab presents SCREEN 12. The DISPLAY CHARACTER CONFIGURATION instructs Interactive COBOL what to do for each possible output character from the COBOL program to the terminal. This option is not available for terminfo.

The current name of the terminal being configured is displayed at the top along with its format.

This table provides entries for all 256 possible sequences with each line representing an entry as two major columns: Character from program and Character to Display.

The first major column, Character from Program, gives the character from the program to be output to the terminal in Decimal, Octal, Hex, and as a Description.

The second major column, Character to Display, give a Value column that allows entry of the value to be displayed along with 5 additional sub-columns showing the Decimal, Octal, Hex, Description, and what this output character looks like on this terminal.



General

Comment: Windows Console

Dec-Oct-Hex	From Program	Value	Dec-Oct-Hex	T
0-000-00	Ctrl-@	\000	0-000-00	C
1-001-01	Ctrl-A	\001	1-001-01	C
2-002-02	Ctrl-B	\002	2-002-02	C
3-003-03	Ctrl-C	\003	3-003-03	C
4-004-04	Ctrl-D	\004	4-004-04	C
5-005-05	Ctrl-E	\005	5-005-05	C
6-006-06	Ctrl-F	\006	6-006-06	C
7-007-07	Ctrl-G	\007	7-007-07	C
8-010-08	Ctrl-H	\010	8-010-08	C
9-011-09	Ctrl-I	\011	9-011-09	C

0-000-00 Ctrl-@

Value: \000

Display: 0-000-00 Ctrl-@ N/D

SCREEN 12. ICEDCFW TERMINAL DISPLAY CHARACTER

When entering data in the Values column the following can be used:

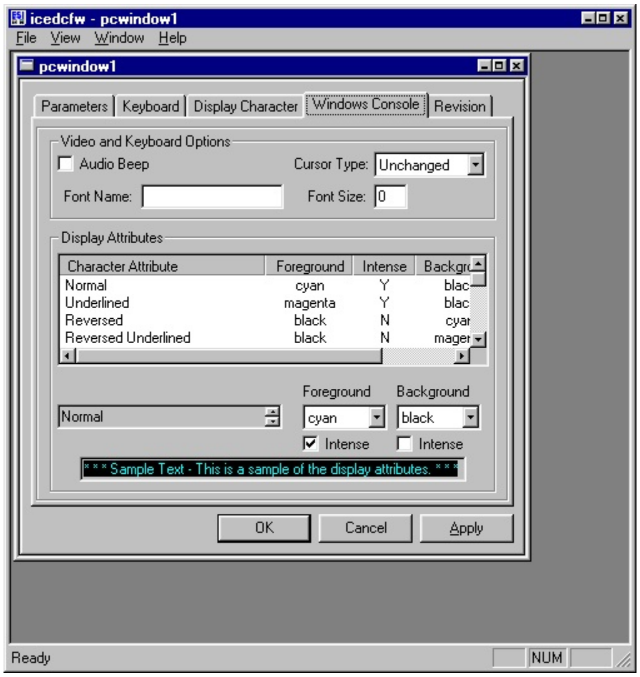
\a enters a bell (Ctrl-G)
\b enters a backspace (Ctrl-H)
\dnnn enters nnn in decimal
\e enters an ESC
\f enters a form-feed (Ctrl-L)
\n enters a linefeed (Ctrl-J)
\r enters a carriage-return (Ctrl-M)

\t enters a tab (Ctrl-I)
\v enters a vertical tab (Ctrl-K)
\xnn enters nn in hex
\\, \^, \", \', and \? enters a single `\', `^', `"', `'', and `?' respectively
^@ through _ enters the control code (\000 - \037)
\nnn enters nnn in octal
any printable character (`!' - `~') is entered as itself

When displayed in the Chr column, all non-control code characters (both 7- and 8-bit) will be displayed as defined within parenthesis ( ), while the control code characters will show a N/D (Not Displayable) in that column.

H.5. **Windows Console** tab (pcwindow)

Selecting the **Windows Console** tab presents SCREEN 13. Here options particular to the Windows Master console can be set. This tab is only available with the pcwindows terminal type.


icedcfw - pcwindow1
File View Window Help
pcwindow1
Parameters | Keyboard | Display Character | Windows Console | Revision
OK
Cancel
Apply
Ready
NUM

Video and Keyboard Options

□ Audio Beep

Cursor Type: Unchanged

Font Name:

Font Size: 0

Display Attributes

Character Attribute	Foreground	Intense	Backgr
Normal	cyan	Y	blac
Underlined	magenta	Y	blac
Reversed	black	N	cyar
Reversed Underlined	black	N	mager

Normal

Foreground: cyan ☑ Intense

Background: black □ Intense

* * * Sample Text - This is a sample of the display attributes. * * *

SCREEN 13. ICEDCFW WINDOWS CONSOLE

This screen configures all the needed Master Console settings.

Audio beep specifies whether to enable the audio beep. The default is for beep to be enabled. In certain cases when the beep is done too often on the Master Console, it can slow the machine for other users, in those cases, it may be wise to disable the beep.

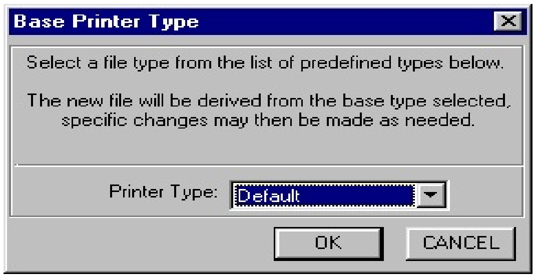
Cursor type specifies what type of cursor to use when the runtime starts. Valid selections are unchanged, line, or block. Unchanged is the default.

For each of the character attribute combinations, the foreground color and intensity and the background color and intensity can be changed. Up to eight colors can be selected. If the color environment (ICCOLOR) has been set to process, these attribute-to-color defaults are NOT used. The actual character attribute is sent to the monitor unchanged.

I. Printer Translations (.pti)

I.1. Overview

To create a new printer translation file, select the **File** tab, New, Printer Translation and the following Base Printer Type screen will be given.



Select a file type from the list of predefined types below.

The new file will be derived from the base type selected, specific changes may then be made as needed.

Printer Type: Default

SCREEN 14. ICEDCFW BASE PRINTER TYPE

From this screen the base printer translation type can be selected from the pull-down menu. Valid selections include the following:

Default: basic one to one mapping
DG to PC: maps the DGI character set to the standard IBM PC character set
PC to DG maps the standard IBM PC character set to the DGI character set

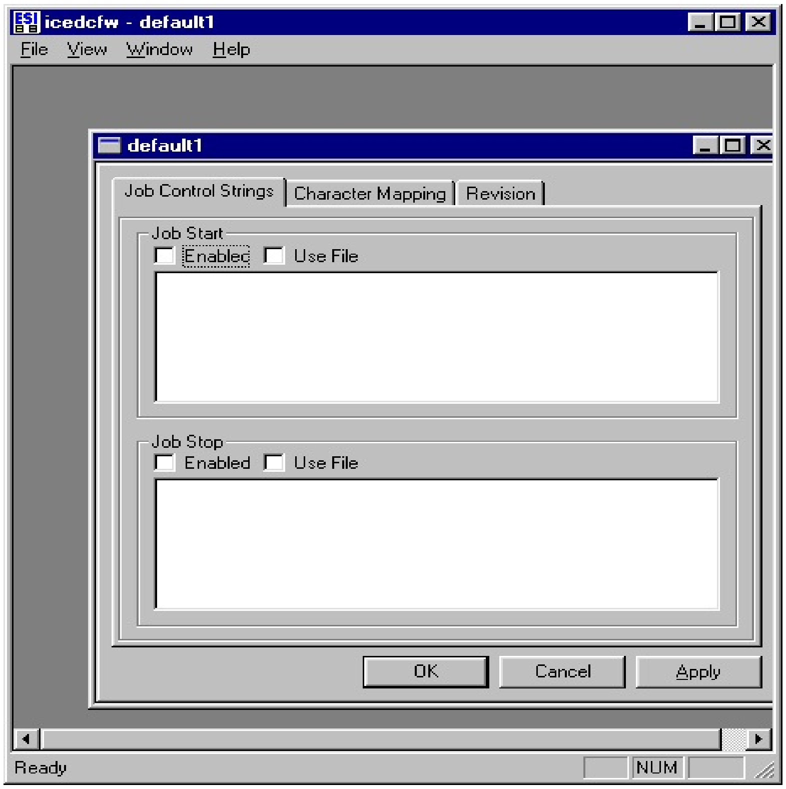
Once a base printer type is selected the file will be named, opened, and positioned to the Character Mapping selection.

To open an existing printer translation file, use the File tab, selecting OPEN and then position to the appropriate directory and file.

The Comment box can be used to insert a comment into a printer translation file. Upto thirty (30) characters can be entered for this printer translation. This option is especially useful if you change a base printer translation to provide some documentation of the change.

I.2. **Job Control Strings** tab

Option 3 from the PRINTER TRANSLATION menu can be used to insert job control strings at the beginning and/or end of a print file. Each can be separately set and can either be a series of values or a filename that contains the codes to be inserted.



SCREEN 15. ICEDCFW PRINTER JOB CONTROL STRING

I.3. **Character Mapping** tab

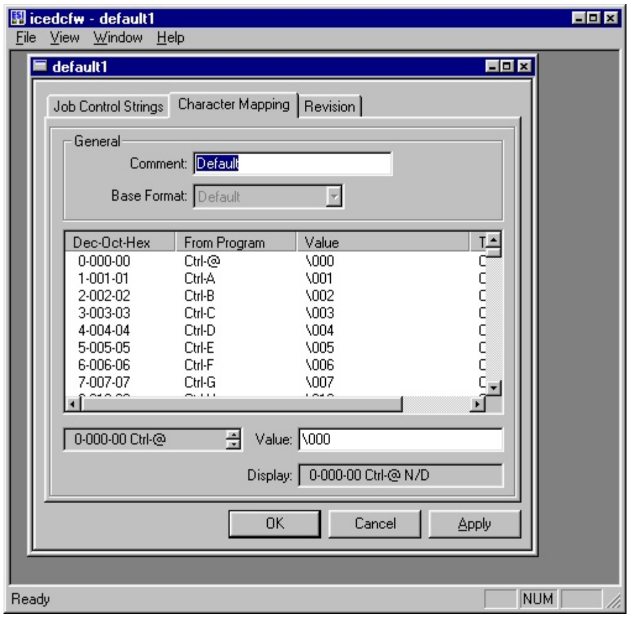
The **Character Mapping** tab presents SCREEN 16. The CHARACTER MAPPING CONFIGURATION instructs Interactive COBOL what to do for each possible output character from the COBOL program to the printer.

The current name of the printer being configured is displayed at the top of the screen.

This table provides entries for all 256 possible sequences with each line representing an entry as two major columns: Character from program and Character to Print.

The first major column, Character from Program, gives the character from the program to be output to the printer in Decimal, Octal, Hex, and as a Description.

The second major column, Character(s) to Print, give a Value(s) column that allows entry of the value(s) to be displayed along with 5 additional sub-columns showing the Decimal, Octal, Hex, Description, and what this output character looks like on this terminal. If more than one character is entered in the Values column, the Dec/Oct/Hex columns will only show “Extended Character Map”. Up to 11 characters are allowed.



Comment: Default

Base Format: Default

Dec-Oct-Hex	From Program	Value	T
0-000-00	Ctrl-@	\000	C
1-001-01	Ctrl-A	\001	C
2-002-02	Ctrl-B	\002	C
3-003-03	Ctrl-C	\003	C
4-004-04	Ctrl-D	\004	C
5-005-05	Ctrl-E	\005	C
6-006-06	Ctrl-F	\006	C
7-007-07	Ctrl-G	\007	C

0-000-00 Ctrl-@ Value: \000

Display: 0-000-00 Ctrl-@ N/D

SCREEN 16. ICEDCFW CHARACTER MAPPING

When entering data in the Values column the following can be used:

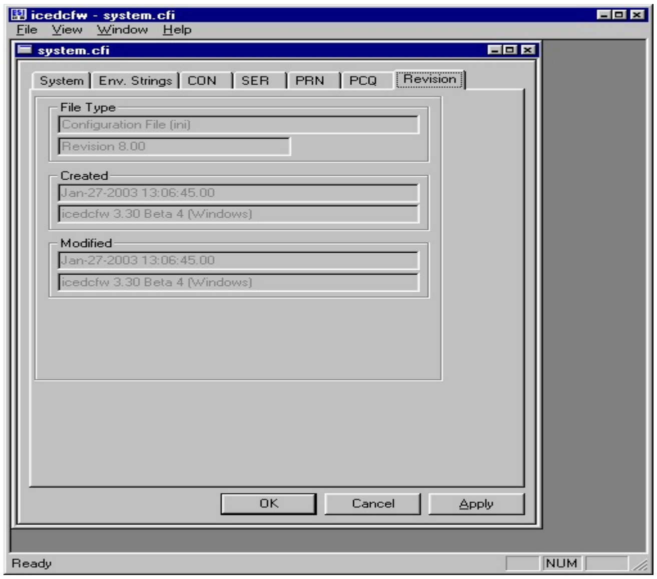
\a enters a bell (Ctrl-G)
\b enters a backspace (Ctrl-H)
\dnnn enters nnn in decimal
\e enters an ESC
\f enters a form-feed (Ctrl-L)
\n enters a linefeed (Ctrl-J)
\r enters a carriage-return (Ctrl-M)

\t enters a tab (Ctrl-I)
\v enters a vertical tab (Ctrl-K)
\xnn enters nn in hex
\\, \^, \", \', and \? enters a single `\', `^', `"', `'', and `?' respectively
^@ through _ enters the control code (\000 - \037)
\nnn enters nnn in octal
any printable character (`!' - `~') is entered as itself

When displayed in the Chr column, all non-control code characters (both 7- and 8-bit) will be displayed as defined within parenthesis ( ), while the control code characters will show a N/D (Not Displayable) in that column.

J. Revision tab

The **Revision** tab is available in most screens to show the current revision information for the selected file.



File Type
Configuration File (ini)
Revision 8.00

Created
Jan-27-2003 13:06:45.00
icedcfw 3.30 Beta 4 (Windows)

Modified
Jan-27-2003 13:06:45.00
icedcfw 3.30 Beta 4 (Windows)

SCREEN 17. ICEDCFW REVISION

K. Exit ICEDCFW

Selecting the Close button [X] from the title bar or selecting Exit from the **File** tab will terminate ICEDCFW. If any modification was done to a file you will be prompted to Save the changes.

VI. ICCONFIG

A. Introduction

ICCONFIG is the Windows character version of the configuration utility. ICEDCFW and/or ICCONFIG can be used to edit configuration files (.cfi), terminal description files (.tdi), and printer translation files (.pti). Each of these files is a .ini based text file with the appropriate sections and definitions.

NOTE: Versions of ICOBOL before 3.30 used configuration files, terminal description files, and printer translation files with the extensions of .cf, .td, and .pt files that were binary files. This format has been discontinued. The ICREVUP utility (documented in the readic.txt file) has been provided to up-rev these files to their .ini-based counterparts. If you need to be able to configure these older files then you must keep an older copy of the configuration utility around to provide that ability. The current configuration utilities will ONLY support the new .ini-based format.

For a new file, ICCONFIG provides a set of default values as a starting point for your configuration. However, every system will require tailoring to account for the requirements of the application, number of users, and system resources available.

FIGURE 1 is a summary of the ICCONFIG menu structure where the #n refers to the SCREEN number in this chapter:

```
MAIN MENU                                 #18
   - SYSTEM CONFIGURATYION (.CFI)         #19
         * SYSTEM PARAMETERS              #20
         * ENVIRONMENT STRINGS            #21
         * CONSOLES AND PROGRAMS (@CONn)  #22
         * SERIALS (@SERn)                #23
         * PRINTERS (@PRNn)               #24
         * PRINTER CONTROL QUEUES (@PCQn) #25
   - TERMINAL DESCRIPTIONS (.TDI)         #26
         * PARAMETERS                     #27
         * KEYBOARD                       #28
         * DISPLAY                        #29
         * COLOR/ATTRIBUTE MAPPING        #30
   - PRINTER TRANSLATIONS                 #31
         * JOB CONTROL STRINGS            #32
         * CHARACTER MAPPING              #33
```

FIGURE 1. ICCONFIG MENU DIAGRAM

Generally we recommend going through each menu sequentially setting the appropriate values. You should understand your particular hardware configuration before getting to the devices.

B. Startup and Main Menu

To start ICCONFIG the syntax is:

```
icconfig [-a[:aflag]|-A file|dir[:aflag]] [-b] [-h|-?] [-l|-L file|dir]
   [-O targetos] [-P ptname] [-q] [-T tdname] [file]
```

Where

-a[:*aflag*] or -A *file*|*dir*[:*aflag*] (Audit)
: Enables auditing (default icconfig.lg). Where *aflag* is a|b|d|p|t|u|da|db|pa|pb|ta|tb|ua|ub, defined as are a-append, b-backup, d-date, p-pid, t-time, and u-username.

-b (Batch)
: Enables batch mode operation.

-h|-? (Help)
: Displays help text.

-l|-L *file*|*dir* (Load .ini)
: Load the appropriate *file*.ini or *dir/file*.ini.

-O *targetos* (Set Operating system environment)
: Specifies the default target operating system environment. If not specified, it defaults based on the current operating environment as UNIX or WINDOWS. Valid selections are UNIX and Windows.

-P *ptname* (Set Printer translation)
: Specifies the default printer translation entry. If not specified, it defaults to a one to one map. Valid selections can be seen with the Help switch.

-q (Quiet)
: Enables quiet operation.

-T *tdname* (Set Terminal description)
: Specifies the default terminal description entry. If not specified, it defaults based on the current operating environment as pcwindow (On Windows) and terminfo (On UNIX). Valid selections can be seen with the Help switch.

file
: Specifies the actual file to be configured. Files with the .cfi extension will go directly to the System Configuration menu, files with the .pti extensions will go directly to the printer translation menu, and files with the .tdi extensions will go directly to the Terminal Description menu. If no extension is given, the `.cfi' extension is added. For a simple file with the .cfi extension the file is sought only in the current directory. For a simple file with the .tdi extension the file is sought in the current directory and then in the $ICROOT/term directory. For a simple file with the .pti extension the file is sought in the current directory and then in the $ICROOT/print directory.

If no argument is provided, ICCONFIG starts in the main menu.

If batch mode operation is specified, ICCONFIG starts and if the configuration file exists it is read and if it was updated it is rewritten and ICCONFIG terminates. If no configuration file exists a default configuration file is created and ICCONFIG terminates.

If both batch mode and Set Terminal description (or Set Printer) are given, ICCONFIG does not use a configuration file but only acts on the given terminal description (or printer translation) just as batch mode alone acts on the configuration file. (Updates it if it exists and needs updating, otherwise creates one).

The -l|-L (Load .ini) switches (Batch update facility) are documented at the end of this section on page 128.

If the given configuration file exists and cannot be read at startup, an error is displayed and ICCONFIG terminates. The error should be fixed before re-running ICCONFIG. If the given configuration file does not exist at startup, a warning is displayed that the file was not found but ICCONFIG continues with the retrieved filename set to blank. If a save is done the file will be saved under the startup name.

For example the line:

```
icconfig samplecf.cfi
```

would set the configuration file to be samplecf.cfi while in ICCONFIG. In the main screen of ICCONFIG the file retrieved (if any) will be displayed and the filename that will be used on a Save or exit with update.

ICCONFIG always assumes the master console type pcwindow.

While in ICCONFIG the general use of keys is:

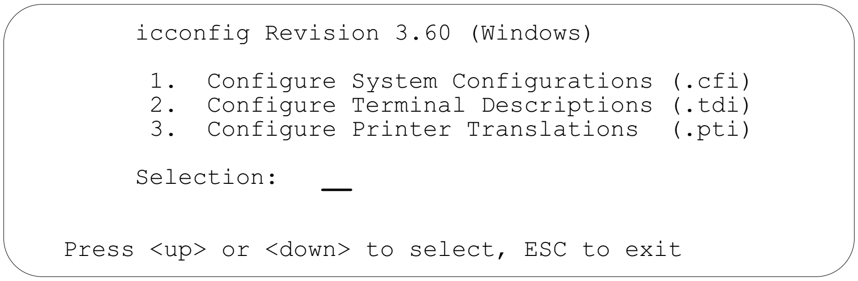
1) the ESC key will exit from the current menu and return to the previous choice, in the MAIN MENU it will exit ICCONFIG.
2) up-arrow (↑) and down-arrow (↓) will move to the entry before or following the current entry. For menus that fit on one screen, at the top it will wrap to the bottom and at the bottom it will wrap to the top. For menus that scroll, at the top it will move to the bottom of the table and at the bottom it will move to the top of the table for up-arrow and down-arrow, respectively. The prompts will show these as <up> and <down> respectively since the arrow-characters are unprintable on most terminals.
3) function keys F1 and F2 will move to the previous field (left) and the next field (right) within the same row of a row-column table menu.
4) function keys F3 and F4 will move to the previous page (up) and the next page (down) keeping the cursor in the same location.
5) function key F5 will copy the fields for the current row or screen to the next row or screen. It is very useful in the initial setup to duplicate a standard entry down through a particular table. In the terminal keyboard configuration table setup it will insert a new entry.
6) the Newline, Carriage-return, or ENTER key will select that choice and, for multi-field menus, move to the next entry.
7) the left-arrow (←) and right-arrow (→) keys act based on the type of field being entered.

Individual fields, within a menu, can be one of several types:

Yes/No fields accept Y, y, N, n, left-arrow, right-arrow, or space to select a value.
Ferris-wheel fields accept left-arrow, right-arrow, or space to select a value.
Data-entry fields, which can be either numeric or character, accept typed in values for the particular selection. Left-arrow and right-arrow in these fields just position within the field.
Menu fields accept either a number, up-arrow, or down-arrow to select a value.

Upon invocation, if no argument is given, ICCONFIG reads the standard configuration file (system.cfi) from the current directory, if available, to set up the initial values for the parameters; otherwise, the standard system defaults for the current operating system are used. ICCONFIG does not follow PATH to find the configuration file. If ICCONFIG detects an error in reading an existing configuration file, an error message is displayed and ICCONFIG terminates.

SCREEN 18 shows the MAIN MENU displayed when ICCONFIG starts. Type the number for the selection to choose or use the up-arrow and down-arrow keys to change the value. Pressing ENTER for the currently displayed value executes that option.



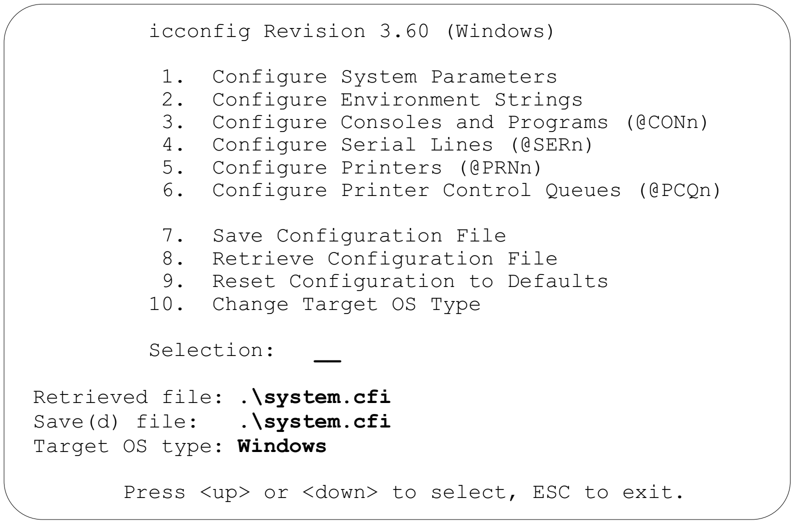
```
         icconfig Revision 3.60 (Windows)

          1.  Configure System Configurations (.cfi)
          2.  Configure Terminal Descriptions (.tdi)
          3.  Configure Printer Translations  (.pti)

         Selection:   __


    Press <up> or <down> to select, ESC to exit
```

SCREEN 18. ICCONFIG MAIN MENU

C. **System Configuration (.cfi)**

C.1. Overview

Option 1 from the MAIN MENU will display SCREEN 19, the SYSTEM CONFIGURATION menu. This menu provides the ability to configure all the sections of system configuration files (.cfi).



```
          icconfig Revision 3.60 (Windows)

           1.  Configure System Parameters
           2.  Configure Environment Strings
           3.  Configure Consoles and Programs (@CONn)
           4.  Configure Serial Lines (@SERn)
           5.  Configure Printers (@PRNn)
           6.  Configure Printer Control Queues (@PCQn)

           7.  Save Configuration File
           8.  Retrieve Configuration File
           9.  Reset Configuration to Defaults
          10.  Change Target OS Type

          Selection:  __

 Retrieved file: .\system.cfi
 Save(d) file:   .\system.cfi
 Target OS type: Windows

        Press <up> or <down> to select, ESC to exit.
```

SCREEN 19. ICCONFIG SYSTEM CONFIGURATION (.cfi)

The Retrieved file line displays the filename that was last retrieved, which may have been from the command line or selections. If no file was retrieved, this field will be blank and default values will be used for this configuration session.

The Save(d) file line displays the filename to which this session of ICCONFIG has been written; using selection 7, otherwise, it will be blank.

The *[target OS]* will show the target operating system for which this configuration is being setup for. If no configuration file is retrieved, it defaults to either "Windows" or "UNIX" based on the current operating system unless ICCONFIG had been started with the Set Operating System switch (-O).

To start from the default case, type 9 to Reset the parameters to the defaults.

Note: All the defaults listed in this manual and in ICCONFIG are what will be used on a Reset or if no configuration file can be found when ICCONFIG starts.

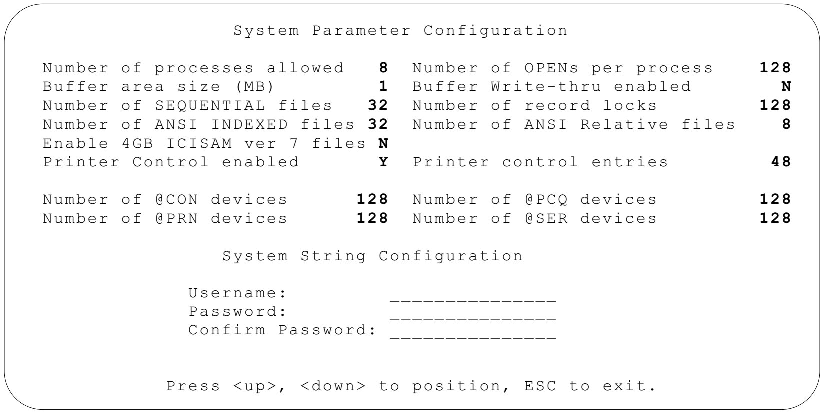
Pressing ESC will exit the screen and if ICCONFIG detects any modification since a Save or a Retrieve, it will prompt with a message asking if you wish to save your changes. If you type Y, a save is done before exiting; otherwise, your changes are discarded.

C.2. Configure System Parameters

Option 1 from the SYSTEM CONFIGURATION menu will show SCREEN 20. These parameters define how the Interactive COBOL runtime allocates various resources. Choose the appropriate values by typing a valid value for each parameter. Press ENTER to choose the displayed value and go on to the next parameter. Up-arrow and down-arrow can also be used to position to the previous or next menu selection. ENTER at the last field will position back to the top. Pressing ESC any time will return to the previous menu after checking to ensure that all parameters are consistent. If there is an error, the menu is re-displayed with the cursor on the offending entry. This entry must be corrected to exit the menu.

Terminal Status in Interactive COBOL will display the actual values that are current for some of these parameters along with the actual in use count for the files, record locks, and several other values.

System Information in Interactive COBOL will display the in use, maximum used, and configured values for many of these parameters during a particular invocation of the runtime.



```
                    System Parameter Configuration

Number of processes allowed   8  Number of OPENs per process    128
Buffer area size (MB)         1  Buffer Write-thru enabled        N
Number of SEQUENTIAL files   32  Number of record locks         128
Number of ANSI INDEXED files 32  Number of ANSI Relative files    8
Enable 4GB ICISAM ver 7 files N
Printer Control enabled       Y  Printer control entries         48

Number of @CON devices      128  Number of @PCQ devices         128
Number of @PRN devices      128  Number of @SER devices         128

                    System String Configuration

             Username:         _______________
             Password:         _______________
             Confirm Password: _______________

          Press <up>, <down> to position, ESC to exit.
```

SCREEN 20. ICCONFIG SYSTEM PARAMETERS

Number of processes, is the maximum number of Interactive COBOL processes including ICEXEC, runtimes, utilities, and compilers that can execute simultaneously when ICEXEC is running. Valid entries are from 1 to 9999; 8 is the default.

Number of OPENs per process is the maximum number of simultaneous file opens that any Interactive COBOL runtime invocation will allow to be opened (i.e., OPEN FD's in a COBOL program). The number of available handles is decreased every time a file is logically opened by any program in the run unit. Valid entries are from 100 to 2048; 128 is the default.

The following 6 items (*Buffer area size*, *Buffer write-thru*, *Number of sequential files*, *Number of Record locks*, *Number of Indexed files*, and *Number of Relative files* apply toICEXEC as a whole.

Buffer area size (MB) is the amount of memory, in MegaBytes, that the runtime system will allocate for buffers. A certain minimum buffer size must be provided that is calculated as "maximum number of processes * 2 * 2KB". Valid entries are from 1 to 1024; 1 is the default. For ICRUN, more than 1MB is generally not needed since most files are opened in network mode such that all writes must flush the data to the network immediately. If large files are opened exclusively for modification then increasing this parameter may be useful. For ICEXEC, this number should be set such that each runtime has a minimum 30/40KB apiece.

Buffer Write-through set to Yes instructs the Interactive COBOL runtime to write all modified pages to the operating system on any operation that modifies data that resides on the disk. Valid entries are Yes or No; No is the default.

Number of SEQUENTIAL files is the number of unique sequential files that can be simultaneously opened by the entire system. Valid entries are from 0 to 4096; 32 is the default.

Number of record locks is the number of simultaneous record locks allowed by the entire system. Valid entries are from 0 to 32767; 128 is the default.

Number of ANSI INDEXED files is the number of unique indexed files that can be simultaneously opened by the entire system. Valid entries are from 0 to 8192; 32 is the default.

Number of ANSI RELATIVE files is the number of unique relative files that can be simultaneously opened by the entire system. Valid entries are from 0 to 4096; 8 is the default.

Enable 4GB ICISAM version 7 files specifies whether to create version 7 ICISAM files with the ability to have 4GB files. If not enabled (No) only 2GB files are allowed. Valid entries are yes and no; No is the default.

Printer Control enabled allows for the Printer Control Utility to be enabled (Yes) or disabled (No). If set to No, files normally placed in the printer control file (system.pq) are not placed there and the IC_PRINT_STAT builtin will return an error. Valid entries are Yes or No; Yes is the default.

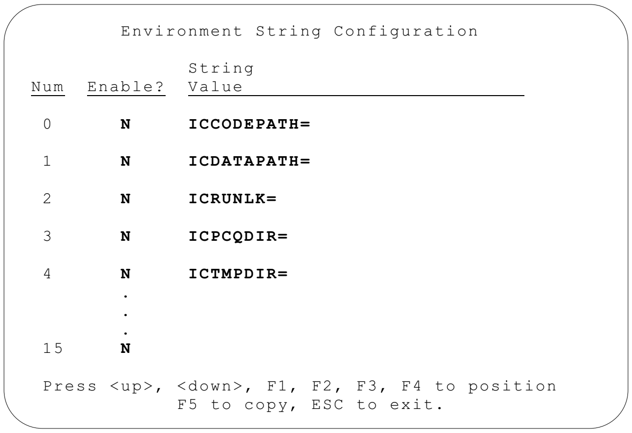
Printer Control entries is the maximum number of entries allowed in the printer control file at once. Once this number is reached, all new files will get a File Status 99 when a new file is being OPEN'ed which would create a new entry in the printer control file. Valid entries are from 48 to 1024 entries; 48 is the default.

Number of @CON, @PRN, @SER, @PCQ devices is the maximum number of each of those devices that you wish to configure. Valid entries are to 2048 entries; 128 is the default. These numbers will be used in the later configuration menus to set the maximum allowed set of devices. The minimum is 1 for @CONs and zero(0) for the others. You do not need to configure more than you actually need.

For Windows configurations, the system strings provide a default username/password for serial consoles that log on without specifying a username/password. This is provided to try to match **ICOBOL** on MS-DOS semantics. In the Program Environment section for these console the username prompt would be disabled. That is the ONLY time this username/password is used by **ICOBOL**. These are only provided when the Target OS is set to Windows.

C.3. Configure Environment Strings

Option 2 from the SYSTEM CONFIGURATION menu allows common environment entries to be specified.



```
        Environment String Configuration

                  String
 Num  Enable?     Value

  0      N        ICCODEPATH=

  1      N        ICDATAPATH=

  2      N        ICRUNLK=

  3      N        ICPCQDIR=

  4      N        ICTMPDIR=
         .
         .
         .
  15     N

  Press <up>, <down>, F1, F2, F3, F4 to position
                  F5 to copy, ESC to exit.
```

SCREEN 21. ICCONFIG ENVIRONMENT STRING

More on environment entries can be found on page 135, 153.

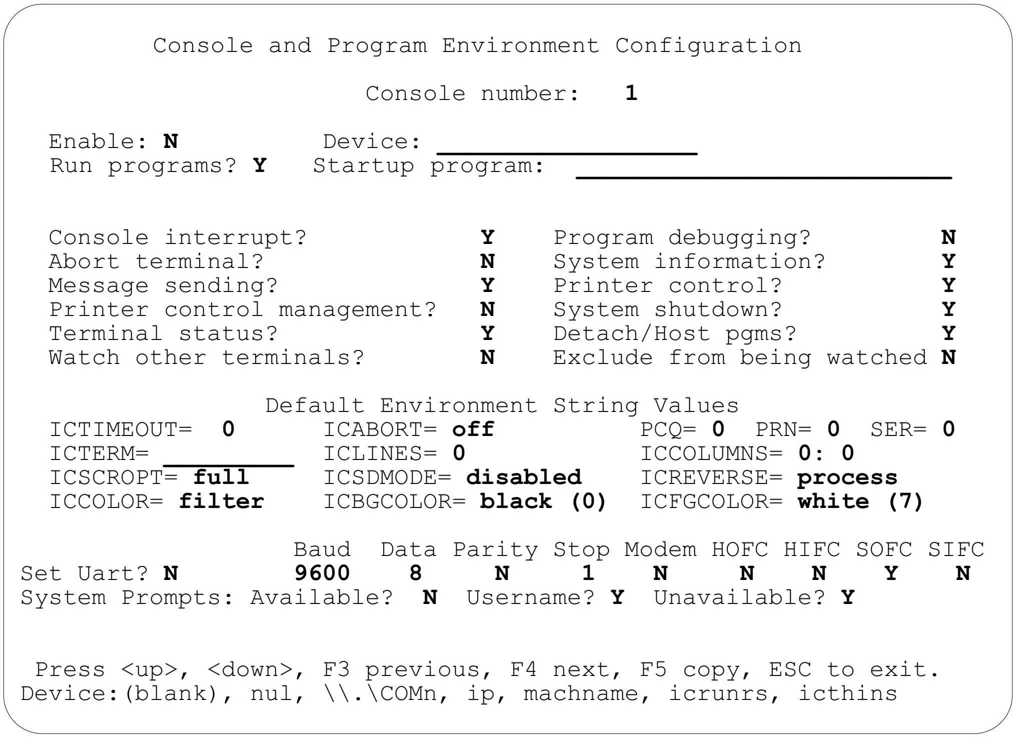
Up to 15 default environment strings can be stored in the configuration file.

An environment string entry allows up to 255 characters.

C.4. Configure Consoles and Programs (@CONn)

Option 3 from the SYSTEM CONFIGURATION menu allows the configuration information for all the logical console lines (@CON0 and up) in the Interactive COBOL system to be defined. Upon selecting option 3, the CONSOLE CONFIGURATION menu is displayed along with the current settings as shown in SCREEN 22. The actual number of entries is controlled by the Number setting in the System Parameters Configuration.

This menu allows a console to be enabled or disabled, an actual hardware device to be assigned to it, and any program setting can be defaulted.



```
         Console and Program Environment Configuration

                      Console number:   1

  Enable: N          Device: __________________
  Run programs? Y   Startup program:  __________________________


  Console interrupt?            Y    Program debugging?        N
  Abort terminal?               N    System information?       Y
  Message sending?              Y    Printer control?          Y
  Printer control management?   N    System shutdown?          Y
  Terminal status?              Y    Detach/Host pgms?         Y
  Watch other terminals?        N    Exclude from being watched N

                 Default Environment String Values
  ICTIMEOUT=  0      ICABORT= off          PCQ= 0  PRN= 0  SER= 0
  ICTERM= _________  ICLINES= 0            ICCOLUMNS= 0: 0
  ICSCROPT= full     ICSDMODE= disabled    ICREVERSE= process
  ICCOLOR= filter    ICBGCOLOR= black (0)  ICFGCOLOR= white (7)

                   Baud  Data Parity Stop Modem HOFC HIFC SOFC SIFC
Set Uart? N        9600    8     N     1    N     N    N    Y    N
System Prompts: Available?  N  Username? Y  Unavailable? Y


 Press <up>, <down>, F3 previous, F4 next, F5 copy, ESC to exit.
Device:(blank), nul, \\.\COMn, ip, machname, icrunrs, icthins
```

SCREEN 22. ICCONFIG CONSOLE and PROGRAM ENVIRONMENT

The first entry is the console to be configured. Valid numbers are 0 to the highest supported console.   Valid selections for each parameter and the defaults are:

Enable set to Yes allows this console to be used by Interactive COBOL.

Device, can be any of the following:

1) “con” allows the console to be available from the Master Console session. (ICTERM=pcwindow). In most cases, many consoles should be configured this way to allow multiple invocations.

2) “icthins” allows the console to be available for ThinClient (gui) programs only.

3) “icrunrs” allows the console to be available for ThinClient programs only.

4) “nul” allows the console to be available for detached programs or for ThinClient (gui) programs.

5) “cgi”, allows the console to be available for cgi programs using icruncgi. You should have enough “cgi” consoles to support the maximum simultaneous number of cgi connections you expect to have.

6) (blank) allows the console to be available for telnet/rlogin/ThinClient sessions.

7) any valid serial device. If a COM port above 9 is specified it must be specified as “\\.\COM10", etc. Only one instance of a serial device can be specified for a console that will allow COBOL programs to be run.

8) a valid *“machine-name*” or *“ip-address”* allows the console to be available for a ThinClient connection from that particular remote machine.

The *machine-name* and/or *ip-address* allows for a specific machine to have a specific console number or numbers.

Multiple consoles can be set to the same device (with the exception of a serial device) to allow multiple runtime sessions to be started on that device. Within the same device name, consoles are used on a first-come first-served basis. A particular console can be selected by using the Terminal number switch (-T *n*).

Run Program specifies whether this console allows a COBOL program to be run on it. Valid selections are Yes or No. The default is Yes for @CON0 but No for all others.

Console lines with the Run program option enabled are called program lines. If the Run Program option is set to No this console device is treated like a serial device and all the remaining settings in this menu are ignored.

The number of program lines (or number of programs) works together with the Maximum number of processes and the licensed number of users to set an upper limit on the maximum number of programs that may be run simultaneously. This maximum number of programs is the lower of these three counts. The lower of the two configured values is shown in the Terminal Status and the System Information screens.

Startup program is the initial COBOL program to run when the runtime initializes this console. Valid selections are no entry or any valid COBOL program with up to 30 characters. The default is no entry causing the COBOL program LOGON (SP2LOGON for ThinClient (gui)) to be run.

Privileges Setup

These options provide for individual control over many system management functions about whether a COBOL program on this console has access to the particular feature. Valid selections are Yes or No.

Console interrupt privilege instructs Interactive COBOL whether to allow the user to abort the currently running COBOL program with the UNIX Intr and Quit key sequences. If set to No, the character is passed on up to the program as data. When set to Yes, the Intr key set for this terminal is trapped by the runtime system and generates an abort to the currently running program and the Quit key for this terminal is trapped by the runtime system and generates a Hangup. The *stty isig* option is NOT used by Interactive COBOL - its initial setting is ignored. The runtime will set and clear the *isig* option as required.

Program debugging privilege: Allows the runtime to be started in debug mode. The default is No.

Abort terminal privilege determines whether COBOL programs running on this console may abort other terminals. The default is No except for CON0.

System Information privilege determines whether COBOL programs running on this console may view the system information screen.

Message sending privilege determines whether COBOL programs running on this console may send messages to other terminals.

Printer control privilege determines whether COBOL programs running on this console may use the Printer Control Utility.

Printer control management privilege is ignored.

The *System Shutdown* privilege is ignored.

Terminal status privilege determines whether COBOL programs running on this console may access the terminal status screen or terminal control utility.

Detach/Host programs privilege determines whether to allow this program to detach COBOL jobs with the IC_DETACH builtin and, whether to allow the "|os-program" call for COBOL programs running on this console.

Watch other terminals privilege determines whether to allow this program to use the Watch Facility (if licensed) to Watch and/or Control another user. If enabled, the Watch and Control commands will be available to this user when in the Terminal Control Utility. If not enabled, the Watch and Control commands will NOT be available.

Exclude from being watched privilege determines whether to allow the program on this terminal to be Watched or Controlled by another terminal. If enabled, then this task will never be allowed to be watcehed or controlled.

Default Environment String Values

This section specifies terminal specific information for a particular console line. The ICTIMEOUT, ICABORT, PCQ, PRN, SER, ICTERM, ICCOLUMNS, ICLINES, ICSCROPT, ICSDMODE, ICREVERSE, ICCOLOR, ICBGCOLOR, and ICFGCOLOR entries can be set in the user's environment to override any of these selections.

ICTIMEOUT sets a default global timeout value in seconds for all ACCEPTs and STOP literals on this console. If no key has been pressed in the specified time interval, the ACCEPT returns with the ESCAPE code set to 99. Valid selections are 0 through 6300; the default is 0 meaning no timeout, i.e., wait forever.

ICABORT instructs Interactive COBOL whether to abort the console (i.e., log it off) if an ACCEPT times out due to the global timeout setting (ICTIMEOUT). Valid selections are off or on; the default is off.

PCQ sets the generic printer control queue (@PCQ) to @PCQn based on the entered number. Valid selections are 0 through 2047; the default is 0. If set to an invalid queue, an error will occur on the OPEN.

PRN sets the generic printer device (@PRN) to @PRNn based on the entered number. Valid selections are 0 through 2047; the default is 0. If set to an invalid printer, an error will occur on the OPEN.

SER sets the generic serial device (@SER) to @SERn based on the entered number. Valid selections are 0 through 2047; the default is 0. If set to an invalid serial device, an error will occur on the OPEN.

ICTERM specifies the terminal description entry to be used for this console. The default for all consoles is terminfo. Valid ICTERM selections are valid terminal description entries with corresponding .TDI files. For an enabled console, this menu cannot be exited without some selection specified for ICTERM.

ICCOLUMNS and *ICLINES* set the number of columns and lines that will be allowed on this console. Valid selections are 0 through 255. The default of 0 says use the values specified in the terminal description entry. For terminfo and pcwindow descriptions, the defaults are those defined in the terminfo database and by the video bios hardware, respectively. These values indicate to Interactive COBOL where the screen wraps (ICCOLUMNS) and scrolls (ICLINES). If set incorrectly, screens may not display properly. The second selection for ICCOLUMNS is for compressed mode if supported by the terminal.

ICSCROPT specifies how the Interactive COBOL SCREEN OPTIMIZER is to perform. Valid selections are off, partial, full, and mute. The default is off. All enabled consoles have at least one screen area reserved. For a 24x80 column screen, a single screen image consumes about 8KB.

OFF says to transmit character codes as they are written by the program.
Partial enables the SCREEN OPTIMIZER to use a simple method of reducing the amount of characters sent to the terminal by comparing data to the single screen image.
Full enables the SCREEN OPTIMIZER to allocate an additional image of the current screen in memory and provides a complex method of comparing the data in the two images to reduce the amount of characters sent to the terminal to only those characters that would change the screen display at the end of an operation.
Mute forces Interactive COBOL to not send any implied codes to the terminal either at startup or termination. Only when executing a program instruction are codes sent to the terminal.

Ctrl-U from the keyboard while in an ACCEPT can be used to refresh the screen.

Usually partial and/or full will provide improved screen performance. Full is preferred with the debugger.

ICSDMODE specifies whether to enable the SCREEN HANDLER and if so in what mode. Valid selections are disabled, underline (0), reverse (1), and drawlines (2). The default is disabled. Drawlines uses the characters for line drawing specified in the terminal description file for the particular terminal.

ICREVERSE instructs Interactive COBOL how to interpret reverse codes from a COBOL program. Valid selections are filter, ignore, and process. The default is process. Reverse codes are Ctrl-B and Ctrl-V along with the two-byte sequences <036>D and <036>E.

Filter tells Interactive COBOL to watch for reverse codes from the program and to NOT send them to the terminal, since it does not support reverse.

Ignore tells Interactive COBOL that the user wants total control of the screen and may be sending binary reverse data to the screen and that Interactive COBOL should ignore all reverse codes (i.e., do not look for reverse codes). If running in this mode, the SCREEN OPTIMIZER cannot correctly repaint a user's screen that includes reverse codes.

Process tells Interactive COBOL to interpret reverse codes from the program and send the appropriate sequences to the terminal.

ICCOLOR instructs Interactive COBOL how to interpret color codes from a COBOL program. Valid selections are filter, ignore, and process. The default is filter.

Filter tells Interactive COBOL to watch for color codes from the program and to NOT send them to the terminal, since it does not support color.

Ignore tells Interactive COBOL that the user wants total control of the screen and may be sending binary color data to the screen and that Interactive COBOL should ignore all color codes (i.e., do not look for color codes). If running in this mode, the SCREEN OPTIMIZER cannot correctly repaint a user's screen that includes colors.

Process tells Interactive COBOL to interpret color codes from the program and send the appropriate sequences to the terminal. When set to Process, the initial background and foreground colors are set by Interactive COBOL at startup.

ICBGCOLOR sets the initial background color to the indicated value when running with ICCOLOR set to Process. Valid selections are black (0), blue (1), green (2), cyan (3), red (4), magenta (5), brown (6), and white (7). The default is black (0).

ICFGCOLOR sets the initial foreground color to the indicated value when running with Color support (ICCOLOR) set to Process. Valid selections are black (0), blue (1), green (2), cyan (3), red (4), magenta (5), brown (6), and white (7). The default is white (7).

Baud Rate support:

***S**et* instructs the runtime whether to use the following port device settings. If set to No, the default, the following settings are not used and an open of the device will use whatever system defaults there are. If set to Yes, the runtime will attempt to set the device settings to the given values.

Baud Rate, *Data Bits*, *Parity*, and *Stop bits* set the appropriate selection to the given value. These values must match those of the attached device or else undefined results will occur. Generally the baud rate should be set to the fastest possible setting that the runtime, port, wiring, and output device can support.

Mdm Ctl (modem control) tells Interactive COBOL how to handle modem control signal Data Carrier Detect (DCD) for a particular device. If Mdm Ctl is set to No, DCD is ignored.

Mdm Ctl set to Yes for program lines (i.e., an @CON logical device is directed to them with the Run Program option set to Yes) causes the initial banner message to be held until Data Carrier Detect (DCD) is asserted, at which time the message will be displayed and the remote user can logon. If the line was set to auto run, the initial program will start when DCD is asserted. If DCD is lowered while a program is active, the program is aborted and the terminal is logged off .

Mdm Ctl set to Yes for serial lines will cause an OPEN to wait until DCD is asserted before returning. If DCD is lowered after the OPEN, the next I/O operation to that line will receive an error and the appropriate action will be taken.

Hrd O (HOFC) tells Interactive COBOL how to handle the Clear To Send (CTS) signal. HOFC set to No says to ignore CTS. HOFC set to Yes says to monitor CTS to decide if the attached device is ready to receive output. If not, it waits until CTS is asserted. Data Set Ready (DSR) must also be asserted for characters to be sent.

Hard I (HIFC) tells Interactive COBOL how to handle the Request to send (RTS) signal. HIFC set to No says to leave RTS high after an open. HIFC set to Yes says to raise and lower RTS to instruct the attached device if it can send data.

HOFC and HIFC should both be enabled when dealing with high-speed modems as SOFC and SIFC are not reliable in these cases.

Sft O (SOFC) tells Interactive COBOL whether to watch for a Ctrl-S (XOFF) coming from the output device to tell Interactive COBOL to stop sending characters and then wait for a Ctrl-Q (XON) to continue transmitting. SOFC set to No says to ignore XON and XOFF codes while Yes says to watch for an XOFF and stop transmitting until an XON is seen.

Sft I (SIFC) tells Interactive COBOL whether to send Ctrl-S (XOFF) and Ctrl-Q (XON) to the input device to tell it to stop transmitting characters and start transmitting characters respectively. This should NEVER be used for a DG terminal or emulator because the Ctrl-S and Ctrl-Q codes are valid attribute control codes and are NOT recognized as XOFF and XON.

These selections are ONLY used by ICEXEC on Windows when supporting serial consoles. These selections allow the Windows product to more closely match the **ICOBOL** for MS-DOS multi-user product.

System Available specifies whether to prompt with a “System is ready” message. The default is no.

Username specifies whether to prompt for username/password. The default is yes. If disabled, a username/password must be found somewhere to allow ICEXEC to log the user on. This is done with the default username/password pair stored in the System Strings section of the configuration.

Unavailable specifies whether to show a “System is unavailable” message when ICEXEC shutdowns.

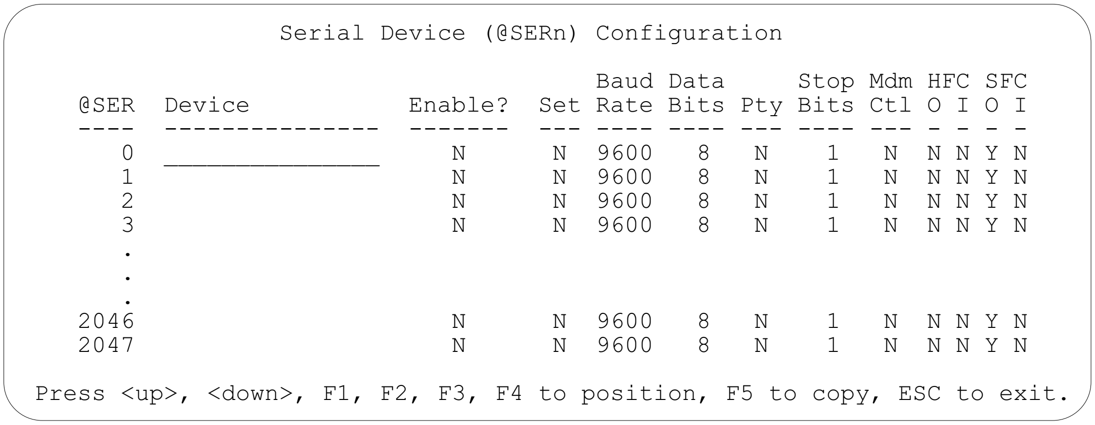
Console lines that are not enabled are ignored. Console lines for which the hardware is not present give a warning at startup and an error on OPEN.

The table is searched from lowest to highest to decide the console-number for programs so the first occurrence of the device will select the console.

C.5. Configure Serial Lines (@SERn)

Option 4 from the SYSTEM CONFIGURATION menu defines the configuration information for the logical serial devices (@SER0 - @SER2047) in the Interactive COBOL system to be defined. Upon selecting option 4, the SERIAL CONFIGURATION menu is displayed along with the current settings as shown in SCREEN 23. The actual number of entries is controlled by the Number setting in the System Parameters Configuration. A serial device differs from a console device in that it can only be used for I/O. Console devices can be used to run programs or for I/O.

This menu defines logical serial devices to point to a particular hardware device.



```
                   Serial Device (@SERn) Configuration

                                                 Baud Data     Stop Mdm  HFC  SFC
   @SER  Device           Enable?  Set Rate Bits Pty Bits Ctl  O I  O I
   ----  ---------------  -------  --- ---- ---- --- ---- ---  - -  - -
      0  _______________     N      N  9600   8   N    1   N   N N  Y N
      1                      N      N  9600   8   N    1   N   N N  Y N
      2                      N      N  9600   8   N    1   N   N N  Y N
      3                      N      N  9600   8   N    1   N   N N  Y N
      .
      .
      .
   2046                      N      N  9600   8   N    1   N   N N  Y N
   2047                      N      N  9600   8   N    1   N   N N  Y N

 Press <up>, <down>, F1, F2, F3, F4 to position, F5 to copy, ESC to exit.
```

SCREEN 23. ICCONFIG SERIAL CONFIGURATION

Valid selections for each parameter and the defaults are:

Parameter	Valid Selections	Default
Device	Any serial (COM, MDN) device	(blank)
Enable	Yes or No	No
S*et*	Yes or NO	No
Baud rate	Any valid baud rate	9600
Data Bits	7 or 8	8
Parity	Yes or No	No
Stop Bits	1 or 2	1
Mdm Ctl	Yes or No	No
HFC O	Yes or No	No
HFC I	Yes or No	No
SFC O	Yes or No	Yes
SFC I	Yes or No	No

Explanations:

Device can be any of the hardware character devices except for parallel printers that support both input and output operations. Two or more serial devices (@SERn) can point to the same hardware device.

Enable set to Yes allows this serial device to be used.

S*et* instructs the runtime whether to use the following port device settings. If set to No, the default the following settings are not used and an open of the device will use whatever system defaults there are. If set to Yes, the runtime will attempt to set the device settings to the given values.

Baud Rate, *Data Bits*, *Parity*, and *Stop bits* set the appropriate selection to the given value. These values must match those of the attached device or else undefined results will occur. Generally the baud rate should be set to the fastest possible setting that the runtime, port, wiring, and attached device can support.

Mdm Ctl (modem control) tells Interactive COBOL how to handle modem control signal Data Carrier Detect (DCD) for a particular device. If Mdm Ctl is set to No, DCD is ignored.

Mdm Ctl set to Yes for serial lines will cause an OPEN to wait until DCD is asserted before returning. If DCD is lowered after the OPEN, the next I/O operation to that line will receive an error and the appropriate action will be taken.

HFC O (HOFC) tells Interactive COBOL how to handle the Clear To Send (CTS) signal. HOFC set to No says to ignore CTS. HOFC set to Yes says to monitor CTS to decide if the attached device is ready to receive output. If not, it waits until CTS is asserted. Data Set Ready (DSR) must also be asserted for characters to be sent.

HFC I (HIFC) tells Interactive COBOL how to handle the Request to send (RTS) signal. HIFC set to No says to leave RTS high after an open. HIFC set to Yes says to raise and lower RTS to instruct the attached device if it can send data.

HOFC and HIFC should both be enabled when dealing with high-speed modems as SOFC and SIFC are not reliable in these cases.

SFC O (SOFC) tells Interactive COBOL whether to watch for a Ctrl-S (XOFF) coming from the output device to tell Interactive COBOL to stop sending characters and then wait for a Ctrl-Q (XON) to continue transmitting. SOFC set to No says to ignore XON and XOFF codes while Yes says to watch for an XOFF and stop transmitting until an XON is seen.

SFC I (SIFC) tells Interactive COBOL whether to send Ctrl-S (XOFF) and Ctrl-Q (XON) to the input device to tell it to stop transmitting characters and start transmitting characters respectively. This should NEVER be used for a DG terminal or emulator because the Ctrl-S and Ctrl-Q codes are valid attribute control codes and are NOT recognized as XOFF and XON.

A hardware device configured as a console device enabled to run programs cannot also be used as a serial device.

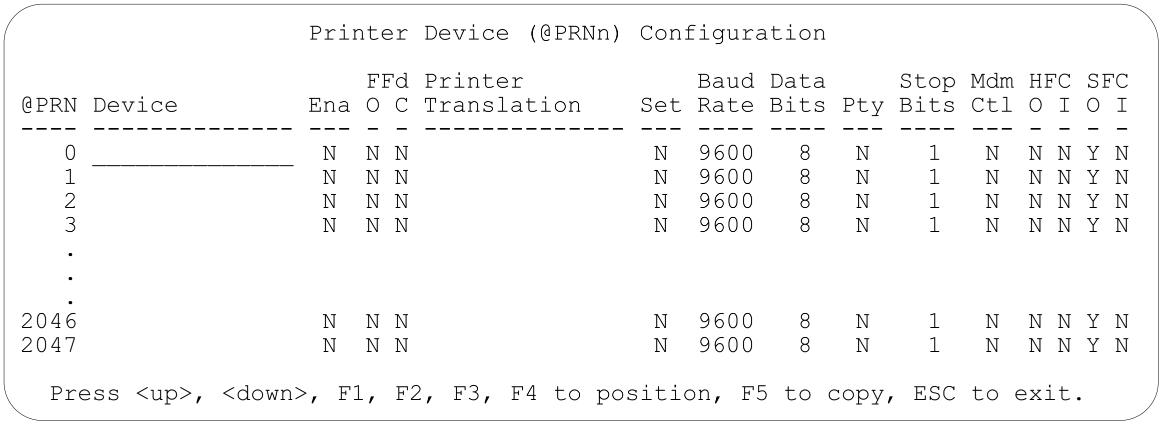
Serial devices that are not enabled or for which the hardware device is not available are ignored and give an error on OPEN.

When opening serial devices and Set is No, the runtime uses the last setting for that serial device to set up the default parameters (baud, parity, number of data bits, etc.). The MODE command can be used to perform these settings if needed. This MODE will be remembered until another setting is stored. In addition, extended open options can be used to set the needed values.

C.6.   Configure Printers (@PRNn)

Option 5 from the SYSTEM CONFIGRATION menu defines the configuration information for logical printers (@PRN0 - @PRN2047) in the Interactive COBOL system to be defined. Upon selecting option 5, the PRINTER CONFIGURATION menu is displayed along with the current settings as shown in SCREEN 24. The actual number of entries is controlled by the Number setting in the System Parameters Configuration.

This menu defines certain characteristics for each logical printer to be specified.



```
                    Printer Device (@PRNn) Configuration

                        FFd Printer            Baud Data     Stop Mdm HFC SFC
@PRN Device         Ena O C Translation    Set Rate Bits Pty Bits Ctl O I O I
---- -------------- --- - - -------------- --- ---- ---- --- ---- --- - - - -
   0 ______________  N  N N                 N  9600   8   N    1   N  N N Y N
   1                 N  N N                 N  9600   8   N    1   N  N N Y N
   2                 N  N N                 N  9600   8   N    1   N  N N Y N
   3                 N  N N                 N  9600   8   N    1   N  N N Y N
   .
   .
   .
2046                 N  N N                 N  9600   8   N    1   N  N N Y N
2047                 N  N N                 N  9600   8   N    1   N  N N Y N

  Press <up>, <down>, F1, F2, F3, F4 to position, F5 to copy, ESC to exit.
```

SCREEN 24. ICCONFIG PRINTER CONFIGURATION

Valid selections for each parameter and the defaults are:

Parameter	Valid Selections	Default
Device	Any serial (COM, MDM) or parallel (LPT) device	(blank)
***E**nable*	Yes or No	No
*FF on **O**PEN*	Yes or No	No
*FF on **C**LOSE*	Yes or No	No
Printer Translation	filename	(blank)
***S**et*	Yes or No	No
Baud Rate	Any valid baud rate	9600
Data Bits	7 or 8	8
Parity	Yes or NO	No
Stop Bits	1 or 2	1
Mdm Ctl	Yes or No	No
HFC O	Yes or No	No
HFC I	Yes or No	No
SFC O	Yes or No	Yes
SFC I	Yes or No	No

Explanations:

Device directs the Interactive COBOL runtime where to send the print data for a particular logical printer. Possible selections are any of the hardware character devices that are not in use as terminal lines or blank for None. Two or more printers (@PRNn) can point to the same hardware device. This would be useful if different printer options are needed.

Enable set to Yes allows this printer device to be used.

FF on OPEN and *FF on CLOSE* instructs the runtime whether to send a Form-Feed to the printer when the appropriate statement is executed on a particular printer.

Printer Translation specifies a printer translation file to be used when printing. If nothing is specified, each character is printed as given. Printer translation files are opened and read when the runtime system is started.

***S**et* instructs the runtime whether to use the following port device settings. If set to No, the default the following settings are not used and an open of the device will use whatever system defaults there are. If set to Yes, the runtime will attempt to set the device settings to the given values. These device settings are ignored for parallel ports.

Baud Rate, *Data Bits*, *Parity*, and *Stop bits* set the appropriate selection to the given value. These values must match those of the attached device or else undefined results will occur. Generally the baud rate should be set to the fastest possible setting that the runtime, port, wiring, and output device can support.

Mdm Ctl (modem control) tells Interactive COBOL how to handle modem control signal Data Carrier Detect (DCD) for a particular device. If Mdm Ctl is set to No, DCD is ignored.

Mdm Ctl set to Yes for serial lines will cause an OPEN to wait until DCD is asserted before returning. If DCD is lowered after the OPEN, the next I/O operation to that line will receive an error and the appropriate action will be taken.

HFC O (HOFC) tells Interactive COBOL how to handle the Clear To Send (CTS) signal. HOFC set to No says to ignore CTS. HOFC set to Yes says to monitor CTS to decide if the attached device is ready to receive output. If not, it waits until CTS is asserted. Data Set Ready (DSR) must also be asserted for characters to be sent.

HFC I (HIFC) tells Interactive COBOL how to handle the Request to send (RTS) signal. HIFC set to No says to leave RTS high after an open. HIFC set to Yes says to raise and lower RTS to instruct the attached device if it can send data.

HOFC and HIFC should both be enabled when dealing with high-speed modems as SOFC and SIFC are not reliable in these cases.

SFC O (SOFC) tells Interactive COBOL whether to watch for a Ctrl-S (XOFF) coming from the output device to tell Interactive COBOL to stop sending characters and then wait for a Ctrl-Q (XON) to continue transmitting. SOFC set to No says to ignore XON and XOFF codes while Yes says to watch for an XOFF and stop transmitting until an XON is seen.

SFC I (SIFC) tells Interactive COBOL whether to send Ctrl-S (XOFF) and Ctrl-Q (XON) to the input device to tell it to stop transmitting characters and start transmitting characters respectively. This should NEVER be used for a DG terminal or emulator because the Ctrl-S and Ctrl-Q codes are valid attribute control codes and are NOT recognized as XOFF and XON.

A hardware device configured as a console device enabled to run programs cannot also be used as a printer device.

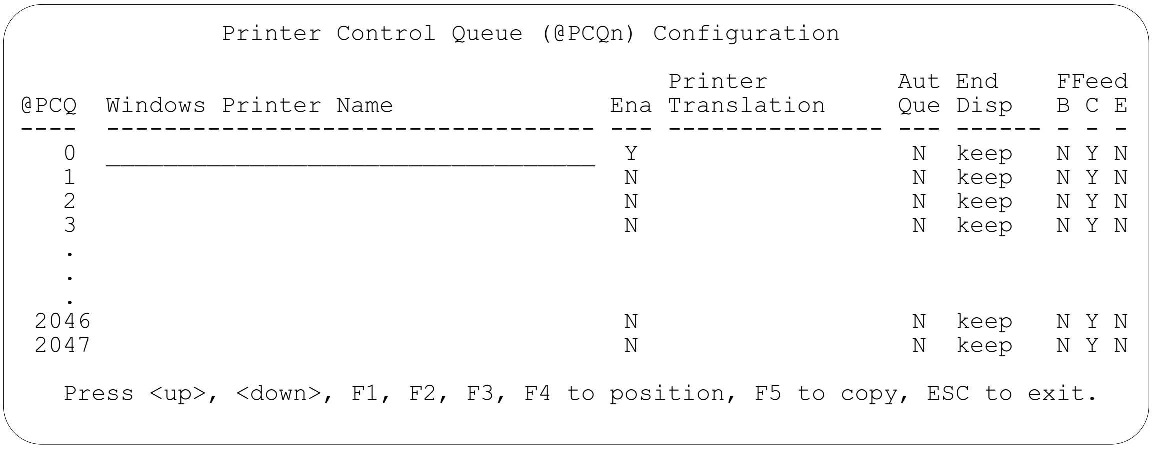
Printers that are not enabled or for which the hardware device is not available, are ignored and give an error on OPEN.

PRN devices are direct linkages from the Interactive COBOL runtime to the device. When used with the COBOL EXCLUSIVE option, the Interactive COBOL runtime prevents other Interactive COBOL processes from opening that device while it is in use.

C.7. Configure Printer Control Queues (@PCQn)

Option 6 from the SYSTEM CONFIGURATION menu defines the configuration information for up to 2048 printer control queues (@PCQ0 - @PCQ2047) in the Interactive COBOL system to be defined. Upon selecting option 6, the PRINTER QUEUE CONFIGURATION menu is displayed along with the current settings as shown in SCREEN 25. The actual number of entries is controlled by the Number setting in the System Parameters Configuration.

This menu defines certain characteristics for each logical printer control queue to be specified.



```
               Printer Control Queue (@PCQn) Configuration

                                               Printer         Aut End    FFeed
@PCQ  Windows Printer Name                 Ena Translation     Que Disp   B C E
----  ---------------------------------- --- --------------- --- ------ - - -
   0  __________________________________  Y                   N  keep   N Y N
   1                                      N                   N  keep   N Y N
   2                                      N                   N  keep   N Y N
   3                                      N                   N  keep   N Y N
   .
   .
   .
2046                                      N                   N  keep   N Y N
2047                                      N                   N  keep   N Y N

   Press <up>, <down>, F1, F2, F3, F4 to position, F5 to copy, ESC to exit.
```

SCREEN 25. ICCONFIG WINDOWS PRINTER QUEUE CONFIGURATION

Valid selections for each parameter and the defaults are:

Parameter	Valid Selections	Default
Windows Printer	Any Printer defined as a Windows Printer or (blank)	(blank)
Enable	Yes or No	No (@PCQ0 Yes)
Printer Translation	filename	(blank)
Auto Queue	Yes or No	No
Disposition	**K**eep, **R**emove, or **D**elete	**K**eep

Explanations:

Windows Printer directs Interactive COBOL where to send the data for a particular logical printer control queue. Any printer defined to the Windows Printer subsystem can be specified. Use the printer name as given in the **Printer Control Panel** (from the **Start** menu, choose **Settings**, **Printers**) for local printers and use the network port name for redirected printers. Blank can be used to select the default printer. ICINFO can also be used to see the available Windows printers on a particular machine.

Enable set to Yes allows this printer control queue device to be used. When the Interactive COBOL runtime starts, it checks with the Windows printer subsystem for each enabled printer control queue.

Printer Translation specifies a printer translation file to be used when printing. If nothing is specified, each character is printed as given. Printer translation files are opened and read when the runtime is started.

Auto Queue instructs the Printer Control utility whether to automatically queue a file to its default print queue when the file has been closed.

Ending Disposition provides the Printer Control utility with the default option for a particular print file of whether to Keep, Remove, or Delete the particular file or entry after it has been printed.

Form Feed options allows for a form-feed to be inserted at the beginning (*B*), end of a copy (*C*), or the end of the printout (*F*).

Printer control queues that are not enabled are ignored and give an error on OPEN.

Printer control queues (@PCQs) are linkages from Interactive COBOL to a particular device through Interactive COBOL print spooler. If a printer control queue is opened directly, e.g., "@PCQ25", the that data is written directly to the Windows printer.

C.8. Save

The Save selection allows any changes of the current values in the configuration file to be saved. You are prompted on the Save(d) file line for the filename (without the `.cfi' extension) to which this session should be saved. The default will be the retrieved name or system.cfi. If no Save is done, then any changes made are NOT saved. If you try to exit and no Save has been done since the configuration was last modified, you are prompted whether a Save should be done.

Note:

Once a Save is done with new values, ICEXEC will not see the changes until the next time ICEXEC is executed. I.E., you cannot change the configuration in effect while ICEXEC is running, although the configuration file can be modified.

C.9. Retrieve

The Retrieve selection allows a configuration file to be read such that it can be viewed or updated. The filename to be retrieved is prompted for on the Retrieved file line.

C.10. Reset to Defaults

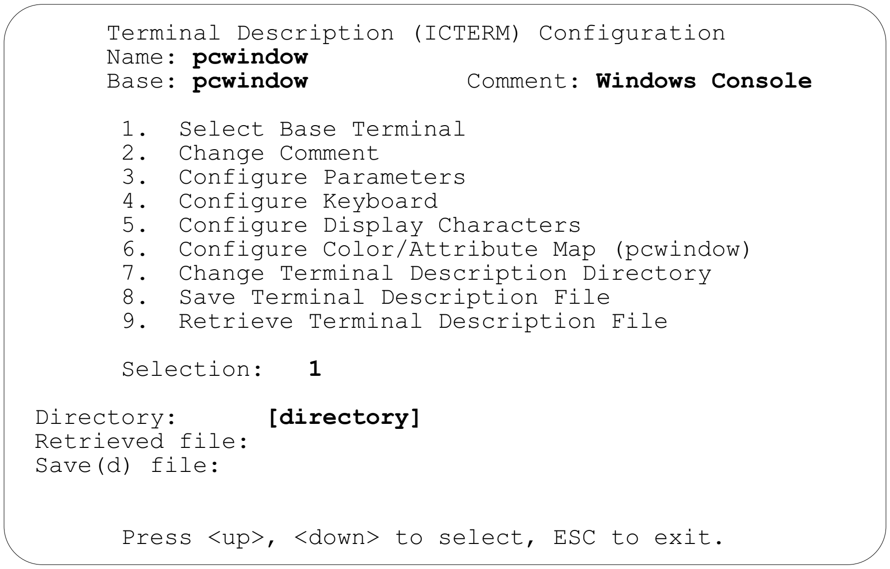
The Reset to Defaults selection allows you to start from scratch and set all new values or just to see the default configuration values. Remember you do not have to Save the results of a work session in ICCONFIG.

D. **Terminal Descriptions (.tdi)**

D.1. Overview

Option 2 from the MAIN MENU will display SCREEN 26. This menu provides the ability to configure different terminal types for ICTERM. The terminal descriptions are saved to a file called the terminal description file with a `.tdi' extension. To build a base set of terminal description files, use the menu to select the appropriate base terminal (selection 1) and then use Save (selection 8) to create that default description.

The name of the current base terminal selection is displayed in this screen and the Configure Keyboard and Configure Display screens.



```
      Terminal Description (ICTERM) Configuration
      Name: pcwindow
      Base: pcwindow           Comment: Windows Console

       1.  Select Base Terminal
       2.  Change Comment
       3.  Configure Parameters
       4.  Configure Keyboard
       5.  Configure Display Characters
       6.  Configure Color/Attribute Map (pcwindow)
       7.  Change Terminal Description Directory
       8.  Save Terminal Description File
       9.  Retrieve Terminal Description File

       Selection:   1

 Directory:      [directory]
 Retrieved file:
 Save(d) file:


       Press <up>, <down> to select, ESC to exit.
```

SCREEN 26. ICCONFIG TERMINAL DESCRIPTION (ICTERM) (.tdi)

To allow a particular terminal description to be available for Interactive COBOL, a terminal description file (.TDI) must be created for that particular ICTERM entry. Each of the base terminal descriptions can be used to create a default description file. This can be done by using option 1 (Select Base Terminal) followed by option 8 (Save terminal description file) for each of the possible base terminal selections.

To change a terminal description file, use option 1 (Select Base Terminal) or option 9 (Retrieve Terminal description file) to choose the description to be changed. Now use options 4 (Configure Keyboard), 3 (Configure Parameters), and/or 5 (Configure Display Characters) to change the needed values and then use option 8 (Save Terminal Description file) to store that terminal description. Option 2 (Change comment) can be used to store a comment about this new description.

To build a new terminal description file use option 1 (Select Base Terminal) or option 9 (Retrieve Terminal description file) to choose a description whose output characteristics match the new terminal. Now use options 4

(Configure Keyboard), 3 (Configure Parameters), and/or 6 (Configure Display Characters) to change the needed values for the particular terminal and then use option 8 (Save Terminal Description file) with a new filename. These tailored versions should generally be saved with a different name and comment to distinguish them from the defaults. Option 2 (Change comment) can be used to store a comment about this new description.

Option 6 (Configure Color/Attribute Map) is only for use with the pcwindow base terminal and allows the Windows master console screen to be setup.

D.2. Select Base Terminal

Option 1 from the TERMINAL DESCRIPTION menu positions to the Base Terminal Description field and allows a base terminal to be selected from a list of available terminals. These can be selected via a ferris wheel using the left-arrow and right-arrow keys. This base set includes the following:

Ansi class:	ANSI, AIX Console, AT&T 605, AT386 Console, SCO Console, SUN Console, XENIX Console, XTERM Console, 386IX Console	
DG class:	DG D200+	
DGUNIX class:	DG D217+ in dgunix mode	
Disk class:	File	
IBM class:	IBM 3151	
Freedom class:	FREEDOM-One	
VTxxx class:	VT100, VT220, VT220PC, Linux	
Wyse 50 class:	WYSE 50	
Wyse class:	WYSE 60	
PC Windows class:	PCWINDOW, PCWINDOWMONO	(On Windows only)
Terminfo class:	TERMINFO	(On UNIX only)

The sets of classes shown above are the default output codes used for each of the particular base terminal descriptions. I.E., all the terminal descriptions in the Ansi class use ansi codes, those in the DG class use DG control codes, etc. . When in the Configure Display screens, these classes are shown at the top as Format.

More information about these base terminal descriptions can be found in the ICTERM Chapter on page 173.

D.3. Change Comment

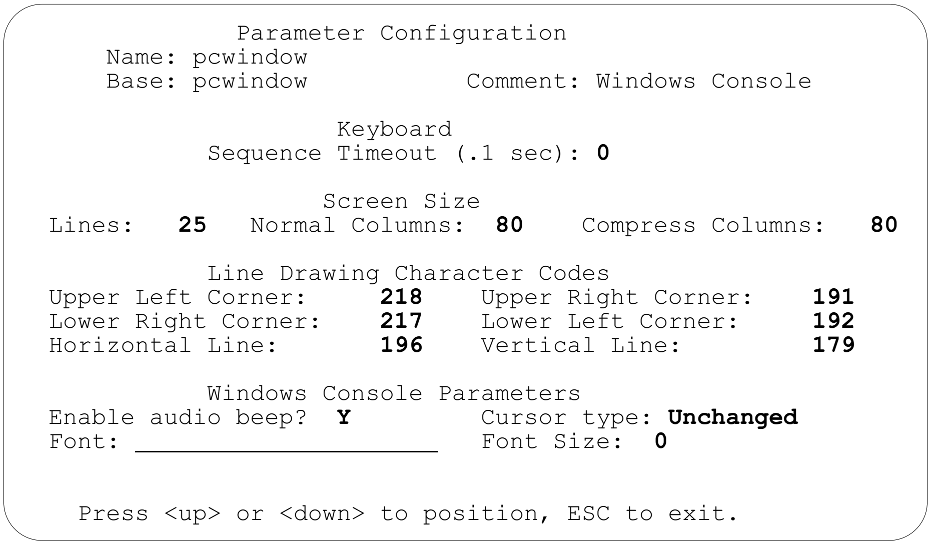
Option 2 from the TERMINAL DESCRIPTION menu can be used to insert a comment into a terminal description file. Selecting option 2 positions to the Comment field allowing up to thirty (30) characters to be entered for this terminal description. This option is especially useful if you change a base terminal description to provide some documentation of the change. For example if you add a HotKey to a DG description you may change to comment to "DG F1 Hot".

D.4. Configure Parameters

Option 3 from the TERMINAL DESCRIPTION menu gives the PARAMETER CONFIGURATION as shown in SCREEN 27.

The current name of the terminal being configured is displayed at the top of the menu along with its format. The format can be any of the valid display classes mentioned previously.

The PARAMETER CONFIGURATION provides the basic defaults for this terminal description for the number of lines and columns and what characters to use for the line drawing character set. The lines and columns values can be overridden by the ICLINES and/or ICCOLUMNS environment entries either in the actual environment or in the PROGRAM ENVIRONMENT screen specified previously in ICCONFIG.



```
              Parameter Configuration
     Name: pcwindow
     Base: pcwindow           Comment: Windows Console

                   Keyboard
          Sequence Timeout (.1 sec): 0

                  Screen Size
Lines:   25   Normal Columns:  80    Compress Columns:   80

          Line Drawing Character Codes
Upper Left Corner:     218    Upper Right Corner:    191
Lower Right Corner:    217    Lower Left Corner:     192
Horizontal Line:       196    Vertical Line:         179

          Windows Console Parameters
Enable audio beep?  Y         Cursor type: Unchanged
Font: _____________________   Font Size:  0


  Press <up> or <down> to position, ESC to exit.
```

SCREEN 27. ICCONFIG PARAMETER CONFIGURATION

The sequence timeout is the number of tenths of seconds to wait for the intercharacter gap between function keys sequences. The default is 0 for pcwindows and 5 for more other cases. Some remote connections, using telnet or rlogin, may need to have this number adjusted upward to correctly handle function key sequences.

The screen size and line drawing codes are NOT available for terminfo type terminals.

The screen size specifies the default rows and columns for this terminal type. If the Compress columns is set to a value other than that set for normal columns then Compressed mode is supported on this terminal.

The line drawing codes are the decimal codes for the ASCII character to be used for each appropriate part of a box. The default values shown are what this terminal supports. If the terminal does not support a real line drawing character set, the default values of 43, 45, and 124 (decimal) which are `+', `-', and `|', are used.

Windows Console Parameters: (Only for pcwindows)

Enable audio beep specifies whether to enable the audio beep. The default is for beep to be enabled. In certain cases when the beep is done too often on the Master Console, it can slow the machine for other users, in those cases, it may be wise to disable the beep.

Cursor type specifies what type of cursor to use when the runtime starts. Valid selections are unchanged, line, or block. Unchanged is the default.

Font specifies the fixed font to use at runtime startup. If not specified, the default is Courier New.

Font size specifies the font size to use at runtime startup. If not specified 11 is used. Valid values are 2 to 99.

D.5. Configure Keyboard

Option 4 from the TERMINAL DESCRIPTION menu presents the screen as shown in SCREEN 28. The KEYBOARD CONFIGURATION menu instructs Interactive COBOL what to do for each possible input sequence from this keyboard.

The current name of the terminal being configured is displayed at the top of the menu along with its format. The format can be either ASCII or terminfo. In ASCII mode, this table instructs Interactive COBOL how to handle each of the entered ASCII sequences on input from the keyboard. In terminfo mode, this table instructs Interactive COBOL how to handle each of the entered terminfo capname sequences on input from the keyboard.

This table can have up to 512 unique input sequences for a particular terminal description.

The first column of the table gives the legend or label of the key and any shift or ctrl key that must be pressed to get this key. Interactive COBOL does not use this column in any way, it is useful only as a label.

Column 2 gives the actual codes (series of 8-bit bytes) that this key generates when pressed on the keyboard. Interactive COBOL normally watches for these code(s) in a timing-sensitive fashion and, if seen, will handle as specified by the next field.

When entering data in column 2 the following can be used:

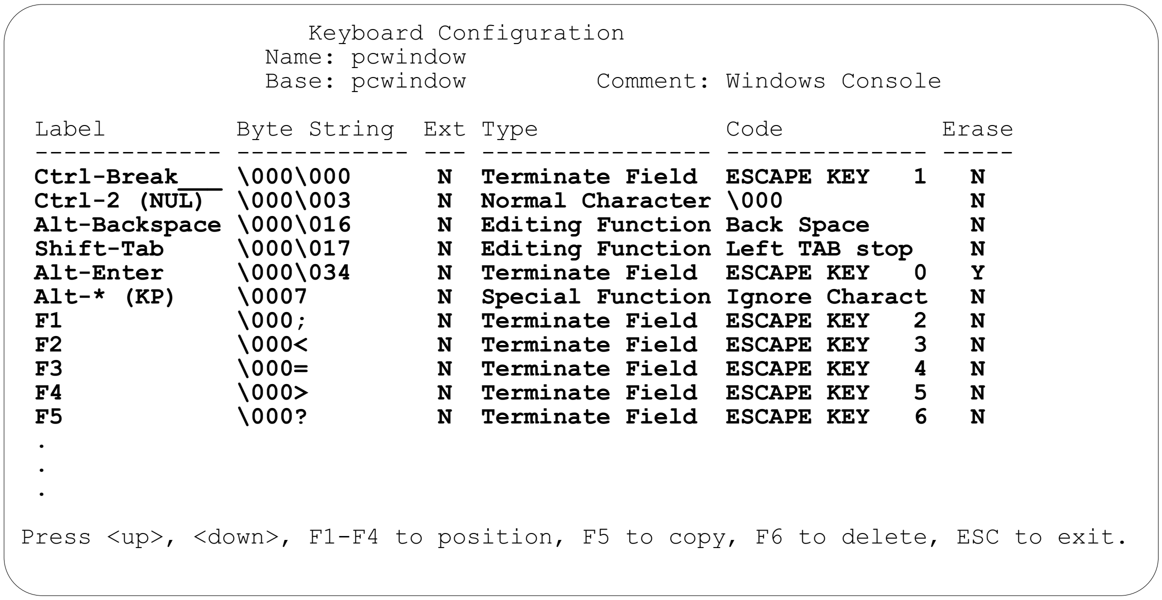
\a enters a bell (Ctrl-G)
\b enters a backspace (Ctrl-H)
\dnnn enters nnn in decimal
\e enters an ESC
\f enters a form-feed (Ctrl-L)
\k allows a terminfo Capname to be entered when using terminfo
\n enters a linefeed (Ctrl-J)

\r enters a carriage-return (Ctrl-M)
\t enters a tab (Ctrl-I)
\v enters a vertical tab (Ctrl-K)
\xnn enters nn in hex
\\\\, \\^, \\", \\', and \\? enters a single `\', `^', `"', `'', and `?' respectively
^@ through _ enters the control code (\\000 - \\037)
\nnn enters nnn in octal
any printable character (`!' - `~') is entered as itself

When displayed on the screen or in the listing, all printable characters (`!' - `~') will be displayed as is, while all others will be shown in octal.

The next columns allow the type to be entered; within that type, how to interpret the key (code), and for next, previous, and terminating types whether to erase to the right of the cursor position. Right-arrow or left-arrow is then used to cycle forward or backward through the valid choices through these ferris-wheel fields.

The up-arrow, down-arrow, F1 (left), F2 (right), F3 (page-up), and F4 (page-down) keys will position to the field to change.



```
                       Keyboard Configuration
                    Name: pcwindow
                    Base: pcwindow         Comment: Windows Console

 Label         Byte String  Ext Type             Code           Erase
 ------------- ------------ --- ---------------- -------------- -----
 Ctrl-Break___ \000\000      N  Terminate Field  ESCAPE KEY   1   N
 Ctrl-2 (NUL)  \000\003      N  Normal Character \000             N
 Alt-Backspace \000\016      N  Editing Function Back Space       N
 Shift-Tab     \000\017      N  Editing Function Left TAB stop    N
 Alt-Enter     \000\034      N  Terminate Field  ESCAPE KEY   0   Y
 Alt-* (KP)    \0007         N  Special Function Ignore Charact   N
 F1            \000;         N  Terminate Field  ESCAPE KEY   2   N
 F2            \000<         N  Terminate Field  ESCAPE KEY   3   N
 F3            \000=         N  Terminate Field  ESCAPE KEY   4   N
 F4            \000>         N  Terminate Field  ESCAPE KEY   5   N
 F5            \000?         N  Terminate Field  ESCAPE KEY   6   N
 .
 .
 .

Press <up>, <down>, F1-F4 to position, F5 to copy, F6 to delete, ESC to exit.
```

SCREEN 28. ICCONFIG KEYBOARD CONFIGURATION

This table is always stored in sorted order based on the input sequence defined in column 2.

Valid Types with their Codes are defined in the following paragraphs. The numbers in parentheses after the Types and Editing codes are the values returned in the IC_GET_KEY builtin function for the appropriate keystroke.

Normal Character (1) - The runtime system will treat keycodes of this type as normal 8-bit ASCII characters. The

Code column contains the actual character code to be used by the runtime system. The value can be 0 to 255 (decimal).

Editing Function (2) - This type of keycode instructs the runtime system to take the action as described in the Code column. The possible actions in the Code column are:

- clear to end-of-field (7)
- end of field (9)
- left word (11)
- left TAB stop (13)
- left a character (1)
- backspace (3)
- insert mode ON/OFF (5)
- sound bell (15)
- beginning of field (8)
- right word (10)
- destructive TAB (12)
- right TAB stop (14)
- right a character (2)
- delete a character (4)
- clear field (6)
- back delete (16)

TAB settings are set at every fourth character position from the beginning of the field, i.e., 1, 5, 9, ...

Terminate Field (3) - This type of keycode causes the runtime system to accept the current field and set the ESCAPE KEY value to the value given in the Code column. The ESCAPE KEY value can be 00 to 99, but remember that value 99 is used for timeouts.

Previous Field (4) - This type of keycode instructs the runtime system to move to the previous field in a screen. If the current field is not the first field in a screen, the field is accepted and the screen is positioned to the preceding field. If the current field is the first (or only) field in a screen, the result depends on the ESCAPE KEY value associated with the key given in the Code column. If the ESCAPE KEY value is 00 (default), the system will beep, and the screen will remain positioned at the first field. If the ESCAPE KEY value is not 00, the field will be accepted, and the screen will exit with the specified ESCAPE KEY value. The ESCAPE KEY value can be 00 to 99, but remember that the system returns 99 for timeouts.

Next Field (5) - This type of keycode instructs the runtime system to move to the next field in a screen. If the current field is not the last field in a screen, the field is accepted and the screen is positioned to the next field. If the current field is the last (or only) field in a screen the field will be accepted and the screen will exit with the specified ESCAPE KEY value given in the Code column. When using the default value, it will act just like a newline had been hit. The ESCAPE KEY value can be 00 to 99, but remember that the system returns 99 for timeouts.

Use of the *Previous Field* and *Next Field* functions on the up-arrow and down-arrow keys with unique ESCAPE KEY values will allow applications to tie several screens together and control the flow from one screen to the other. The default setting of up-arrow is Previous field-ESCAPE KEY 00. The default setting of down-arrow is Next field-ESCAPE KEY 00.

Hot Key Function (6) - Allows for a particular hotkey program to be called whenever this key is entered. The code column allows a value from 00 to 99 to be set such that a COBOL CALL "hotkey<nn>" will be called with the given value replacing the <nn>. There must be a COBOL program available and executable with this name or else a beep will be given.

Previous Row (7) - This type of key code instructs the runtime system to move to the "best fitting" field on a previous row in a screen. If the current field is not in the topmost row of the screen, it is accepted and the cursor is positioned to the "best fitting" field. If the current field is in the topmost row of the screen, the result depends on the ESCAPE KEY value associated with the key in Code Column in ICCONFIG. If the ESCAPE KEY value is 00, the screen will remain positioned in the current field and the bell will sound. If the ESCAPE KEY value is not 00, the field will be accepted and the screen will exit with the specified ESCAPE KEY value. The "best fitting" field is defined to be a field in a preceding row which has the same column position (1st choice), a higher column position (2nd choice) or a lower column position (last choice) than the current field. In any case, the field selected will be the first screen row proceeding the current one which contains ANY fields.

Next Row (8) - This type of key code instructs the runtime system to move to the "best fitting" field on a subsequent row in a screen. If the current field is not in the bottommost row of the screen, it is accepted and the cursor is positioned to the "best fitting" field. If the current field is in the bottommost row of the screen, the result depends on the ESCAPE KEY value associated with the key in Code Column in ICCONFIG. If the ESCAPE KEY value is 00, the screen will remain positioned in the current field and the bell will sound. If the ESCAPE KEY value is not 00, the field will be accepted and the screen will exit with the specified ESCAPE KEY value. The "best fitting" field is defined to be a field in a subsequent row which has the same column position (1st choice), a lower column position (2nd choice) or a higher column position (last choice) than the current field. In any case, the field selected will be the first screen row following the current one which contains ANY fields.

Special Function (0) - Is a set of special internal actions to be taken by the runtime upon receipt of this keystroke. *Special Function* keys do not return in IC_GET_KEY. The actions are defined by the following Codes:

> *Illegal Character* - The runtime system will beep when it receives a keystroke of this type.
>
> *Ignored Character* - The runtime system ignores keycodes of this type.
>
> *Refresh Screen* - The runtime system will clear the current screen and totally refresh the screen from its internal image.
>
> *Enter minus* - This runtime system enters a minus character key followed by a *Terminate Field* with an ESCAPE KEY 0, as two separate keystrokes.

The Erase column is only valid for *Terminate Field*, *Next Field*, *Previous Field*, *Previous Row*, and *Next Row* types. If Erase is set to No (the default), the runtime accepts the current field as currently displayed. If set to Yes, all characters to the right of the cursor in the current field are discarded. It is equivalent to first entering the clear to end-of-field key followed by the same Terminate, Next, or Previous key without the Erase option.

When configuring for UNIX and using the terminfo base setting, Terminfo Capname codes should be entered by preceding the Capname with a backslash (\). For example, *\kcud1* would be entered for cursor down.

The Ext column is the timing-insensitive option and can be used to configure multi-character keystroke sequences for those terminals that do not support the needed number of function keys. The lead-in character for a timing-insensitive sequence must not have been previously defined as a timing-sensitive character, otherwise an error is given by ICCONFIG. An example of how timing-insensitive keys can be entered is given below:

> Let's say you want the Ctrl-R character to be the lead-in character for your timing-insensitive codes and you wish to use Ctrl-R followed by the `1' key for function key F1, followed by a `2' for F2, and so on up to `9', Ctrl-R followed by an `a' for F10, `b' for F11, . . . , `f' for F15 and the shifted states of the second character to get the shifted function keys. I.E., Ctrl-R followed by `!' for Shift-F1, and Ctrl-R followed by `A' for Shift-F10 etc. .
>
> Now, to allow the Ctrl-R character to be the lead-in for the functions keys F1-F15 in the normal and shift states and allow the Ctrl-R Ctrl-R keystrokes to be interpreted as a single Ctrl-R, change the default Ctrl-R Byte String entry from "\022" to "\022\022" with the Ext column set to Y.
>
> Now for each needed function key add the appropriate line. For the first case it would be:
>
> In column 1 (Label), give an appropriate label:
> ```
> MyF1
> ```
>
> In column 2 (Byte String), give the string:
> ```
> \0221
> ```
>
> In column 3 (Ext), enable the key:
> ```
> Y
> ```
>
> In column 4 (Type), give the type code as:
> ```
> Terminate Field
> ```

In column 5 (Code), give the appropriate function key code to be returned:

```
ESCAPE 2
```

In column 6 (Erase), give the appropriate value on whether to erase to the right of the cursor:

```
N
```

Now continue these responses row by row to add the needed keys as given below:

\0222	to generate F2
.	.
\0229	to generate F9
\022a	to generate F10
.	.
\022f	to generate F15
\022!	to generate Shift-F1
.	.
\022(	to generate Shift-F9
\022A	to generate Shift-F10
.	.
\022F	to generate Shift-F15

Additional timing-insensitive keys can be added by following the above example.

D.6. Configure Display Characters

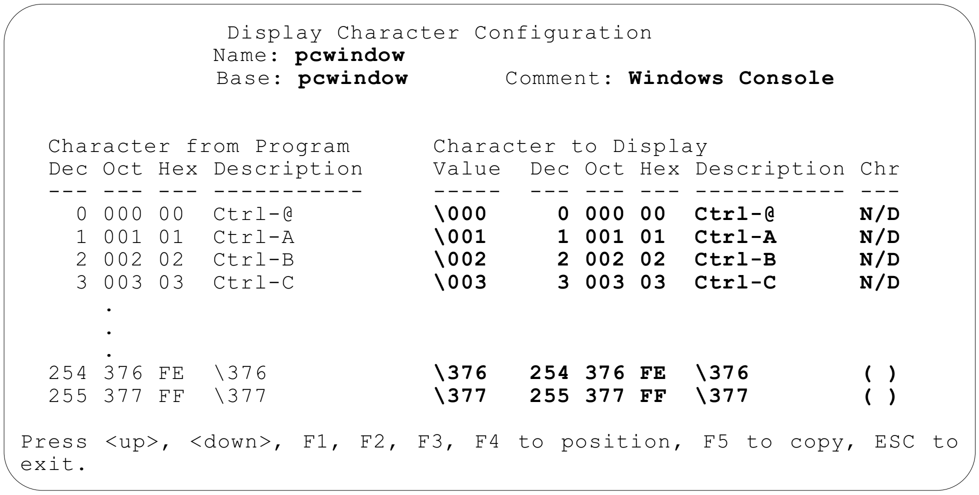
Option 5 from the TERMINAL DESCRIPTION menu presents the screen as shown in SCREEN 29. The DISPLAY CHARACTER CONFIGURATION instructs Interactive COBOL what to do for each possible output character from the COBOL program to the terminal. This option is not available for terminfo.

The current name of the terminal being configured is displayed at the top of the menu along with its format.

This table provides entries for all 256 possible sequences with each line representing an entry as two major columns: Character from program and Character to Display.

The first major column, Character from Program, gives the character from the program to be output to the terminal in Decimal, Octal, Hex, and as a Description.

The second major column, Character to Display, give a Value column that allows entry of the value to be displayed along with 5 additional sub-columns showing the Decimal, Octal, Hex, Description, and what this output character looks like on this terminal.



```
               Display Character Configuration
              Name: pcwindow
              Base: pcwindow       Comment: Windows Console


  Character from Program      Character to Display
  Dec Oct Hex Description     Value  Dec Oct Hex Description Chr
  --- --- --- -----------     -----  --- --- --- ----------- ---
    0 000 00  Ctrl-@          \000     0 000 00  Ctrl-@      N/D
    1 001 01  Ctrl-A          \001     1 001 01  Ctrl-A      N/D
    2 002 02  Ctrl-B          \002     2 002 02  Ctrl-B      N/D
    3 003 03  Ctrl-C          \003     3 003 03  Ctrl-C      N/D
      .
      .
      .
  254 376 FE  \376            \376   254 376 FE  \376        ( )
  255 377 FF  \377            \377   255 377 FF  \377        ( )

Press <up>, <down>, F1, F2, F3, F4 to position, F5 to copy, ESC to
exit.
```

SCREEN 29. ICCONFIG DISPLAY CHARACTER CONFIGURATION

When entering data in the Values column the following can be used:

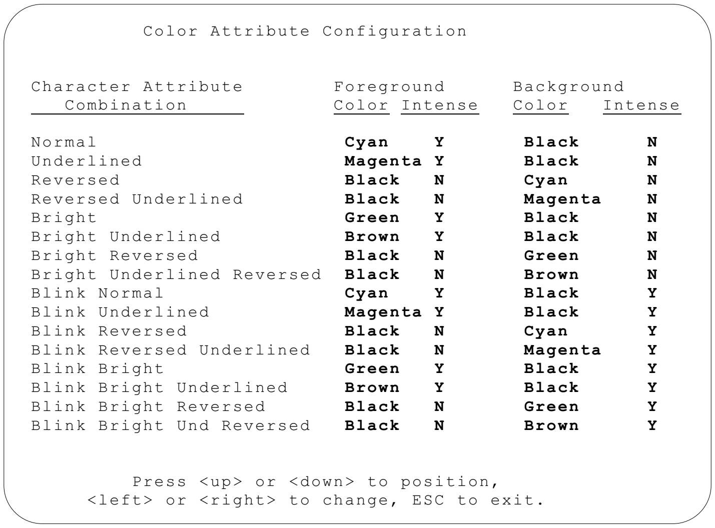
\a enters a bell (Ctrl-G)
\b enters a backspace (Ctrl-H)
\dnnn enters nnn in decimal
\e enters an ESC
\f enters a form-feed (Ctrl-L)
\n enters a linefeed (Ctrl-J)
\r enters a carriage-return (Ctrl-M)

\t enters a tab (Ctrl-I)
\v enters a vertical tab (Ctrl-K)
\xnn enters nn in hex
\\\\, \\^, \\″, \\′, and \*?* enters a single `\', `^', `"', `'', and `?' respectively
^@ through _ enters the control code (\000 - \037)
\nnn enters nnn in octal
any printable character (`!' - `~') is entered as itself

When displayed in the Chr column, all non-control code characters (both 7- and 8-bit) will be displayed as defined within parenthesis ( ), while the control code characters will show a N/D (Not Displayable) in that column.

D.7. Configure Color / Attribute Map (pcwindow)

Option 6 from the TERMINAL DESCRIPTION menu can be used to set the color and attribute mapping for the pcwindow type terminals. This screen allows changing the mappings of the displayable attributes. For each of the character attribute combinations, the foreground color and intensity, and the background color and intensity be changed. Up to eight colors can be selected. If the color environment (ICCOLOR) has been set to process, these attribute-to-color defaults are NOT used. The actual character attribute is sent to the monitor unchanged.



```
            Color Attribute Configuration


  Character Attribute          Foreground      Background
     Combination               Color Intense   Color   Intense

  Normal                        Cyan     Y       Black      N
  Underlined                    Magenta Y        Black      N
  Reversed                      Black    N       Cyan       N
  Reversed Underlined           Black    N       Magenta    N
  Bright                        Green    Y       Black      N
  Bright Underlined             Brown    Y       Black      N
  Bright Reversed               Black    N       Green      N
  Bright Underlined Reversed    Black    N       Brown      N
  Blink Normal                  Cyan     Y       Black      Y
  Blink Underlined              Magenta Y        Black      Y
  Blink Reversed                Black    N       Cyan       Y
  Blink Reversed Underlined     Black    N       Magenta    Y
  Blink Bright                  Green    Y       Black      Y
  Blink Bright Underlined       Brown    Y       Black      Y
  Blink Bright Reversed         Black    N       Green      Y
  Blink Bright Und Reversed     Black    N       Brown      Y


            Press <up> or <down> to position,
        <left> or <right> to change, ESC to exit.
```

SCREEN 30. ICCONFIG COLOR ATTRIBUTE MAP

D.8. Change Terminal Description Directory

Option 7 from the TERMINAL DESCRIPTION menu can be used to change the default directory from which terminal description file are saved to or retrieved from. The default is set to the term subdirectory if the ICROOT environment parameter was set or else the current directory.

D.9. Save and Retrieve Terminal Description File

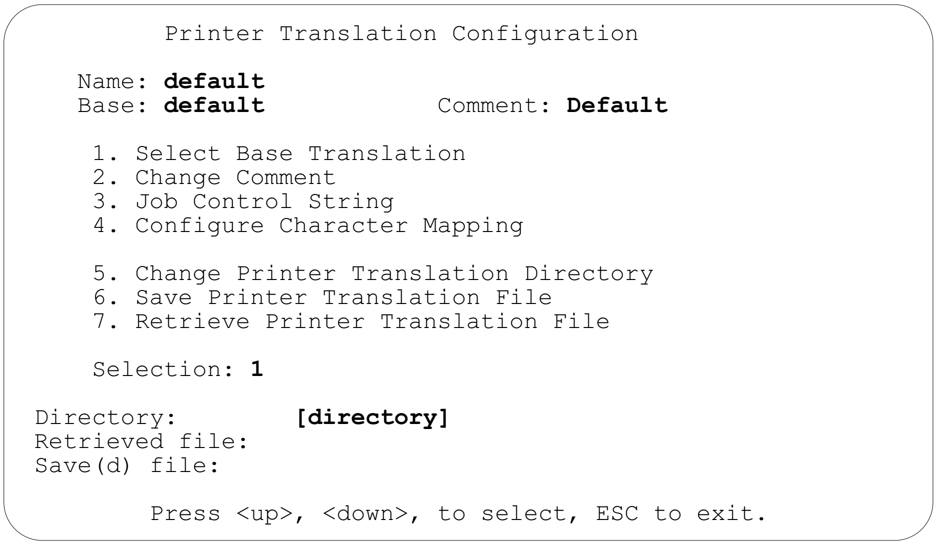
Options 8 and 9 from the TERMINAL DESCRIPTION menu, allow for terminal description file(s) to be saved or retrieved with the `.tdi' extension. The Save(d) file and Retrieved file fields are used to prompt for the filename to be saved or retrieved respectively.

E. Printer Translations (.pti)

E.1. Overview

Option 3 from the MAIN MENU will display SCREEN 31. This menu provides the ability to configure different printer translations files for use when printing through the @PRN's or @PCQ's. Printer translations are saved to files called printer translation files with a `.pti' extension. To build a base set of printer translation files, use the menu to select the appropriate base translation (selection 1) and then use save (selection 7) to create that default translation. The default translation should never be saved and used as that is just a one to one mapping of characters which will be done automatically if no printer translation is specified. On UNIX, when using the UNIX spooler, the UNIX spooler should be configured to use the appropriate filter to perform the printer translation instead of using Interactive COBOL printer translation facility.

The name of the current base printer selection is displayed in this screen and the Configure Character Mapping screen.



```
          Printer Translation Configuration

    Name: default
    Base: default            Comment: Default

     1. Select Base Translation
     2. Change Comment
     3. Job Control String
     4. Configure Character Mapping

     5. Change Printer Translation Directory
     6. Save Printer Translation File
     7. Retrieve Printer Translation File

     Selection: 1

 Directory:        [directory]
 Retrieved file:
 Save(d) file:

         Press <up>, <down>, to select, ESC to exit.
```

SCREEN 31. ICCONFIG PRINTER TRANSLATION (.pti)

To allow a particular printer translation to be available for Interactive COBOL, a printer translation file (.PTI) must be created for that particular entry. Each of the base printer translations can be used to create a default file. This can be done by using option 1 (Select Base Translation) followed by option 6 (Save) for each of the possible base selections although the default one to one map should never be saved. It should only be used as a starting point to make changes for a new printer translation file.

To change a printer translation file, use option 1 (Select Base Translation) or option 7 (Retrieve) to choose the translation to be changed. Now use option 4 (Configure Character Mapping) to change the needed values and then use option 6 (Save) to store that printer translation. Option 2 (Change comment) can be used to store a comment about this new translation.

To build a new printer translation file, use option 1 (Select Base Translation) or option 6 (Retrieve) to choose a translation whose output characteristics most closely match the new translation. Now use option 4 (Configure Character Mapping) to change the needed values for the particular translation and then use option 6 (Save) with a new filename. These tailored versions should generally be saved with a different name and comment to distinguish them from the defaults. Option 2 (Change comment) can be used to store a comment about this new translation.

E.2. Select Base Translation

Option 1 from the PRINTER TRANSLATION menu positions to the Base Printer Translation field and allows a base translation to be selected from a list of available translations. These can be selected via a ferris wheel using the left-arrow and right-arrow keys. This base set includes the following:

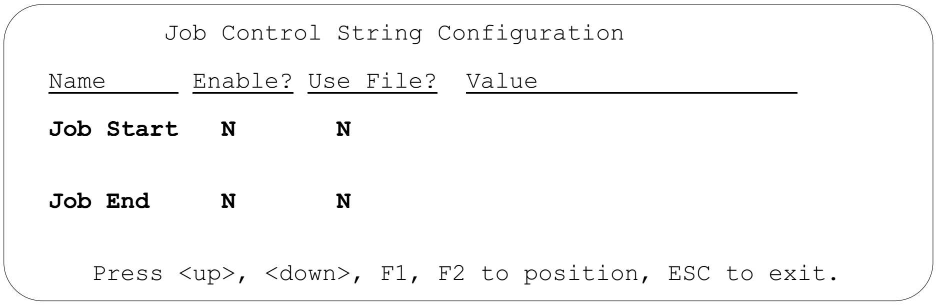
Default:	basic one to one mapping
DG to PC:	maps the DGI character set to the standard IBM PC character set
PC to DG	maps the standard IBM PC character set to the DGI character set

E.3. Change Comment

Option 2 from the PRINTER TRANSLATION menu can be used to insert a comment into a printer translation file. Selecting option 2 positions to the Comment field allowing up to thirty (30) characters to be entered for this printer translation. This option is especially useful if you change a base printer translation to provide some documentation of the change.

E.4. Job Control String

Option 3 from the PRINTER TRANSLATION menu can be used to insert job control strings at the beginning and/or end of a print file. Each can be separately set and can either be a series of values or a filename that contains the codes to be inserted.



```
        Job Control String Configuration

  Name      Enable? Use File?  Value

  Job Start   N       N


  Job End     N       N

     Press <up>, <down>, F1, F2 to position, ESC to exit.
```

SCREEN 32. ICCONFIG PRINTER JOB CONTROL STRING CONFIGURATION

E.5. Configure Character Mapping

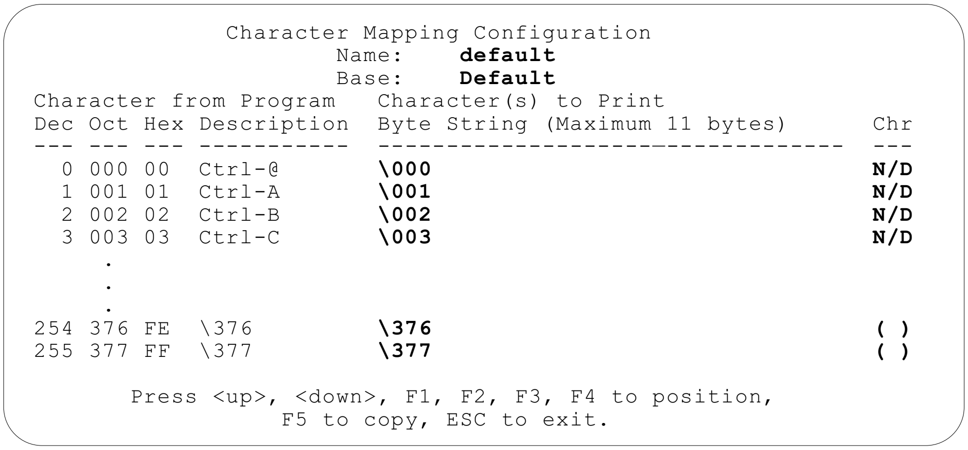
Option 4 from the PRINTER TRANSLATION menu presents SCREEN 33. The CHARACTER MAPPING CONFIGURATION instructs Interactive COBOL what to do for each possible output character from the COBOL program to the printer.

The current name of the printer being configured is displayed at the top of the menu.

This table provides entries for all 256 possible sequences with each line representing an entry as two major columns: Character from program and Character(s) to Print.

The first major column, Character from Program, gives the character from the program to be output to the printer in Decimal, Octal, Hex, and as a Description.

The second major column, Character(s) to Print, gives a Value column that allows entry of the value to be displayed along with an additional sub-column showing what this output character looks like on this terminal. Up to 11 bytes are allowed.



```
                 Character Mapping Configuration
                         Name:    default
                         Base:    Default
Character from Program   Character(s) to Print
Dec Oct Hex Description  Byte String (Maximum 11 bytes)      Chr
--- --- --- -----------  --------------------------------    ---
  0 000 00  Ctrl-@       \000                                N/D
  1 001 01  Ctrl-A       \001                                N/D
  2 002 02  Ctrl-B       \002                                N/D
  3 003 03  Ctrl-C       \003                                N/D
      .
      .
      .
254 376 FE  \376         \376                                ( )
255 377 FF  \377         \377                                ( )

        Press <up>, <down>, F1, F2, F3, F4 to position,
                    F5 to copy, ESC to exit.
```

SCREEN 33. ICCONFIG CHARACTER MAPPING CONFIGURATION

When entering data in the Values column the following can be used:

\a enters a bell (Ctrl-G)
\b enters a backspace (Ctrl-H)
\dnnn enters nnn in decimal
\e enters an ESC
\f enters a form-feed (Ctrl-L)
\n enters a linefeed (Ctrl-J)
\r enters a carriage-return (Ctrl-M)

\t enters a tab (Ctrl-I)
\v enters a vertical tab (Ctrl-K)
\xnn enters nn in hex
\\, \^, \", \', and \? enters a single `\', `^', `"', `'', and `?' respectively
^@ through _ enters the control code (\000 - \037)
\nnn enters nnn in octal
any printable character (`!' - `~') is entered as itself

When displayed in the Chr column, all non-control code characters (both 7- and 8-bit) will be displayed as defined within parenthesis ( ), while the control code characters will show a N/D (Not Displayable) in that column.

E.6. Change Printer Translation Directory

Option 5 from the PRINTER TRANSLATION menu can be used to change the default directory from which printer translation files are saved to or retrieved from. The default is set to the print subdirectory if the ICROOT environment parameter was set or else the current directory.

E.7. Save and Retrieve Printer Translation File

Options 6 and 7 from the PRINTER TRANSLATION menu, allow for printer translation file(s) to be saved or retrieved with the `.pti' extension. The Save(d) file and Retrieved file fields are used to prompt for the filename to be saved or retrieved respectively.

F. Exit ICCONFIG

Pressing ESC while in the MAIN MENU terminates ICCONFIG and returns you to the program from which you called ICCONFIG. If ICCONFIG detects any modification since a Save or a Retrieve, it will prompt with a message asking if you wish to save your changes. If you type Y, a save is done before exiting; otherwise, your changes are discarded.

G. Batch Update Facility (.cfi)

ICCONFIG supports a batch update facility. The -l|-L (Load .ini) switches are available to load a .ini file into the current (or new configuration). This facility uses standard .ini file processing to allow for non-interactive changing of the configuration file.

The -l|-L file|dir (Load .ini), instructs ICCONFIG to use the .ini file specified to change the current configuration. Only those sections and line(s) that are being changed need to be present in the .ini file to be loaded. If a change is made, ICCONFIG will display that the file was updated. If no changes were made, no message will be displayed and the configuration file will not be updated. Only valid lines are read from the .ini file. The Load switch requires the Batch switch (-b).

The ICCONFIG .ini file processing provides the ability to specify an entire configuration file (.cfi). No support is provided for terminal description files or printer description files.

Rules for processing the .ini file:

1. Section names (names inside of [ ]) and keywords (names on the left of the =) are case sensitive.

2. Entries are not read from the .ini file in the order they appear in the .ini file. ICCONFIG reads sections and keywords within a section in a predetermined order (the order does match a generated .ini file).

3. Invalid entries (either sections or keywords) are never read.

4. If more than one matching keyword in a section, only the first one is read.

5. Values for keywords are checked for correctness. An error is given with the keyword and section names. A special value <empty> is used/provided to denote no value for a keyword.

6. Except for trailing spaces on a line, spaces are significant.

7. Comment lines can be entered in the .ini file with the ";" character.

8. Valid section names are shown in a default .cfi file. By looking at a main section and looking at its default entries values you can see all the valid entries for that section.

So to perform a batch update, one would start with a valid .cfi file and pick the section(s) needed and copy them to the appropriate .ini file to be used for the batch update.

VII. ICEXEC

A. Introduction

ICEXEC is a service that coordinates execution of Interactive COBOL processes and configures global structures on a single machine. ICEXEC is required to allow for unique terminal numbering (on this machine), support for dumb terminals (or pc's with emulators) connected to serial ports to execute the **ICOBOL** runtime, and to enable telnet and rlogin sessions to execute the **ICOBOL** runtime. The ICEXEC utility must be running before any Interactive COBOL runtime (ICRUN, ICRUNW, ICRUNCGI), ICNETD surrogate (ICIOS, ICTHINS, ICRUNRS), or ICSMVIEW is allowed to execute. All other Interactive COBOL executables (ICOBOL, ICCHECK, ICPACK, etc.) can be run with or without ICEXEC.

B. Syntax

The syntax is:

```
icexec [-a[:aflag] | -A file|dir[:aflag]] [-C file|dir] [-h|-?] [-O c|s|t]
    [-P file|dir] [-q] [-s] [-t] [-T min:max]
```

Where

-a[:*aflag*] or -A *file|dir*[:*aflag*] (Audit)

Enables auditing (default icexec.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|dir* (Configuration file)

Specifies the configuration file. If only a directory is specified, the filename system.cfi is appended, otherwise the given filename is used. If no configuration file is given, then system.cfi in the current directory is used.

-h | -? (Help)

Display help text.

-O c|s|t (Operation)

Specifies an operation to perform. Valid operations are:

c - check if running,
s - start, and
t - terminate.

-P *file|dir* (Printer Queue location)

Specifies the printer queue file. If only a directory is specified, the filename system.pq is appended, otherwise the given filename is used. If no file is given, then system.pq in the current directory is used.

-q (Quiet)

Enables quiet operation.

-s (Service)

Tells ICEXEC that it is running as a service. Required when “localsystem” is not the starting account.

-t (Trace)

Enables tracing. Useful for debugging. If tracing is selected, then any runtime started by ICEXEC will have an Audit file set to icrun_CON*xx*.lg where *xx* is the console number.

-T *min*:*max* (Terminal range)

Specifies a range of consoles to use from the configuration file. If not specified, all enabled consoles are enabled.

ICEXEC 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, ICEXEC is installed as a service using the ICSVCMGR utility. More on ICSVCMGR can be found on page 49. ICSVCMGR can be used to change the default command line if needed.

C. Using ICEXEC for Multi-User Operation

ICEXEC is used to provide support for serial lines to have a logon and run COBOL programs. In addition, it provides support for telnet/rlogin sessions to have COBOL sessions.

ICCONFIG and ICEDCFW have been updated in the following fashion to support multi-user operation.

A. Under System Parameters. The Number of processes allowed can now be configured. ICEXEC uses this number as the maximum number of connections to the shared area to be allowed. This includes runtimes (icrun) and icnet surrogates (icios, icthins, icrunrs). (This parameter is ignored on Windows 9x.)

B. Under Logical Devices. The Console Device, Printer Device, and Serial Device configuration menus have a new section added for line characteristics. The other portions of these menus have been shrunk in order to accommodate these new settings. These new settings allow for Baud Rate, Date Bits, Parity, Stop Bits, Modem Control, Hardware Output and Input Flow Control, and Software Output and Input Flow control to be set. These line characteristics are ignored for master console and telnet/rlogin/thinclient sessions.

The runtime requires an ICEXEC to already be running. It does NOT read the configuration file or control the printer control file so it does not support the -C and -P parameters. The standard multi-terminal builtins will show other **ICOBOL** runtime processes that are running on this machine.

Master Console

For the master console, connections use a console with a "con" as the device name and run terminal enabled. If you wish to have more than one runtime running on the master console, multiple consoles must be configured to allow each runtime to have a unique terminal number. When ICRUN is invoked, it will connect to the shared area and if no terminal number (-T n) is given, the next available console with a device of "con" will be selected as the console number. Line characteristics are ignored for master console sessions. ICTERM must be set to “pcwindow”.

Serial ports

To enable serial line terminal support, the appropriate serial device (COM1, COM2, etc.) must be specified in the Logical Console Configuration with the appropriate line characteristics (baud, parity, etc.). In addition, the Program Environment must be set appropriately for the Console line. Especially important is the ICTERM setting. When ICEXEC starts, for serial lines that have been set to run terminals ICEXEC sets the line characteristics and then allows a serial user to specify his/her operating system username and password and ICTERM setting. It then validates the username/password with the operating system, sets the working directory, and starts a runtime session with the appropriate parameters. In addition, for lines with modem control enabled it waits for Carrier Detect (CD) before the initial prompt. Valid serial devices can include any hardware that provides serial driver support on Windows. This can include dumb uart boards, smart boards with Windows drivers, etc. In-house testing is done using standard COM ports and COM ports provided by Comtrol (www.comtrol.com) Rocketport boards.

CGI

cgiCOBOL runtimes use a console with a "cgi" as the device name and run terminal enabled. If you wish to have more than one runtime running on cgi then multiple consoles must be configured to allow each runtime to have a unique terminal number. ICTERM must be set to “file”.

ThinClient

Thinclient runtimes (icrunrs) use a console with either a matching “*machine-name*” or “*ip-address*”, or an “icrunrs” or a (blank) as the device name and run terminal enabled. If you wish to have more than one runtime running

ThinClient then multiple consoles must be configured to allow each runtime to have a unique terminal number.

ThinClient (gui)

Thinclient(gui) runtimes (icthins) use a console with either a matching “*machine-name*” or “*ip-address*”, or an “icthins” or a “nul” as the device name and run terminal enabled. If you wish to have more than one runtime running ThinClient (gui)s then multiple consoles must be configured to allow each runtime to have a unique terminal number.

Telnet connections

For telnet/rlogin connections, two types of connections can be made. Most telnet server packages support an "Advanced mode" and a "Simple mode" of operation. Line characteristics are ignored for telnet sessions.

In "Advanced mode", the telnet server actually interprets codes from the telnet client and sends codes to the telnet client based on a particular known terminal definition (ansi, vt100, etc.). In this mode, the telnet server presents a Windows console mode interface to any Windows application. When running in "Advanced mode", the **ICOBOL** runtime must communicate just as it would with the Windows Master console, i.e., ICTERM=pcwindow. It is the responsibility of the telnet server to translate Windows i/o into the appropriate i/o for the telnet client. This include function keys, attributes, etc. To support telnet connections in this mode of operation the Console device under Logical Console Configuration must be set to "con" and enabled. (This mode looks just like the master console to ICRUN.) Under Program Environments, the run program option must be enabled and all the other program environment settings should be set to a default that is appropriate. The environment strings can be overriden by the environment available when the **ICOBOL** runtime starts. To start the **ICOBOL** runtime, the program ICRUN must be executed. It will connect to the shared area and if no terminal number (-T n) was given, the next available console with a device of "con" will be selected as the console number. Generally in “Advanced mode”, you should try to keep from running GUI-type executables. In most cases the GUI program will start but it will be on a “desktop” that is running on the main console that is “invisible” to the user who started it or to the person on the main console.

In "Simple mode", the telnet server does not interpret any codes from the telnet client and sends the actual codes to the telnet client that it receives from the application. In this mode, its the application's responsibility to interpret and send the correct codes for the telnet client. For the **ICOBOL** runtime, ICTERM must be set to the appropriate type of terminal that the telnet client is supporting (vt100, vt220, dg, ibm, wyse, etc.). ICCONFIG or ICEDCFW can be used to tailor a particular terminal definition (.tdi) file. To support telnet clients in this mode of operation the Console device under Logical Console Configuration must be set to a (blank) device and enabled. Under Program Environments, the run program option must be enabled and all the other program environment settings should be set to a default that is appropriate. The environment strings can be overridden by the environment available when the **ICOBOL** runtime starts. I.E., the user can set his/her own ICTERM etc. from the command prompt before starting ICRUN. To start the **ICOBOL** runtime, the program ICRUN must be executed. It will connect to the shared area and if no terminal number (-T n) was given, the next available console with a (blank) device name will be selected as the console number.

For serial lines and telnet sessions in "simple mode", ICRUN detects that it is not on the master console and uses the ICTERM setting to process terminal input and output.

On Windows at installation time, ICEXEC is automatically installed as a service using the given configuration file and printer control queue. ICEXEC will automatically start at boot time. ICEXEC can be controlled either through the Services menu (Start->Settings->Control Panel (Services)) or from the command prompt with the ICSVCMGR utility. **After making changes to the configuration file, ICEXEC must be stopped and restarted in order for those changes to become effective.**

On startup, ICEXEC reads the configuration file and creates a global area that all subsequent runtimes on that machine use.

For serial lines that have been set to run programs, ICEXEC sets up the appropriate line characteristics and then any needed prompting for Username/Password and terminal type, modem connection etc. and setting the working directory before starting ICRUN with the needed parameters. The working directory is the directory specified at installation time. If the Trace switch (-t) was specified to ICEXEC, then each serial line ICRUN process will be started with an audit file set to "icrun_CON*xx*.lg" where *xx* is the terminal number.

While running, ICEXEC sends any messages to the audit file in the following format:

```
    (date) (time) icexec (pid#):severity:text
```

Where

date is the current date,
time is the current time,
pid# will be the pid on which ICEXEC is running,
severity shows the level of message severity, consisting of PANIC, ERROR, INFO, and WARNING, and
text message contains specific information.

If ICEXEC detects any error conditions while it is running, an appropriate message is sent to the audit file.

If ICNETD I/O surrogates (icios) are used while ICEXEC is running, then logical devices (@PCQ.., @SER.., @PRN...) may be used by remote clients.

Telnet connections have been tested with Georgia Softworks Windows Telnet Server (www.georgiasoftworks.com), Ataman's TCP Remote Logon Services (www.ataman.com), GoodTech Systems Telnet Server (www.goodtechsys.com), and Pragma System's Interaccess TelnetD Server (www.pragmasys.com) products although there are other commonly available telnet servers for Windows from such companies as Seattle Labs (www.seattlelabs.com), MicroSoft (www.microsoft.com), etc. (please tell us if you find more). We have found the Georgia Softworks product to be one of the better and faster packages.

General Telnet issues:

ICCONFIG, ICEDCFW, ICSP2, ICQPRW, and ICREVW will not run in a telnet session in "Simple mode".

DG mode terminals must be run in "Simple mode".

Georgia Softworks issues:

Use the provided telnet client on Windows machines to support the runtime running with ICTERM=pcwindow.

Ataman issues:

In "Simple mode", timeouts do not work on read to the console.

Pragma Systems issues:

Provides good "Advanced mode" support for some terminal types.

Only supports telnet connections, no rlogin.

In Stream mode ("Simple mode"), you must disable echoing input.

In Stream mode, timeouts do not work on read to the console.

D. **Processing**

ICEXEC maintains the global shared area to which all **ICOBOL** processes that deal with files attach. It continuously monitors active processes. If ICEXEC detects that a process terminates unexpectedly for any reason it checks to see if that process had any files opened and if they had been modified. If so, ICEXEC will open the affected file(s) and flush any modified buffers to the disk. If they had not been modified, ICEXEC just clears the shared area information for that file. ICEXEC will log any unexpected error conditions that it detects along with what it has to do to keep the files and shared area in a consistent state. Generally unexpected terminations should be investigated to determine their cause and especially if process traps or core dumps the problems should be reported.

VIII. STARTING ICRUN

A. Introduction

Interactive COBOL on Windows is a product that provides the ability to run Interactive COBOL programs.

This chapter discusses how to start the Interactive COBOL on Windows runtime (ICRUN & ICRUNW).

ICRUN.EXE requires about 1MB of free memory to start. ICRUN.EXE is a console mode executable while ICRUNW.EXE is a GUI mode executable. ICRUNW requires a master console while ICRUN can run on the master console, serial lines, or telnet sessions.

Differences of the GUI version with the console version.

A. The icon displayed with the GUI will not be the command icon (usually set to MSDOS). The icon is set to an IC bitmap.
B. GUI version only runs in a GUI session, **Not** in a telnet session or from a COM line. The console version is required for telnet sessions and COM lines.
C. When started from a command window, 1) the GUI does not inherit the lines and columns from that window; and 2) the GUI returns to the command window immediately since it is running in a new window (exit codes will not normally be seen).
D. GUI version, Alt-Enter key behaves as an Enter, does not go Full Screen.
E. GUI version, Command-line errors generate a Message box.
F. When calling the command processor, the command will be executed in a new window.

The ICRUNW and ICRUNRC executables try to select appropriate Bold and Underline fonts to match the selected font and allow the Bold and Underline screen attributes to be more fully implemented.

The console version should be used when running in a batch stream.

From this point on ICRUN and ICRUNW will be treated the same unless specifically specified.

ICRUN requires that ICEXEC is running.

B. Environment Entries

B.1. Overview

ICRUN searches for specific Interactive COBOL entries in the user's current environment. These entries allow the user to tailor a particular session of Interactive COBOL for a particular application, company, etc. Environment entries can be set in Windows by using the *SET* command or they can be specified in the configuration file using ICCONFIG or ICEDCFW.

Interactive COBOL on Windows environment entries in addition to the common ones of ICROOT, ICTMPDIR, ICPERMIT_MACHINE, and TZ are:

DATAFILE	Generic data file (@DATA)
ICABORT	Enable aborting on global timeout
ICBGCOLOR	Specify the initial background color
ICCODEPATH	PATH for .CX files
ICCOLOR	Specify how to support color
ICCOLUMNS	Columns for terminal
ICDATAPATH	PATH for data files
ICFGCOLOR	Specify the initial foreground color
ICFONT	Specify the font name for the GUI runtime (icrunw)

ICFONTSIZE	Specify the point size for the font for the GUI runtime (icrunw)
ICLINES	Lines for terminal
ICPCQDIR	Directory for spooled print files
ICPCQFILTER	Specify a default Printer Control Utility Filter
ICREVERSE	Specify how to support reverse
ICRUN	Default switches for the runtime
ICRUNDIR	Specifies a default directory
ICRUNLK	Link file
ICSCROPT	SCREEN OPTIMIZER selection
ICSDMODE	SCREEN HANDLER selection
ICTERM	Terminal type
ICTIMEOUT	Specify global timeout for ACCEPT
LISTFILE	Generic list file (@LIST)
PCQ	Generic printer control queue
PRN	Generic printer device
SER	Generic serial device
PTS	Print Pass Thru device
PTSMODE	How to support @PTS
PTSFONT	PTS font
PTSFONTSIZE	PTS fontsize

B.2. DATAFILE

DATAFILE specifies a filename for the @DATA file.

The syntax is:

```
DATAFILE=filename
```

Where

filename

Specifies a filename to be used to replace @DATA in a COBOL OPEN.

B.3. ICABORT

ICABORT specifies whether to abort the Interactive COBOL process if a global timeout (ICTIMEOUT) occurs.

The syntax is:

```
ICABORT=on|off
```

If not specified in the environment, then the ICABORT from the configuration file is used.

If ICTIMEOUT is not set or has been overridden by an ACCEPT with TIME-OUT statement or the IC_SET_TIMEOUT builtin, the abort is ignored.

B.4. ICBGCOLOR, ICCOLOR, ICFGCOLOR

ICCOLOR instructs Interactive COBOL how to interpret color codes from a COBOL program. Valid selections are filter, ignore, and process, the default is filter.

The syntax is:

```
ICCOLOR=filter|ignore|process
```

Where

filter
: Causes the runtime to watch for color codes from the program and to NOT send them to the terminal, since it does not support color. Filter is the default.

ignore
: Tells the runtime that the user wants total control of the screen and may be sending binary color data to the screen and that the runtime should ignore all color codes (i.e., do not look for color codes). If running in this mode the SCREEN OPTIMIZER should not be enabled as the runtime cannot repaint a user's screen.

process
: The runtime interprets color codes from the program and sends the appropriate sequences to the terminal. When set to Process the initial background and foreground colors are set by the runtime at startup.

ICBGCOLOR sets the default background color to the indicated value when running with ICCOLOR set to Process. Valid selections are black (0), blue (1), green (2), cyan (3), red (4), magenta (5), brown (6), and white (7) either as the name or the number. The default is black (0).

The syntax is:

```
ICBGCOLOR=black|blue|green|cyan|red|magenta|brown|white|
          0|1|2|3|4|5|6|7
```

ICFGCOLOR sets the default foreground color to the indicated value when running with ICCOLOR set to Process. Valid selections are black (0), blue (1), green (2), cyan (3), red (4), magenta (5), brown (6), and white (7) either as the name or the number. The default is white (7).

The syntax is:

```
ICFGCOLOR=black|blue|green|cyan|red|magenta|brown|white|
          0|1|2|3|4|5|6|7
```

If not specified in the environment then the appropriate entry from the configuration file is used.

Currently only DG and ANSI based terminals support color.

B.5. ICCODEPATH

ICCODEPATH specifies directories or library files (up to 16) that ICRUN will search for a COBOL program (.CX) file when a simple program name is encountered in a CALL, CALL PROGRAM, or from the command line..

The syntax is:

```
ICCODEPATH=dir|file[;dir|file]...
```

Where

dir
: Specifies a directory in which ICRUN should look for the .CX files of programs with simple names.

file
: Specifies a COBOL library file in which ICRUN should look for the .CX files of programs with simple names. Library files are built by the ICLIB utility.

If not specified in the environment, then the ICCODEPATH entry from the configuration file is used. If neither is specified, the default ICCODEPATH as stored in the registry upon Interactive COBOL installation is used.

If ICCODEPATH is specified, the current directory is searched last by default and must be specified by an emptyentry or a period in ICCODEPATH if you wish it to be searched sooner. The list is searched for programs

(.CX files) in the order given in ICCODEPATH.

ICCODEPATH must only include directory names or library names.

The ICCODEPATH only applies to a program with a simple name. The syntax is the same as for a system PATH.

A sample entry is:

```
    ICCODEPATH=.;D:\PGMS;C:\icobol.cl;C:\MAIN\PGMS;E:;c:\program files\icobol
```

which searches the current directory (current drive), D:\PGMS, the library file C:\icobol.cl, C:\MAIN\PGMS, E: (current directory), and finally the installation direction for a particular .CX file.

B.6. ICCOLUMNS, ICLINES, and ICTERM

ICTERM tells Interactive COBOL what type of terminal is attached to a particular line. ICLINES and ICCOLUMNS specify the number of lines and columns for the terminal.

The syntax is:

```
    ICTERM=terminal-type
    ICLINES=lines
    ICCOLUMNS=columns[:ccolumns]
```

Where

terminal-type

Specifies a valid ICTERM entries described below or in the ICTERM Chapter starting on page 173, with a corresponding terminal description file

lines

Specifies the number of lines for this terminal. It can range from 24 to 255.

columns[:*ccolumns*]

Specifies the number of columns for this terminal. It can range from 80 to 255. If the second value is given, then both normal and compressed mode are supported. The larger value is the compressed number of columns. The first value given will be the mode that the screen starts in.

If not specified in the environment, then the ICTERM from the configuration file is used.

If ICLINES and/or ICCOLUMNS are not specified, or are 0, then that specified in the configuration file is used if non-zero. If ICLINES and/or ICCOLUMNS are zero in the configuration file, then the numbers in the appropriate terminal description file are used except for the terminfo and pcwindow terminal descriptions. If no ICLINES or ICCOLUMNS information can be found, the defaults of 24 lines by 80 columns are used.

The only valid terminal type for the master console is a pcwindow-type. For serial lines and telnet sessions in “simple mode” any of the other ICTERM settings can be used but it should match the emulation being used.

When using CGI, ICTERM=file should be set.

More on these terminal types can be found in the ICTERM Chapter, starting on page 173.

ICLINES should be set to the line at which the terminal will scroll the screen when a line-feed (<lf>) is sent. (Line-feed and newline are the same.)

ICCOLUMNS should be set to the column position after which the terminal will wrap to the next line. If both normal and compressed spacing is available then two values should be specified.

With the pcwindow-type selection almost all combinations of lines and columns can be selected as the necessary scroll-bars will be presented depending on the selected font and size.

B.7. ICDATAPATH

ICDATAPATH specifies directories and/or COBOL libraries (up to 16) that ICRUN will search for to open COBOL data file with a simple name.

The syntax is:

```
ICDATAPATH=dir|file[;dir|file]...
```

Where

dir

Specifies a directory in which ICRUN should look for data files with simple names.

file

Specifies a COBOL library file in which ICRUN should look for data files with simple names.

If not specified in the environment, then the ICDATAPATH entry from the configuration file is used. If neither is specified, a default ICDATAPATH of the current directory is used.

If ICDATAPATH is specified, the current directory is searched last by default and must be specified by an empty entry or a period in ICDATAPATH if you want it to be searched sooner. The list is searched for data files in the order given in ICDATAPATH.

ICDATAPATH must only include directory and library names.

The ICDATAPATH only applies to a data file with a simple name. The syntax is the same as for a system PATH.

Data files found in a COBOL library can only be opened for Read access.

A sample entry is:

```
ICDATAPATH=.;D:\DATA;C:\MAIN\DATA;E:
```

which searches the current directory (current drive), D:\DATA, C:\MAIN\DATA, and finally E: (current directory) for a particular data file.

B.8. ICFONT, ICFONTSIZE

ICFONT can be used to specify a default fixed font for the GUI runtime (icrunw) to use when painting the console screen. The default selection is “Courier New”. ICFONTSIZE can be used to specify the default point size for the GUI runtime (icrunw) to use when painting the console screen. The default selection is 11.

The syntax is:

```
ICFONT=font
ICFONTSIZE=fontsize
```

Where

font

Specifies a font to be used.

fontsize

Specifies the point size for the font. *Fontsize* must be a integer value between 2 and 99 inclusive and must be available for the given font.

The default font is "Courier New" and the default point size is 11.

Some fixed fonts that are available on most machines are:

Courier, Courier New, Fixedsys, and Terminal.

Courier New is a TrueType font such that all point sizes are available. Courier is usually a bit map with a limited selection of sizes.

If the indicated font or size is not available, Windows will pick a font or size that matches as close as possible to the given settings. If a proportional font is selected the screen handler will do the best it can to kept the columns aligned.

The ICRUNW and ICRUNRC executables try to select appropriate Bold and Underline fonts to match the selected font and allow the Bold and Underline screen attributes to be more fully implemented.

The term "Bold" can be appended to a font to provide a slighter darker appearance if needed.

B.9. ICPCQDIR

ICPCQDIR specifies a particular directory in which simple printer filenames should be created (printer control directory).

The syntax is:

```
ICPCQDIR=dir
```

Where

dir

Specifies a valid pathname for the directory in which COBOL printer files with simple names are to be located.

If not specified in the environment then the ICPCQDIR from the configuration file is used. If neither is specified, printer files are located in the current directory.

The Printer files affected are those with an ASSIGN TO PRINTER filename and ASSIGN TO PRINTER-1 filename where the filename specified is a simple filename, i.e., no directory qualifiers. Each user can have his own printer control directory or there can be a common one for a group of people or the entire network.

B.10. ICPCQFILTER

ICPCQFILTER specifies a particular printer control utility filter. The display may be modified for the life of the runtime by setting the ICPCQFILTER environment variable to establish the default filter. The specified filter will be active every time the Printer Control Utility is entered. This method applies to all methods of starting the Printer Control Utility (IC_PRINT_STAT).

ICPCQFILTER, takes values in a format similar to the command lines of the Interactive COBOL utilities.

The syntax is:

```
ICPCQFILTER=command-line
```

Where *command-line* includes:

-B *min*:*max*

Printed by console numbers ranging from *min* to *max*

-D *directory*
Pathname of directory directly or indirectly containing the print job

-F *filename*
Simple filename of the print job

-I *min*:*max*
Owned by console numbers ranging from *min* to *max*

-M *mode*
The job's current status. Valid modes are:

1=Not yet printed 2=Already printed 3=Error occurred
4=Update in progress 5=Queued to print 6=Holding in queue
7=Printing 8=Retrying 9=Terminating

-O *owner*
Username of the owner

-P *printedby*
Username of the last user to print the file

-Q *min*:*max*
Printer control queues ranging from *min* to *max*

-r
Read access to a file

-S *min*:*max*
Filesize (in bytes) ranging from *min* to *max*

None of the command line options may be specified more than once.

For example, if ICPCQFILTER is set to "-O mary -Q 2:3", the Printer Control Utility would only display files owned by the user "mary" and destined for either @PCQ2 or @PCQ3.

B.11. ICREVERSE

ICREVERSE informs Interactive COBOL how to interpret reverse codes from a COBOL program.

The syntax is:

```
ICREVERSE=filter|ignore|process
```

Where

filter
Causes the runtime to watch for reverse codes from the program and to NOT send them to the terminal, since it does not support reverse.

ignore
Tells the runtime that the user wants total control of the screen and may be sending binary reverse data to the screen and that the runtime should ignore all reverse codes (i.e., do not look for reverse codes). If running in this mode, the SCREEN OPTIMIZER cannot correctly repaint a user's screen that includes reverse.

process
The runtime interprets reverse codes from the program and sends the appropriate sequences to the terminal. Process is the default.

If not specified in the environment then the ICREVERSE from the configuration file is used.

B.12. ICRUN

The contents of the *ICRUN* environment variable are treated like switches entered from the command line and processed before any other switches or arguments when starting ICRUN or ICRUNW.

The syntax is:

```
ICRUN=icrun-switches
```

Where

icrun-switches

Specifies any valid command line switches for the runtime but not a command-line argument.

Remember the switch indicator (either `-' or `/') must be consistent through the entire command line.

B.13. ICRUNDIR

ICRUNDIR specifies a particular directory that the runtime will position to after it has successfully processed the command line and optionally opened its adit log. All subsequent current directory usage will use this value

The syntax is:

```
ICRUNDIR=dir
```

Where

dir

Specifies a valid pathname for the directory into which the runtime will position to just before it starts to run the initial program.

If the specified value is NOT a valid directory, an error will be given at startup and the runtime will exit.

If the Info switch (-i) is given, a message will be given about processing the ICRUNDIR environment variable.

B.14. ICRUNLK

ICRUNLK specifies where the link file can be found.

The syntax is:

```
ICRUNLK=file
```

Where

file

Specifies a valid filename for the link file.

If not specified in the environment then the ICRUNLK from the configuration file is used. If neither is specified, ICRUN does not look for a link file.

If specified in one form or other, then the link file must exist, be readable, and be a valid link file.

The entry can be a directory (in which case icrun.lk is appended for the file) or a complete filename. This feature can be used to link filenames of one type or length to another without changing the COBOL programs. This file is created with the ICLINK utility explained in the Utilities Manual. The link file is opened, read, and closed at startup.

B.15. ICSCROPT

ICSCROPT informs Interactive COBOL and ICCONFIG whether to enable the Interactive COBOL SCREEN OPTIMIZER.

The syntax is:

```
ICSCROPT=off|on|full|partial|mute
```

Where

off

Disables screen optimization

partial

Enables simple single screen optimization

on and full

Enables full screen optimization

mute

Disables any screen optimization and prevents Interactive COBOL from sending any implied codes of its own at startup or termination.

If not specified in the environment then the ICSCROPT entry from the configuration file is used.

The SCREEN OPTIMIZER keeps track of all data sent to the console and prevents rewriting the same data multiple times. It uses an image of the current screen always in memory.

Usually screen optimization will provide improved screen performance.

The full option usually provides better screen update performance than the partial option, but it requires more memory and cpu time.

B.16. ICSDMODE

ICSDMODE instructs Interactive COBOL how to enable the Interactive COBOL SCREEN HANDLER that provides many SCREEN DEMON™ like features.

The syntax is:

```
ICSDMODE=disabled|underline|0|reverse|1|drawlines|2
```

Where

disabled

Disables the SCREEN HANDLER

0 or underline

Run in standard SCREEN DEMON format, which is to underline the row above the box and underline the last row in the box for the top and bottom lines, and use reverse video for the sides.

1 or reverse

Use reverse video for the entire box. This means that two (2) more lines than in standard mode are hidden under the box.

2 or drawlines

Use the line drawing character set of a terminal for the entire box. As in 1 above, two (2) more lines than in standard mode are hidden under the box. If a particular terminal does not have a line drawing character set +, -, and | are used for the corners, horizontal, and vertical portions of the box, respectively. Currently, only the terminal types ibm, xenix, 386ix, and pcwindow support the line drawing characters by default.

If not specified in the environment then the ICSDMODE from the configuration file is used.

For more information on the Interactive COBOL SCREEN HANDLER, see the Screen Handler section in the ICRUN Chapter of the Interactive COBOL Language Reference & Developer's Guide.

B.17. ICTIMEOUT

ICTIMEOUT specifies a global timeout in seconds for all screen ACCEPTs (and STOP literal). If no key has been hit in that time the ACCEPT will timeout with the ESCAPE KEY set to 99.

The syntax is:

```
ICTIMEOUT=n
```

Where

n

Valid timeout values are:

<=0 or >=65535	Wait Forever
>6300 and <65535	Set to 6300
1-6300	Set to *n* seconds

6300 seconds is 1 hr. & 45 min.

If not specified in the environment, then the ICTIMEOUT from the configuration file is used.

The ACCEPT with TIME-OUT statement and the IC_SET_TIMEOUT builtin will override this selection.

B.18. LISTFILE

LISTFILE specifies a filename for the @LIST file.

The syntax is:

```
LISTFILE=filename
```

Where

filename

Specifies a filename to be used to replace @LIST in a COBOL OPEN.

B.19. PCQ, PRN, and SER

PCQ, PRN, and *SER* allow the generic printer control queue (@PCQ), the generic printer (@PRN), and the generic serial (@SER) devices to be specified for each process.

The syntax for each is:

```
PCQ=n
PRN=n
SER=n
```

Where

n

Specifies a number from 0 to 2047 for PCQ, PRN, and SER that is the appropriate device (@PCQn, @PRNn, or @SERn) enabled and defined in the configuration file.

B.20. PTS, PTSMODE, PTSFONT, PTSFONTSIZE

These environment variables control how the process handles the local printer device, i.e. printer-pass thru.

The syntax is:

```
PTS=default | prompt | prn | printer-name
PTSMODE= char | paint
PTSFONT=prompt | font-name
PTSFONTSIZE=ptsize
```

For PTS

prn use the setting for PRN0. (This is how ICRUNW worked in pre-3.47.)
default use the default Windows printer.
Prompt display the Print Dialog to select a particular printer. If the dialog is Canceled or Closed no printer is selected.
printer-name use the specified Windows printer.

For PTSMODE

char specifies that print data is sent directly to the print driver. (This is how it worked in pre-3.47).
paint specifies that data should actually be "painted" onto the printer. In this mode, fonts, and fontsizes can be specified. In addition, this mode will work with GDI-printers.

For PTSFONT (only used when in paint mode)

prompt says to provide the ChooseFont dialog to select fonts and sizes.
font-name selects the specified font.

For PTSFONTSIZE (only used when in paint mode)

pointsize specifies a valid size from 2 to 99.

A setting of PTS=prompt, PTSFONT=prompt, and PTSMODE=paint provides for showing the Windows print dialog, allowing font, color, and size to be specified, and the "painting" of data to the printer.

Also note that the Local print option from the Printer Control Utility uses Printer Pass Thru (@PTS)

C. **Syntax**

The syntax for the runtime system is:

```
icrun [-a[:aflag]|-A file|dir[:aflag]] [-B b|i] [-D yyyymmdd[:hhmmss]]
   [-E var=value]... [-G {drsu}...] [-h|-?] [-i] [-M mode]
   [-N {bdeiowx}...] [-p] [-q] [-s] [-S os] [-T n] [-U l|n|u] [-W title]
   [-z|-Z ddir] [program [argument]...]
```

or icrunw for the GUI version.

Where

-a[:*aflag*] or -A *file*|*dir*[:*aflag*] (Audit)

Enables auditing (default icrun.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 b|i (Brand)

Specifies the record manager to use by default. ICISAM is specified by i and is the default, while Btrieve is specified by a b. To use Btrieve, a Btrieve requestor must be available and the BTRIEVE license option must be provided for the **ICOBOL** Runtime license.

-D yyyymmdd[:hhmmss] (Date bias)

Specifies a date with an optional time from which to bias all COBOL date and time functions (ACCEPT FROM DAY, DATE, TIME, DAY-OF-WEEK, the IC_FULL_DATE builtin, message sending, etc.). The current date/time is subtract from this date/time and the resultant value is added to all date and time functions. No date/time changes are made to the operating system. This can be used to set a date/time forward or backward for testing purposes.

-E *var*=*value* (set Environment value)

Specifies that the environment variable *var* should be setup with the given *value*. This is especially useful for ThinClients to pass environment information over to the server runtime..

-G {drsu}... (General)

Specifies general option(s):

d (Duplicate) Generate FILE STATUS 02 with ANSI COBOL 74 programs when using ICISAM version 7 files and a duplicate key is processed as defined by ANSI.

r (RDOS) Uses the first underscore in an ACCEPT field as a terminator that erases all characters to the right of it from the string of characters entered into the field. This behavior is how Interactive COBOL on RDOS operated.

s (Switch) Enable strict switch processing. The first "/" indicates the start of program switches.

u (Upper-case) Upper-case program names returned by ACCEPT FROM ENVIRONMENT.

-h|-? (Help)

Display help text.

-i (Information)

Displays all Information (Info) messages while starting. The default is to not display information messages.

-M *mode* (window Mode)

Specify the initial window mode for the runtime. Valid values for *mode* are the values for the IC_WINDOWS_SHOW_CONSOLE builtin plus 0. 0 does nothing, 1 is Hide, 2 is Maximize, 3 is Minimize, etc.

-N {bdeiow}... (No options)

Specifies NO option(s). Valid option values are:

b (No-re-assignment) Do not reassign terminal number.
d (No-device summary) Do not display logical device summary
e (No-embedded spaces) Do not allow embedded spaces in filenames.
i (No-interrupts) Do not allow console interrupts
o (No-OCCURS) Do not do OCCURS DEPENDING ON bounds checking
w (No-warnings) Do not display any warning messages
x (No-close) Do not offer the Close option on the Window for icrunw

-p (Prompt for username)

Prompt for username to pass to the server on a remote connection (ICNETD).

-q (Quiet)

Enables quiet operation.

-s (Startup program)

Forces the runtime to use the program from the command line as the initial startup program for the first console, although it will operate in Logon mode after that.

-S *os* (System Code)

Overrides the default system code returned by an ACCEPT FROM ENVIRONMENT. *os* must be a value from 00 to 99.

-T *n* (Terminal number)

Specifies the console number in the configuration file.

-U l|n|u (Username)

Specifies the case in which user name will be returned in the ACCEPT FROM USER NAME statement. Valid cases are upper-case (u), lower-case (l), or no conversion (n). The default is lower-case (l).

-W *title* (Title)

Sets the window title for the runtime to be "*title* - program" instead of "icrun - program"

-z | -Z *ddir* (Debug)

Instructs the runtime to start in debug mode using the current directory (-z) or the directory *ddir* as the default directory for symbol files (.sy).

program (Program mode)

Specifies a COBOL program name including optional program-switches. This forces the runtime into Program mode and starts the program on the next available console. When the program terminates, the runtime terminates. When used together with the Quiet switch (-q) no runtime termination message is displayed when it terminates.

argument

Provide optional arguments that will be passed to the specified program's Linkage section. Each argument gets placed in the next Linkage item. If more arguments are specified than there are Linkage items they are ignored.

When using the Prompt for username switch, the Save option is ignored in the username/password screen. For thick clients, the prompt is done when a connect to a network resource via ICNETD is made.

More on program switch processing can be found in the CALL PROGRAM statement in the Language Reference Manual.

Any argument or selection that has embedded spaces in it must be quoted or else it will be taken as multiple arguments.

When using an audit log with the runtime:

A) If any Error is encountered (like Indexed out of Range, etc.) the message will be written to the audit log along with the pc where the error occurred, the next pc, the exception status register, and the name of the program.

B) If an asyncronous event (like a Ctrl-C, Ctrl-Break, a Windows-Close, an Abort or Kill from another console) are encountered, then that message will be written to the audit log along with the pc where the error occurred, the next pc, the exception status register, and the name of the program.

The format for the above cases:

```
Error:  message
    OP=nnn1 PC=nnn2 E=nn3 in program-name
```

Where

nnn1	is the current pc
nnn2	is the next pc that would have been executed
nn3	is the current exception status
program-name	is the current COBOL program.

When running in logon-mode, obviously there could be multiple of these type of messages in the audit log as the user would only need to hit newline to continue from these errors.

Running with an audit log is always recommended, as it will allow you to look back over events if needed. In general, these runtime audit logs should be small. The Audit option with pid is probably the best type of auditing to use.

```
icrun -a:p
```

(for example)

Then you can use the pid number to look up the process in the icpermit and/or icexec logs if needed.

Several examples for Program mode are:

1)
```
icrun -q mainjob
```

Runs the program MAINJOB in Program mode with no runtime specific information displayed.

2)
```
icrun -q mainjob/a/b/c
```

Runs the program MAINJOB in Program mode with no runtime specific information displayed and the COBOL switches A, B, and C set to ON.

3)
```
icrun mainjob available-data
```

Runs the program MAINJOB in Program mode with information displayed at startup passing the data given by available-data into the first argument in MAINJOB's linkage section.

4)
```
icrun mainjob "data one" "data two"
```

Runs the program MAINJOB in Program mode with information displayed at startup passing the data given by "data one" and "data two" as two separate arguments into the first and second arguments in MAINJOB's linkage section.

The runtime searches the user's environment for its environment entries (as explained in the previous section).

To run the runtime, the executable file (ICRUN.EXE or ICRUNW.EXE) and the appropriate .dll files (icrun.dll, sp2.dll, icbltn.dll) must be available and the following conditions must be meet:

1) An ICPERMIT (described on page 67, 68, 71) is running with a free runtime license,

or

2) Remote authorization has been specified and an ICPERMIT is running on a remote machine with a free runtime license.

3) An ICEXEC must be running to supply configuration information.

4) A terminal description file to provide terminal information, e.g., pcwindow.tdi, dg.tdi, etc..

Two optional files can also be supplied. These are:

A link file (icrun.lk) can be supplied to link old filenames to new filenames. This feature can be used to link RDOS type filenames to MS-DOS type files without changing the COBOL programs. This file is created with the ICLINK utility explained in the Utilities Manual.

A COBOL library file (icobol.cl) can be supplied, which holds COBOL programs to be executed. This optimizes CALL and CALL PROGRAM statements. The ICLIB utility is used to create and maintain the COBOL library file. ICLIB is explained in the Utilities Manual.

ICRUN.EXE can be located in the current directory or anywhere along the current PATH.

The link file (icobol.lk) is sought using the ICRUNLK entry.

After being loaded, the runtime allocates memory as needed.

Additional program-areas are allocated for each call. Program-areas are placed in a re-use list whenever a program is CANCEL'ed or when the console executes a STOP RUN.

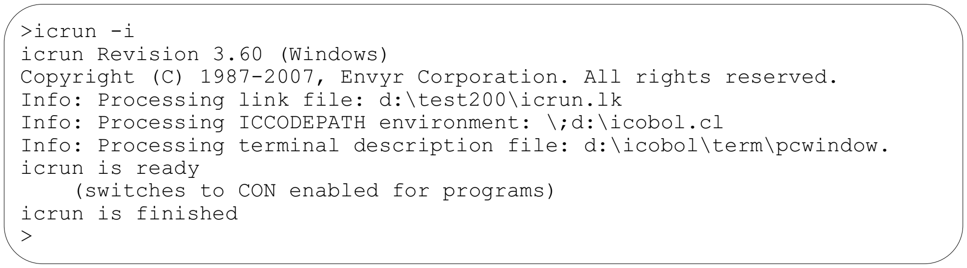
To start the runtime system from the command line type "ICRUN" or "ICRUNW" and press the ENTER key. The runtime performs the following steps with the resultant messages specified as either Information or Warning messages. As the runtime starts it can be interrupted at any point by pressing Ctrl-Break. By default, information messages are not displayed.

1) The runtime revision information and copyright notice.
2) Authorization is checked.
3) The terminal descriptions file is processed and its full pathname displayed.
4) Confirmation of processing ICRUNLK with its full pathname.
5) Confirmation of processing ICCODEPATH and/or ICDATAPATH with their contents displayed.
6) Confirmation of processing ICPCQFILTER.
7) Finally, if no errors were encountered, a message saying that the system is ready is displayed.
8) Memory is requested as needed to fulfill requirements for program loading and screen saving.

The runtime's messages are placed in the base console or sent to the audit log.

SCREEN 35 is a sample display of the base console of a runtime session with the Info switch.

Info messages are just that, information messages showing that something was done. These messages are displayed only if the Information switch (-i) was given.



```
>icrun -i
icrun Revision 3.60 (Windows)
Copyright (C) 1987-2007, Envyr Corporation. All rights reserved.
Info: Processing link file: d:\test200\icrun.lk
Info: Processing ICCODEPATH environment: \;d:\icobol.cl
Info: Processing terminal description file: d:\icobol\term\pcwindow.
icrun is ready
    (switches to CON enabled for programs)
icrun is finished
>
```

SCREEN 34. SAMPLE ICRUN STARTUP SCREEN

Warning messages are items that should be examined and possibly corrected if there really is a problem.

If any time during startup the runtime detects a fatal error, the startup will terminate. Fatal errors require you to perform some corrective action. The error message will suggest the particular area that needs to be corrected.

Several more common fatal errors and the corrective steps to take are:

1) *Error: Connection refused (oserr=10061) Connecting to localhost*
Error: The required software license is not authorized.

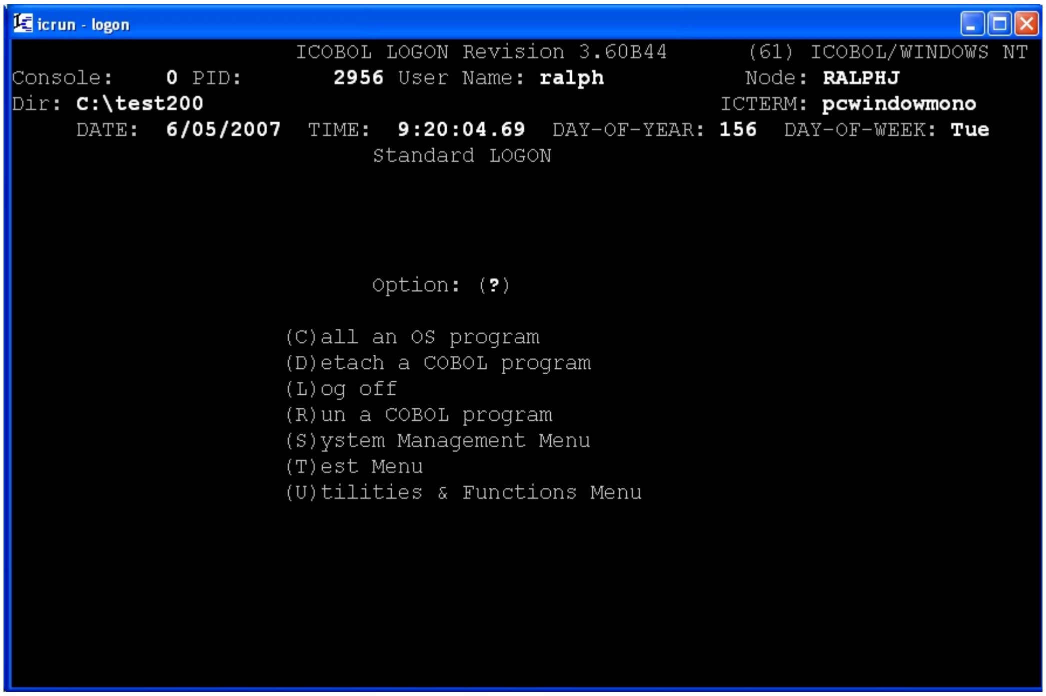
Local licensing is being used and no ICPERMIT is running on this machine. Start an ICPERMIT to provide license services. See more on ICPERMIT on page(s) 67, 68, 71.

2) *Error: Connection refused (oserr=10061) Connecting to mainmachine*
Error: The required software license is not authorized.

Remote licensing is being used and either there is no machine on the network with the remote name or no ICPERMIT is running on that machine. See more on ICPERMIT on page(s) 67, 68, 71.

3) *Insufficient memory: Allocating* ****

The runtime ran out of memory allocating the specified data structure, usually it shows how much more memory is required for this structure. More memory can be added or use ICCONFIG or ICEDCFW to reduce the amount of buffers allocated, reduce the number of active devices in the run unit, reduce other numbers in the system parameter screen, etc. until the error disappears.



```
                        ICOBOL LOGON Revision 3.60B44      (61) ICOBOL/WINDOWS NT
Console:    0 PID:        2956 User Name: ralph            Node: RALPHJ
Dir: C:\test200                                          ICTERM: pcwindowmono
     DATE:  6/05/2007  TIME:   9:20:04.69  DAY-OF-YEAR: 156  DAY-OF-WEEK: Tue
                              Standard LOGON



                              Option: (?)

                       (C)all an OS program
                       (D)etach a COBOL program
                       (L)og off
                       (R)un a COBOL program
                       (S)ystem Management Menu
                       (T)est Menu
                       (U)tilities & Functions Menu
```

The Title Bar top left shows the process name (icrun) and the name of the current COBOL program (logon).

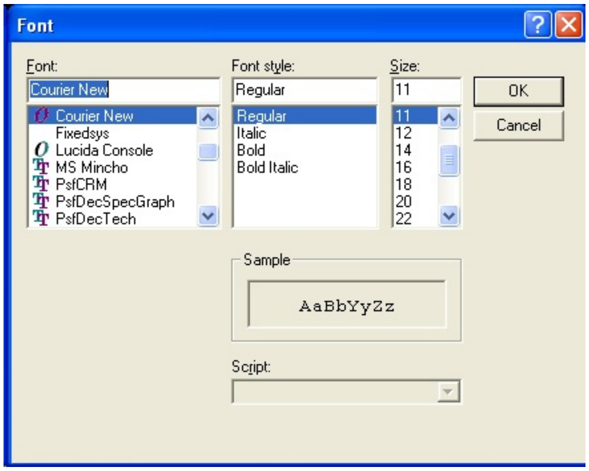
Right-clicking in the Title bar will give the System menu including:

Restore, Move, Size, Minimize, Maximize, Close, Edit, Font, Compress, Select printer, Printer font, and About selections.

Where

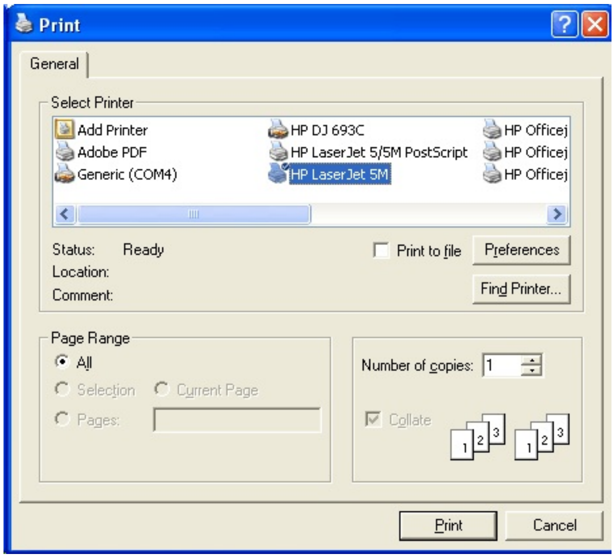
Edit brings up the Copy, Print, Paste, and Select all dialog

Font brings up the ChooseFont dialog box as shown below to allow the screen font, style, and size to be changed.



Compress toggles between compressed and non-compressed mode if available, otherwise it will be grayed out. In compress mode, a check-mark will be displayed.

Select printer, brings up the Printer selection dialog as shown below to select the Windows printer used for printer pass thru (@PTS).



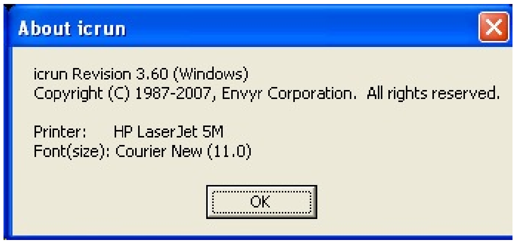
Status: Ready
Location:
Comment:

● All

Number of copies: 1

Printer font selects the font, style, and size for the printer selected above. It implies paint mode.

About shows the current revision of ICRUN along with currently selected printer, font, and size as shown below:



icrun Revision 3.60 (Windows)
Copyright (C) 1987-2007, Envyr Corporation. All rights reserved.

Printer: HP LaserJet 5M
Font(size): Courier New (11.0)

Right-clicking in the screen area will give the Edit menu that includes:
Copy, Print, Paste, and Select all dialog

Holding down the left-mouse button will allow data within the screen to be selected by moving the cursor..

D. Termination

To terminate the runtime, the IC_SHUTDOWN builtin can be called.

If enabled, Ctrl-C can be used to terminate the runtime.

If enabled, the close button [x] from the Title bar can also be clicked to terminate the runtime.

Do not use the Ctrl-Alt-Del combination to terminate the runtime.

E. Failure

On Shutdown, if any device cannot be properly terminated, ICRUN will have tried to clean up the file system by flushing all open files.

IX. THINCLIENT (Char)

A. Introduction

ThinClient support is provided to allow a standard application to be run on a server (either UNIX or Windows) and deploy the application's user interface to any client machine connected to the server via a TCP/IP network. The clients can be any UNIX or Windows based machines for which an icrunrc executable is provided. If running under Windows then both gui and character applications are supported.

The main advantage to the thin-client architecture (versus the standard thick-client architecture which is done via a runtime accessing data files over the network) is that the business logic and database files are maintained and executed in a central location yet end-users still have access to the user interface and all its associated conveniences. Only the standard **ICOBOL** screen interfaces will be shipped to the client. The COBOL program itself runs on the server machine accessing files directly on that machine. When using a UNIX icrunrc, NO gui Sp2 and/or FormPrint interfaces are allowed when running ThinClient.

You can think of the ICRUNRC/ICRUNRS combination as a type of telnet session with ICRUNRC being the telnet client/terminal emulator and ICRUNRS being the telnetd server connection.

The TCP/IP connection between the ThinClient client (icrunrc) and the ThinClient server (icrunrs) is encrypted.

No telnetd server must be purchased for the Windows server machine to run client connections.

On Windows, no terminal emulator must be purchased for the client to connect to Windows servers or UNIX servers when using the ThinClient.

On Windows, the ICRUNRC executable tries to select appropriate Bold and Underline fonts to match the selected font and allow the Bold and Underline screen attributes to be more fully implemented.

When using the ThinClient, you are assured that only COBOL access can be made to the remote machine. No interactive command session can be invoked.

B. Environment Entries

B.1. Overview

ICRUNRC searches for specific Interactive COBOL entries in the user's current environment. These entries allow the user to tailor a particular session of Interactive COBOL for a particular application, company, etc. Environment entries can be set in Windows by using the *SET* command.

Interactive COBOL on Windows environment entries in addition to the common ones of ICROOT and TZ are:

ICBGCOLOR	Specify the initial background color
ICCOLOR	Specify how to support color
ICCOLUMNS	Columns for terminal
ICFGCOLOR	Specify the initial foreground color
ICFONT	Specify the font name for the GUI runtime (icrunw)
ICFONTSIZE	Specify the point size for the font for the GUI runtime (icrunw)
ICLINES	Lines for terminal
ICREVERSE	Specify how to support reverse
ICRUNRC	Default switches for the icrunrc
ICSCROPT	SCREEN OPTIMIZER selection
ICTERM	Terminal type
PTS	Print Pass Thru device
PTSMODE	How to support @PTS

PTSFONT	PTS font
PTSFONTSIZE	PTS fontsize

B.2. ICBGCOLOR, ICCOLOR, ICFGCOLOR

ICCOLOR specifies how to interpret color codes from a COBOL program. Valid selections are filter, ignore, and process, the default is filter.

The syntax is:

```
ICCOLOR=filter|ignore|process
```

Where

filter

Causes the client to watch for color codes from the program and to NOT send them to the terminal, since it does not support color. Filter is the default.

ignore

Tells the client that the user wants total control of the screen and may be sending binary color data to the screen and that the client should ignore all color codes (i.e., do not look for color codes). If running in this mode the SCREEN OPTIMIZER should not be enabled as the client cannot repaint a user's screen.

process

The client interprets color codes from the program and sends the appropriate sequences to the terminal. When set to Process the initial background and foreground colors are set by the client at startup.

ICBGCOLOR sets the default background color to the indicated value when running with ICCOLOR set to Process. Valid selections are black (0), blue (1), green (2), cyan (3), red (4), magenta (5), brown (6), and white (7) either as the name or the number. The default is black (0).

The syntax is:

```
ICBGCOLOR=black|blue|green|cyan|red|magenta|brown|white|
          0|1|2|3|4|5|6|7
```

ICFGCOLOR sets the default foreground color to the indicated value when running with ICCOLOR set to Process. Valid selections are black (0), blue (1), green (2), cyan (3), red (4), magenta (5), brown (6), and white (7) either as the name or the number. The default is white (7).

The syntax is:

```
ICFGCOLOR=black|blue|green|cyan|red|magenta|brown|white|
          0|1|2|3|4|5|6|7
```

Currently only pcwindows, DG, and ANSI based terminals support color.

B.3. ICCOLUMNS, ICLINES, and ICTERM

ICTERM specifies what type of terminal is attached. ICLINES and ICCOLUMNS specify the number of lines and columns for the terminal.

The syntax is:

```
ICTERM=terminal-type
ICLINES=lines
ICCOLUMNS=columns[:ccolumns]
```

Where

terminal-type

Specifies a valid ICTERM entries described below or in the ICTERM Chapter starting on page 173, with a corresponding terminal description file

lines

Specifies the number of lines for this terminal. It can range from 24 to 255.

columns[:*ccolumns*]

Specifies the default number of columns for this terminal. It can range from 80 to 255. If the second value is given, then a secondary set of columns are supported. The lowest value is considered the normal mode and the higher value is considered the compressed mode.

If ICTERM is not specified in the environment, then pcwindow is used.

If ICLINES and/or ICCOLUMNS are not specified, or are 0, then the numbers in the appropriate terminal description file are used except for the terminfo and pcwindow terminal descriptions. If no ICLINES or ICCOLUMNS information can be found, the defaults of 24 lines by 80 columns are used.

The only valid terminal type for the master console is pcwindow-type selections..

More on these terminal types can be found in the ICTERM Chapter, starting on page 173.

ICLINES should be set to the line at which the terminal will scroll the screen when a line-feed (<lf>) is sent. (Line-feed and newline are the same.)

ICCOLUMNS should be set to the column position after which the terminal will wrap to the next line. If both normal and compressed spacing is available then two values should be specified.

B.4. ICFONT, ICFONTSIZE

ICFONT can be used to specify a default fixed font for the client to use when painting the console screen. The default selection is “Courier New”. ICFONTSIZE can be used to specify the default point size for the GUI client to use when painting the console screen. The default selection is 11.

The syntax is:

```
ICFONT=font
ICFONTSIZE=fontsize
```

Where

font

Specifies a font to be used.

fontsize

Specifies the point size for the font. *Fontsize* must be a integer value between 2 and 99 inclusive and must be available for the given font.

The default font is "Courier New" and the default point size is 11.

Some fixed fonts that are available on most machines are:

Courier, Courier New, Fixedsys, and Terminal.

Courier New is a TrueType font such that all point sizes are available. Courier is usually a bit map with a limited selection of sizes. If a proportional font is selected the screen handler will do the best it can to kept the columns aligned.

Appropriate Bold and Underline fonts will be selected to match the selected font and allow the Bold and Underline screen attributes to be more fully implemented.

If the indicated font or size is not available, Windows will pick a font or size that matches as close as possible to the given settings.

The term “Bold” can be appended to a font to provide for a slighter darker appearance if needed.

B.5. ICREVERSE

ICREVERSE specifies how to interpret reverse codes from a COBOL program.

The syntax is:

```
ICREVERSE=filter|ignore|process
```

Where

filter
: Causes the client to watch for reverse codes from the program and to NOT send them to the terminal, since it does not support reverse.

ignore
: Tells the client that the user wants total control of the screen and may be sending binary reverse data to the screen and that the client should ignore all reverse codes (i.e., do not look for reverse codes). If running in this mode, the SCREEN OPTIMIZER cannot correctly repaint a user's screen that includes reverse.

process
: The client interprets reverse codes from the program and sends the appropriate sequences to the terminal. Process is the default.

B.6. ICRUNRC

The contents of the *ICRUNRC* environment variable are treated like switches entered from the command line and processed before any other switches or arguments when starting ICRUNRC.

The syntax is:

```
ICRUNRC=icrunrc-switches
```

Where

icrunrc-switches
: Specifies any valid command line switches for icrunrc.

Remember the switch indicator (either `-' or `/') must be consistent through the entire command line.

B.7. ICSCROPT

ICSCROPT specifies whether to enable the Interactive COBOL SCREEN OPTIMIZER.

The syntax is:

```
ICSCROPT=off|on|full|partial|mute
```

Where

off
: Disables screen optimization

partial
  Enables simple single screen optimization
on and full
  Enables full screen optimization
mute
  Disables any screen optimization and prevents the client from sending any implied codes of its own at startup or termination.

The SCREEN OPTIMIZER keeps track of all data sent to the console and prevents rewriting the same data multiple times. It uses an image of the current screen always in memory.

Usually screen optimization will provide improved screen performance.

The full option usually provides better screen update performance than the partial option, but it requires more memory and cpu time. Full is the default.

B.8. PTS, PTSMODE, PTSFONT, PTSFONTSIZE

These environment variables control how the process handles the local printer device, i.e. printer-pass thru.

The syntax is:

```
PTS=default | prompt | prn | printer-name
PTSMODE= char | paint
PTSFONT=prompt | font-name
PTSFONTSIZE=ptsize
```

For PTS

default use the default Windows printer.
Prompt display the Print Dialog to select a particular printer. If the dialog is Canceled or Closed no printer is selected.
printer-name use the specified Windows printer.

For PTSMODE

char specifies that print data is sent directly to the print driver.
paint specifies that data should actually be "painted" onto the printer. In this mode, fonts, and fontsizes can be specified. In addition, this mode will work with GDI-printers.

For PTSFONT (only used when in paint mode)

prompt says to provide the ChooseFont dialog to select fonts and sizes.
font-name selects the specified font.

For PTSFONTSIZE (only used when in paint mode)

pointsize specifies a valid size from 2 to 99.

A setting of PTS=prompt, PTSFONT=prompt, and PTSMODE=paint provides for showing the Windows print dialog, allowing font, color, and size to be specified, and the "painting" of data to the printer.

Also note that the Local print option from the Printer Control Utility uses Printer Pass Thru (@PTS)

C. **Syntax**

The syntax is:

```
icrunrc [-a[:aflag] | -A file|dir[:aflag]] [-E var=val]... [-h|-?] [-i]
   [-M machine[:port]] [-N wx] [-p] [-q] [-t] [ -- icrunrs-switches ] [ program
   [arg]... ]
```

Where icrunrc switches are:

-a[:*aflag*] or -A *file*|*dir*[:*aflag*] (Audit)
  Enables auditing (default icrunrc.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 *var*=*val* (Environment)
  Set the environment variable *var* to *val* in the current environment.

-h | -? (Help)
  Display help text.

-i (Info)
  Display additional informational messages

-M *machine*[:*port*]] (Machine)
  Specifies the remote machine as either an ip-address or a machine-name and optional port to connect to. If *port* is not given, 7334 is the default. If *machine* is not given, then localhost is used.

-N wx (No option)
  Specifies a NO option:
    w - No warnings messages are generated.
    x - No Close (X) option on the window.

-p (Prompt for username)
  Prompt for username to pass to the server.

-q (Quiet)
  Enables quiet operation.

-t (Trace)
  Enables trace information to be written to the audit log. Useful for debugging. When combined with the Info switch even more trace information is given.

-– (Start Icrunrs switches)
  Two dashes (--) instruct the ThinClient client to pass all remaining switches and arguments to the ThinClient server.

Program
  The COBOL program to start. If not specified then the program specified in the server's configuration file is used. If that is not set then logon is used.

Arg
  Any arguments needed by *program*.

When using the Prompt for username switch, the Save option is ignored in the username/password screen. For thin clients the prompt is done immediately.

Where *icrunrs-switches* are:

-B b|c|i (Brand)
  Specifies the default record manager brand:
    b=Btrieve (Windows), c=C-ISAM (Unix), i=ICISAM (default is i)

-C l|n|u (Case)
  Specifies the default Case conversion (Unix):
    l=lower,n=none,u=upper (default is l)

-D *date*[:*time*] (Date/time override)
  Specifies an override for the default date/time. Date/time formats are YYYYMMDD & HHMMSS

-E *var*=*val* (Environment)

Sets the environment variable var to value for icrunrs

-G drsu (General)
Specifies a General switch:
d=duplicate key I-O Status, r=RDOS-like ACCEPT, s=strict switch processing,
u=uppercase program name in ACCEPT FROM ENVIRONMENT

-i (Info)
Put out Info messages

-M mode (window Mode)
Specify the initial window mode for the runtime. Valid values for mode are the values for the IC_WINDOWS_SHOW_CONSOLE builtin plus 0. 0 does nothing, 1 is Hide, 2 is Maximize, 3 is Minimize, etc.

-N beinow (No)
Specifies NO options:
b=no auto reassign, e=no embedded spaces, i=no console interrupts,
n=no Nagle Algorithm suppressed, o=no OCCURS DEPENDING bounds check,
w=no warnings

-q (Quiet)
Specifies Quiet operation

-s (Startup mode)
Run *program* in startup mode

-S *n* (System code)
Specifies the System code value to be returned in ACCEPT FROM ENVIRONMENT

-T *n* (Terminal)
Specifies the Terminal number is n

-U l|n|u (Username)
Convert username case:
l=lower,n=none,u=upper (default is n)

-z | -Z *dir* (Debugger)
Run debugger, use remote (current or specified) dir for .sy

ICRUNRC Environment variables:

ICROOT	Help, print, and term descriptions	ICRUNRC	Command line options
ICCOLOR	Color handling mode	ICBGCOLOR, ICFGCOLOR	Color values
ICLINES	Screen height	ICCOLUMNS	Screen width
ICREVERSE	Reverse-video Control	ICSCROPT	Screen Optimizer mode
ICTERM	Terminal description		
ICFONT (win)	GUI font	ICFONTSIZE (win)	GUI font pointsize

ICRUNRS Environment variables:

ICROOT	Help and print descriptions	ICRUNRS	Command line options
ICPCQDIR	Print job file directory		
ICRUNLK	File name link file	PRN,SER,PCQ	Default device units
ICCODEPATH	.CX file path list	ICDATAPATH	Data file path list
ICTIMEOUT	Timeout Delay	ICABORT	Abort on Timeout
ICPCQFILTER	PCQ filter command	ICPERMIT_MACHINE	Servermachine[:port]
DATAFILE	@DATA resolution	LISTFILE	@LIST resolution
ICSDMODE	Screen Handler Mode	ICTMPDIR	Temporary file directory

The ICRUNRS environment variables are from the ICNETD - ICRUNRS process tree.

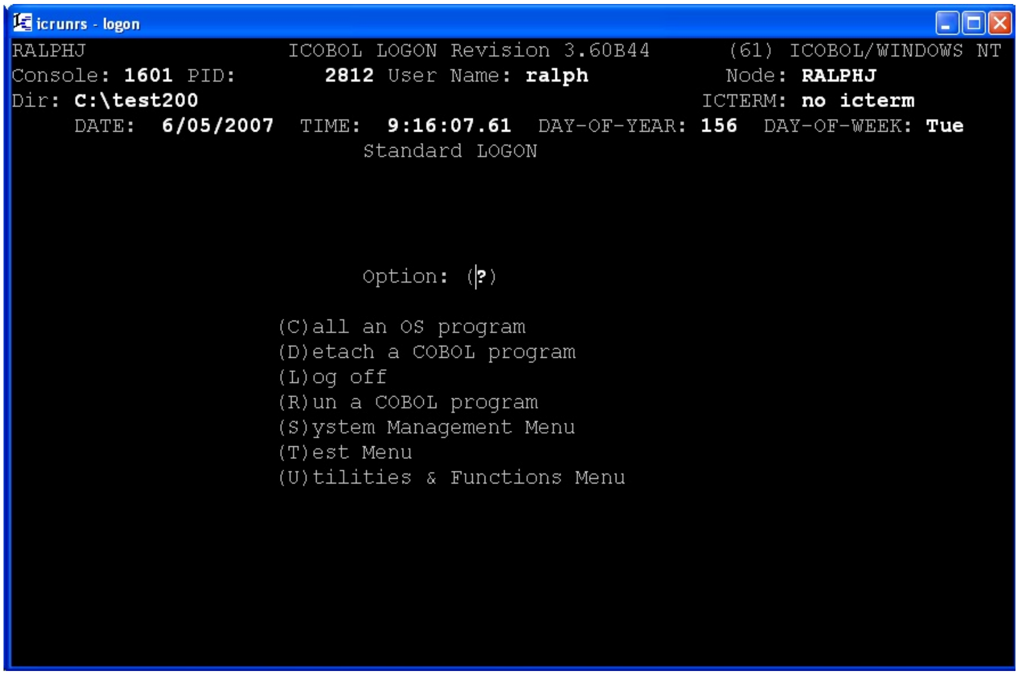
When ICNETD starts a ThinClient surrogate (either icrunrs or icthins), it will pass the client's ip-address and host-name in the environment variables ICREMOTEADDRESS and ICREMOTEHOST respectively. (Basically "-E ICREMOTEADDRESS=n.n.n.n -E ICREMOTEHOST=machine" on the client command line.) These can then be queried from a COBOL program 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 does look for these variables when it detects a Thinclient.

The trace switch, -t, requires the audit option.

When used with the Tracing switch (-t), the Info switch (-i) causes more trace info to be dumped to the audit log.

Below is a sample shot of the screen using the standard logon program provided showing the client machine-name as the Node name on line 2 at the far right.



```
RALPHJ                  ICOBOL LOGON Revision 3.60B44       (61) ICOBOL/WINDOWS NT
Console: 1601 PID:         2812 User Name: ralph            Node: RALPHJ
Dir: C:\test200                                           ICTERM: no icterm
     DATE:  6/05/2007  TIME:  9:16:07.61  DAY-OF-YEAR: 156  DAY-OF-WEEK: Tue
                             Standard LOGON




                             Option: (?)

                      (C)all an OS program
                      (D)etach a COBOL program
                      (L)og off
                      (R)un a COBOL program
                      (S)ystem Management Menu
                      (T)est Menu
                      (U)tilities & Functions Menu
```

The top left shows the process name (icrunrs) and the name of the current COBOL program (logon).

Right-clicking in the system area will give the System menu including:

Restore, Move, Size, Minimize, Maximize, Close, Edit, Font, Compress, Select printer, Printer font, and About selections.

Right-clicking in the screen area will give menu that includes:

Copy, print, Paste, and Select all

Holding down the left-mouse button will allow data within the screen to be selected.

D. ThinClient Features

The ThinClient connection provides for compression on the downstream leg. This should help performance on slower network connections.

The ThinClient server, icrunrs, has an additional No switch. The "-N n" switch will cause the socket connection from the server (icrunrs) to the client (icrunrc) to have the TCP Nagle Algorithm NOT be suppressed. In some cases, this may help screen performance over a slower WAN connection.

In the default case, the Nagle Algorithm is always disabled.

This option can be passed to icrunrs from icrunrc as:

```
icrunrc -M server-mac -- -N n
```

This option should be used with care as it has the tendency to slow down performance on an LAN. If auditing is enabled for the surrogates, an Info message is written to the audit log if info has also be enabled.

ICSCROPT=full should be enabled for both the icrunrc client and the icrunrs server to provide the best screen buffering options, especially over a slow WAN connection.

On the client machine, the ICRUNRC command will invoke the runtime thinclient. All screen input and output will be performed by this thin client.

On the server machine, an ICRUNRS will be started by ICNETD to run COBOL programs for that particular ThinClient session.

The ICRUNRC uses all the normal screen environment setting to set up its screen capabilities for the client. These include ICTERM, ICSCROPT, ICLINES, ICCOLUMNS, etc. .

The ICRUNRS server does require a license, called the RemoteRuntime Client license (ICRUNRC) in addition to an standard Runtime license. The ICRUNRS surrogate will run under control of ICNETD under both UNIX and Windows.

When using ThinClients, the environment for ICNETD is very important as it is what is passed to the ICRUNRS startup. Especially important is ICCODEPATH, etc.

On Windows, the ICRUNRC install is a separate install question with both the runtime and full installers.

When ICRUNRS starts, it requires a console that has been enabled and has the device name of <blank>, icrunrs, matching *ip-address*, or a matching *machine-name*, set to run programs.

To use ICRUNRC you must communicate with an ICNETD from 3.30 or higher.

The ThinClient client (icrunrc) DOES NOT use ICEXEC or require a license on the client itself.

On UNIX, a very small install can be done such that only the executable icrunrc, the messages directory with system.ms, and the term directory with any needed terminal description files are all that are required. An icconfig can be used to change/update the terminal description files.

On Windows, when using ICRUNRC the IC_WINDOWS_SETFONT builtin can be used even when connected to a UNIX server to change the client font and font size.

On Windows, the user to be logged on must have the "Log on locally" privilege.

On UNIX

The ThinClient client provides seamless pass - thru support for whatever terminal type is used under the thinclient client as given by the ICTERM entry.

On Windows

The ThinClient client supports the Windows GUI environment via the ICTERM=pcwindow screen interface much as the standard GUI runtime.

The ThinClient client supports Print-Pass Thru by using the default Windows printer at the time of the Print-Pass Thru On. If no default printer or an error is received, then the data will go to the screen.

The Print Screen command is supported in the same fashion.

In addition all GUI calls to Sp2 and Formprint are supported. See the next chapter for more specific information on using ThinClient with sp2 and Formprint.

E. Using ThinClient

ThinClient client

In the Windows install, a ThinClient client option is presented to allow the ThinClient client portion of the runtime to be installed for those clients that need ThinClient access to a server computer. You are prompted for a server machine. Only Local installs are allowed. ThinClient client cannot be installed as a client install (i.e., using executables on the server.)

When the ThinClient client (icrunrc) starts, it connects to the ThinClient server on a server machine running ICNETD. (icrunrc takes a -M machine argument to specify the server machine.). The client then performs a logon to the server with a username/password just as if you were accessing a file using the ICNETD client/server support. After a good logon, the ICNETD server will start a ThinClient server (icrunrs) to provide screen communication with the ThinClient client.

On the client ensure that the following are accessible in the current directory or via PATH, ICROOT, etc:

- terminal description file (.tdi)

No licensing is required on the client side.

ThinClient server

The ThinClient server (icrunrs) is started by icnetd and runs the logon program by default. On Windows, the ThinClient server is currently installed when ICNETD is selected. On UNIX, the ThinClient server is installed by default. When the ThinClient server is invoked by ICNETD, it requests a Remote Runtime Client license and an **ICOBOL** runtime license from the license manager and then starts the COBOL program logon.cx by default. The ThinClient server uses consoles with device set to “machine-name” or “ip-address” then “icrunrs” and then (blank). The ICTERM setting is passed from the client. Note that all users that attach to ICNETD via a thinclient connection must have the "Log on Locally" privilege when the server is an 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, ICCODEPATH, ICDATAPATH, etc:

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

Once the application is running, it will make console 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 console calls sent to the client will result in a response from the end user. Each ThinClient server (icrunrs) requires a Remote Runtime Client license AND an **ICOBOL** runtime license. There is NO SEAT licensing across the network.

The ThinClient requires at least ICNETD 2.80 (ICOBOL Rev 3.30).

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 server, turn on ICNETD surrogate 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. Without this log file, all **ICOBOL** messages are lost.

B.2 On the server, turn on ICNETD server tracing (icnetd -O a). Provides more logging information in the icnetd.lg file.

To use the – switch to pass information to the ThinClient server, consider you are going to a UNIX machine called aix2 and you want to support mixed cases filenames then you can do the following:

```
icrunrc -M aix2 -- -C u
```

In this case, the -M aix is processed by the ThinClient client (icrunrc) while the -C u is passed to the server and to the ThinClient server (icrunrs) to be processed.

X. THINCLIENT (GUI)

A. Introduction

ThinClient (gui) support is provided to allow a COBOL sp2 and/or FormPrint-based application to be run on a server and deploy the application's graphical user interface to any Windows client machine connected to the server via a TCP/IP network. ThinClient (gui) clients only run on Windows (i.e., a GUI environment must be present to provide the user-interface layer.) and only provides a gui (sp2/Formprint) interface. If both a screen and a gui interface are required use the basic ThinClient client.

The main advantage to this architecture is that the business logic and database files are maintained in a central location yet end-users still have access to the graphical user interface and all its associated conveniences. Note that only the Sp2 and FormPrint interfaces are shipped to the thinclient client, all standard **ICOBOL** interfaces will be performed on the server. **NO standard ICOBOL console i/o (DISPLAY, ACCEPT, STOP lit, etc.) can be done when running ThinClient (gui).**

B. Syntax

The syntax is:

```
icthinc [-a[:aflag] | -A file|dir[:aflag]] [-h|-?] [-M machine[:port]] [-p] [-q]
   [ -- icthins-switches ] [ program [arg]... ]
```

Where icthinc switches are:

-a[:*aflag*] or -A *file|dir*[:*aflag*] (Audit)
Enables auditing (default icthinc.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)
Display help text.

-M *machine*[:*port*]] (Machine)
Specifies the remote machine as an ip-address or machine-name and an optional port to connect to. If *port* is not given, 7334 is the default. If *machine* is not given, then localhost is used.

-p (Prompt for username)
Prompt for username to pass to the server.

-q (Quiet)
Enables quiet operation.

–- (Start Icthins switches)
Two dashes (--) instructs the ThinClient client to pass all remaining switches and arguments to the ThinClient server.

Program
The COBOL program to start. If not specified then the program specified in the server’s configuration file is used. If that is not set then sp2logon is used.

Arg
Any arguments needed by *program*.

When using the Prompt for username switch, the Save option is ignored in the username/password screen. For thin clients the prompt is done immediately.

Where *icthins-switches* are:

-B b|c|i (Brand)

Specifies the default record manager brand:
b=Btrieve (Windows), c=C-ISAM (Unix), i=ICISAM (default is i)

-C l|n|u (Case)
Specifies the default Case conversion (Unix):
l=lower,n=none,u=upper (default is l)

-D *date*[:*time*] (Date/time override)
Specifies an override for the default date/time. Date/time formats are YYYYMMDD & HHMMSS

-E *var*=*val* (Environment)
Sets the environment variable var to value for icrunrs

-G drsu (General)
Specifies a General switch:
d=duplicate key I-O Status, r=RDOS-like ACCEPT, s=strict switch processing,
u=uppercase program name in ACCEPT FROM ENVIRONMENT

-i (Info)
Put out Info messages

-N beiow (No)
Specifies NO options:
b=no auto reassign, e=no embedded spaces, i=no console interrupts,
o=no OCCURS DEPENDING bounds check, w=no warnings

-q (Quiet)
Specifies Quiet operation

-s (Startup mode)
Run *program* in startup mode

-S *n* (System code)
Specifies the System code value to be returned in ACCEPT FROM ENVIRONMENT

-T *n* (Terminal)
Specifies the Terminal number is n

-U l|n|u (Username)
Convert username case:
l=lower,n=none,u=upper (default is n)

ICTHINC Environment variables:

ICROOT	Help and terminal descriptions	ICTHINC	Command line options

ICTHINS Environment variables:

ICROOT	Help and terminal descriptions	ICTHINS	Command line options
ICABORT	Abort on Timeout	ICCODEPATH	.CX file path list
ICDATAPATH	Data file path list	ICPCQDIR	Print job file directory
ICPCQFILTER	PCQ filter command	ICPERMIT_MACHINE	Servermachine[:port]
ICRUNLK	File name link file	ICTIMEOUT	Timeout Delay
ICTMPDIR	Temporary file directory	PRN,SER,PCQ	Default device units
DATAFILE	@DATA resolution	LISTFILE	@LIST resolution

The ICTHINS environment variables are from the ICNETD - ICTHINS process tree.

When ICNETD starts a ThinClient surrogate (either icrunrs or icthins), it will pass the client's ip-address and host-name in the environment variables ICREMOTEADDRESS and ICREMOTEHOST respectively. (Basically "-E ICREMOTEADDRESS=n.n.n.n -E ICREMOTEHOST=machine" on the client command line.) These can then be queried from a COBOL program 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 does look for these variables when it detects a Thinclient.

C. **Using ThinClient (gui)**

ThinClient (gui) client

In the Windows install, a ThinClient (gui) client option is presented to allow the ThinClient (gui) client portion of the runtime to be installed for those clients that need ThinClient (gui) access to a server computer. You are prompted for a server machine. Only Local installs are allowed. ThinClient (gui) client cannot be installed as a client install (i.e., using executables on the server.)

When the ThinClient (gui) client (icthinc.exe) starts, it connects to the ThinClient (gui) server on a server machine running ICNETD. (icthinc.exe takes a -M machine argument to specify the server machine.). The client then performs a logon to the server with a username/password just as if you were accessing a file using the ICNETD client/server support. After a good logon, the ICNETD server will start a ThinClient (gui) server (icthins) to provide Sp2 and FormPrint communication with the ThinClient (gui) client.

On the client ensure that the following are accessible in the current directory or via PATH, ICROOT, SP2DIR, SP2.CF, QPRDIR, QPR.CFG, etc:

- panel files
- fonts.sp2 and/or fonts.qpr
- image files
- help files
- repeat group files
- other needed .dll (jpeg)

- sp2tc.ini

No licensing is required on the client side.

ThinClient (gui) server

The ThinClient (gui) server (icthins) is started by icnetd and runs the sp2logon program by default. On Windows, the ThinClient (gui) server is currently installed when ICNETD is selected. On UNIX, the ThinClient (gui) server is installed by default. When the ThinClient (gui) server is invoked by ICNETD, it requests an SP2 runtime license from the license manager and then starts the COBOL program sp2logon.cx by default. The ThinClient (gui) server uses consoles with device set to a matching *ip-address*, a matching *machine-name*, icthins , or NUL or NULL. The ICTERM setting should be set to file. Note that all users that attach to ICNETD must have the "Log on as Batch" privilege when the server is an 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, 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 (gui) server (icthins) requires an SP2 runtime license AND an **ICOBOL** runtime license. There is NO SEAT licensing across the network.

The ThinClient (gui) requires at least ICNETD 2.70 for Windows (ICOBOL Rev 3.10) or ICNETD 2.73 for UNIX (ICOBOL Rev 3.13).

To debug ThinClient (gui) consider the following:

A. Make sure the program(s) run without ThinClient (gui) before moving to ThinClient.
B. With ThinClient (gui)
   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 surrogate tracing (icnetd -O b). This will cause icthins_(pid).lg files to be created for each icthins 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 ThinClient (gui) support can be found in the readsp2.txt file.

To use the – switch to pass information to the ThinClient (gui) server, consider you are going to a UNIX machine called aix2 and you want to support mixed cases filenames then you can do the following:

```
icthinc -M aix2 -- -C u
```

In this case, the -M aix is processed by the ThinClient (gui) client (icthinc) while the -C u is passed to the server and to the ThinClient (gui) server (icthins) to be processed.

XI. TUNING

A. Overview

Interactive COBOL has been designed to be configurable such that small systems are not burdened with large system requirements and large system are not limited by the small systems constraints. Generally the default configuration options provided by Interactive COBOL allow a basic system to be started. The tips below should be taken as a starting point to configuring a well performing system that does not include wasted memory but does allow some flexibility for growth.

B. Interactive COBOL

To get the very best possible performance try the following tips:

1) Use the fewest possible selections in ICCODEPATH with a library as the first or second selection. (helps CALL & CALL PROGRAM)

2) Use a printer control directory (ICPCQDIR) that is different from your default directory for your printer files if these are created and deleted very often to avoid creating sparse directory entries.

3) Use a COBOL library file for all your programs, at least the most often used ones. (helps CALL & CALL PROGRAM)

The System Information utility under the runtime should be viewed to help select parameters in the System Parameter menu of ICCONFIG or ICEDCFW that can be configured up or down to provide a more optimal configuration.

C. Windows

To get the very best possible performance try the following tips:

1) Keep your hard disk organized. (helps all I/O)

2) On a frequent basis check for disk fragmentation and run a compress program if needed. (Disk Defragmenter is provided with Windows 9x) (helps all I/O)

3) On a frequent basis backup your changing data.
(gives you a feeling of comfort! and security!)

D. Tips & Techniques

The following sections are intended to provide a general reference for selecting and installing systems to insure the best possible performance with the least possible confusion. It should be noted that for any particular application these guidelines may not apply; although, they can be used as a guide to help in understanding your individual performance requirements.

E. Disk Fragmentation

Disk Fragmentation is when individual files are scattered all over the disk instead of being allocated contiguously. During normal everyday use, especially the multiuser Interactive COBOL runtime environment, many files will be written and/or created during the day resulting in disk fragmentation. There are many available disk optimizers (reorganizers) that are available to remedy this situation. Disk Defragmenter ships with Windows 9x. You can also always dump all your files off the disk, reformat, and restore your files to correct fragmentation. This is much more tedious and generally uncalled for because of the disk optimizers' price/performance, and various other useful features.

It would be a very useful habit to run a weekly disk optimizer to keep your disk from being too fragmented. This could possibly be done after your weekly backup operations.

The CHKDSK utility (or SCANDISK) using a template, i.e., CHKDSK *.*, can also be used to inform you of files that have non-contiguous storage. For more on this feature of CHKDSK (or SCANDISK) please refer to your operating system manual.

F. Character Set(s)

Each individual device on the system may support different character sets. On terminal devices the terminal description files provide support for any specific key sequence to be entered as data. On output, the Interactive COBOL runtime generally treats all characters as 8-bit bytes with the range from 0 to 31 (decimal) and 128 to 159 as being unprintable control codes. The remaining characters are treated as some type of printable character. The ASCII Codes chart on page 195 shows the DG/DG International and US/PC character sets.

G. Crashes, Power Failures, or Aborts

After system crashes caused by power failures, resets, or any reason that ICRUN did not have a proper shutdown the CHKDSK or SCANDISK utility should be run on any drives that were being used by ICRUN at the time of the crash. If any lost clusters are shown then rerunning CHKDSK with the Fixup switch (/F) should be done to reclaim the disk space as .CHK files in the root(\) of the problem drive. These .CHK files are the lost areas on the disk that were in the process of being updated.

You should also check the status of all of your ICISAM files to be sure that they were not corrupted. This can be done with the ICCHECK utility using the Reliable switch (-r).

H. Redirection and Pipes

Windows provides internal functions that programs can use to receive input and display output. These are called *standard input* and *standard output*. With the Windows redirection feature these can be redirected to get or send data to a device other than the console. For more information on redirection and pipes please see your Windows manual.

Examples of how redirection can be useful are given below with the command line and an explanation of how the command works.

```
icinfo > icinfo.ls
```

Runs ICINFO with its output being redirected to the file icinfo.ls

```
iccheck >NUL
```

Runs ICCHECK with its output being redirected to the NUL device. Since ICCHECK will return an ERRORLEVEL of 1 if an error is encountered, a .BAT file can check the ERRORLEVEL to see if there were any errors with the ICCHECK.

Redirection will NOT work with ICCONFIG, ICEDCFW, or ICREVWbut will work with all other utilities. Only standard output redirection is useful as none of the utilities require input.

Redirection will work with ICRUN to capture messages that normally go to the console screen. This same thing can be accomplished with the Audit switch (-a or -A) to ICRUN.

Piping can be used to display only one screen of output at a time by sending output through a filter program like MORE. For example:

```
icinfo | more
```

Runs ICINFO and displays one screen of output and then prompts to display more by using the MORE filter.

I. Backup

Backup is a very necessary function to insure that data is not lost and/or to allow for a quicker recovery from machine and disk problems that may have corrupted your data.

One item to note is that for your most frequent backups you do not need to backup those files that do not change. Obviously the most important files are those that change on a daily basis because of additions or deletions. However, all files should be backed up periodically to insure that you have good copies in the event of a hardware or disk problem.

J. Network Tips

A good feature to use from your network, if it supports it, is to update your node's clock from the network clock whenever a node attaches to the network. This insures one set of date and time stamps on the files.

When running ICRUN, if multiple workstations share the printer control file, users at any terminal on the network will be able to see others' reports. If the reports are stored in a sharable and accessible directory others will be able to to print the reports given the appropriate access. Files that need to be shared should always be addressed by their UNC name. To create simple files with a UNC name, make sure that ICPCQDIR is set to the appropriate UNC name. For files created with a directory specifier, make sure that the program specifies a UNC name. DO NOT USE A MAPPED DRIVE as part of the name. If you do not wish to share the printer control file, each workstation (node) should have its own unique printer control file.

The Terminal number switch (-T *n*) can be used to provide a unique set of console line numbers over the entire network.

XII. ICTERM DESCRIPTIONS

A. Overview

Following is a definition of how the default terminal description entries are setup. These default definitions are stored in ICCONFIG and ICEDCFW and can be written to disk by saving each default terminal description into the default terminal description files (.TDI). Interactive COBOL uses the ICTERM entry to instruct it as to how to handle input and output to terminal lines. ICCONFIG or ICEDCFW can be used to change and add to these entries. If an entry is modified in any fashion, it is recommended that a new entry be built with a suitable name and comment noting that it differs from the defaults provided.

Included in each section is a basic input (from the keyboard) and output (to the display) section. More detail can be found by using ICCONFIG or ICEDCFW to view each terminal description or by printing a copy of the terminal description. Certain descriptions that are available in the Terminal Description menu are not included here as they pertain mainly to UNIX systems.

OUTPUT

On output the following DG codes are intercepted and cause the appropriate action on the terminal chosen.

Code(s)	Action	Code(s)	Action
\002	Reverse off	\030	Cursor right
\007	Bell	\031	Cursor left
\010	Cursor home	\032	Cursor down
\011	(tab) Space	\034	Dim ON
\012	Newline	\035	Dim off
\013	Erase to eol	\036A<n>	Set FG color
\014	Erase screen and home cursor	\036B<n>	Set BG color
\015	Carriage return	\036D	Reverse ON
\016	Blink ON	\036E	Reverse off
\017	Blink off	\036F'	Print Pass Through ON
\020	Position cursor col,line	\036F?2	Print Pass Through off
\021	Print Screen	\036F?3	Print Pass Through ON
\022	Scroll ON	\036Fa	Print Pass Through off
\023	Scroll off	\036FE	Erase screen and home cursor
\024	Underscore ON	\036FF	Erase to end of screen
\025	Underscore off	\036FJ	Select normal spacing
\026	Reverse ON	\036FK	Select compressed spacing
\027	Cursor up		

Where color *n* is Ascii: 0 - black, 1 - blue, 2 - green, 3 - cyan, 4 - red, 5 - magenta, 6 - yellow, 7 - white.

Notes:

1. Print Pass Through off when going to a non-DG terminal will not generate a Ctrl-F when the printing is finished. This is strictly a DG terminal function.

2. A newline in DG causes the cursor to move to the first column of the next line.

3. Color codes are only interpreted if Color Support (ICCOLOR) has been set to process. (Not the default).

4. Reverse codes are interpreted unless ICREVERSE has been set to Ignore or Filter.

5. If a particular terminal cannot handle an attribute (either a color or character attribute, or both) then that attribute will appear to be ignored.

6. DG terminals default to Bright mode with an attribute for DIM. Most other terminals default to Dim mode with an attribute for BRIGHT.

INPUT

The following input keyboard codes are common to all terminals unless specifically overridden.

Ctrl code	Action
Ctrl-A	Position to end (END)
Ctrl-B	Position left word
Ctrl-E	Insert Mode ON/Off
Ctrl-F	Position right word
Ctrl-I	Destructive tab (TAB)
Ctrl-J	(Newline) Accepts the entire field (ESCAPE 00)
Ctrl-M	(Carriage Return) Acts like a Newline except on DG terminals where it is a truncating terminator (ESCAPE 00)
Ctrl-N	Position back tab
Ctrl-O	Position forward tab
Ctrl-P	Position to beginning (HOME)
Ctrl-R	Delete a character
Ctrl-T	Backspace
Ctrl-U	Refresh screen
Ctrl-V	Erase to end of field
ESC (Ctrl-[ )	Sends an ESC (ESCAPE 01)

Decimal code 32 (space) - 126 (~) along with their 8-bit counterparts 160 - 254 are passed through as valid characters to the COBOL programs.

Function keys (other than ESC) are treated just as DG function keys in that they accept the entire field and terminate the ACCEPT with ESCAPE KEY set to the appropriate value. These will be defined as ESCAPE nn where nn will refer to the ESCAPE KEY returned to the COBOL program.

The following is a generalization for the keycap legends, if they exist; not all keyboards will generate a Shift state for each key.

code	action
TAB	Destructive tab
Shift-TAB	Position back tab
Ins,Insert	Insert Mode ON/Off
HOME	Position to beginning
END	Position to end
Shift-END	Position to beginning
DEL	Backspace
↑	Move to previous field (Beep at top)
↓	Move to next field (Fall out at bottom)
→	Move right a character (Beep at end of field)
←	Move left a character (Beep at beginning of field)
Shift-↑	ESCAPE 70
Shift-↓	ESCAPE 77
Shift-→	ESCAPE 71
Shift-←	ESCAPE 72

The following keys are generally ignored by Interactive COBOL unless otherwise configured.

PageUp PageDown Page Send ..

For the best support with all the DG function keys, a terminal that supports the DG emulation is preferred; otherwise not all function keys with their appropriate shift states may be available.

To be able to run on various terminal types, the use of function keys should be restricted to the lowest common denominator (i.e., the one with the fewest function keys). There are not many terminals that support the Ctrl-Shift state of function keys. There are also not many terminals that have more than 10 or 12 function keys. A good start is to use only function keys f1 - f10 in base and shifted states before moving on to additional keys.

In the following tables when the generation of a key is described, the backslash character (\) implies that the next three digits comprise an octal code returned by the key.

B. **DG**

DG type (ICTERM=dg)

Assumes a DG D200 or upward compatible terminal.

Color is supported on D220, D230, and D470 type-terminals.

Compressed mode is supported for terminals that support compressed mode.

Additional Input Keys:

legend	action
ERASE EOL (Ctrl-K)	Erase to end
ERASE PAGE (Ctrl-L)	Erase entire field
Shift-CMD-PRINT (\036\001)	ESCAPE 74
Shift-HOME (\036\010)	ESCAPE 75
CMD-PRINT (\036\021)	ESCAPE 73

Function Keys

	ESCAPE KEY nn values				Generated by \036 (plus the following)			
	Normal	Shift	Ctrl	Ctrl-Shift	Normal	Shift	Ctrl	Ctrl-Shift
F1	2	10	18	26	q	a	1	!
F2	3	11	19	27	r	b	2	"
F3	4	12	20	28	s	c	3	#
F4	5	13	21	29	t	d	4	$
F5	6	14	22	30	u	e	5	%
F6	7	15	23	31	v	f	6	&
F7	8	16	24	32	w	g	7	'
F8	9	17	25	33	x	h	8	(
F9	34	41	48	55	y	i	9	)
F10	35	42	49	56	z	j	:	*
F11	36	43	50	57	{	k	;	+
F12	37	44	51	58	|	l	<	,
F13	38	45	52	59	}	m	=	-
F14	39	46	53	60	~	n	>	.
F15	40	47	54	61	p	`	0	/
C1	62	66			\	X		
C2	63	67			]	Y		
C3	64	68			^	Z		
C4	65	69			_	[		
					(no \036 for the following)			
↑		70			\027			
↓		77			\032			
→		71			\030			
←		72			\031			

Notes:

1. When using a non-DG terminal in a DG emulation mode, make sure the terminal does NOT have transmit XON/XOFF enabled since ^S and ^Q are used as real control codes when going from the host system to the terminal.

2. When positioning past column 126 on a DG terminal (or emulation), ICRUN will use the D400 window positioning code (\036FP<col><col><line><line>) since the Ctrl-P sequence is not valid for these locations. (col and line are ASCII 0-? to represent hex values).

C. DGUNIX

DGUNIX type (ICTERM=dgunix)

Assumes a DG D217+ or upward compatible terminal.

DGUNIX is very similar to DG but no binary key sequences or command sequences are used. All sequences are in ASCII.

Compressed mode is supported for terminals that support compressed mode.

Additional Input Keys:

legend	action
DEL (\010)	Backspace
ERASE EOL (\036PE)	Erase to end
ERASE PAGE (\036PH)	Erase entire field
HOME (036PF)	Position to beginning
Shift-CMD-PRINT (\036P1)	ESCAPE 74
Shift-HOME (\036Pf)	ESCAPE 75
CMD-PRINT (\036P0)	ESCAPE 73

Function Keys

	ESCAPE KEY nn values				Generated by \036 (plus the following)			
	Normal	Shift	Ctrl	Ctrl- Shift	Normal	Shift	Ctrl	Ctrl- Shift
F1	2	10	18	26	q	a	1	!
F2	3	11	19	27	r	b	2	"
F3	4	12	20	28	s	c	3	#
F4	5	13	21	29	t	d	4	$
F5	6	14	22	30	u	e	5	%
F6	7	15	23	31	v	f	6	&
F7	8	16	24	32	w	g	7	'
F8	9	17	25	33	x	h	8	(
F9	34	41	48	55	y	i	9	)
F10	35	42	49	56	z	j	:	*
F11	36	43	50	57	{	k	;	+
F12	37	44	51	58	|	l	<	,
F13	38	45	52	59	}	m	=	-
F14	39	46	53	60	~	n	>	.
F15	40	47	54	61	p	`	0	/
C1	62	66			\	X		
C2	63	67			]	Y		
C3	64	68			^	Z		
C4	65	69			_	[		
↑		70			PA	Pa		
↓		77			PB	Pb		
→		71			PC	Pc		
←		72			PD	Pd		

D. ANSI

ANSI type (ICTERM=ANSI)

Assumes a standard ANSI or upward compatible terminal.

Color is supported if the corresponding terminal supports the ANSI color sequences of:

ESC [3*f*;4*b*m

where

f is the foreground color (0 - 7)
b is the background color (0 - 7).

colors are Ascii: 0 - black, 1 - red, 2 - green, 3 - yellow, 4 - blue, 5 - magenta, 6 - cyan, 7 - white.

Compressed mode is supported for terminals that support compressed mode.

Additional Input Keys:

```
legend      action          Generated by

PageUp      ESCAPE 63       \033[5~
PageDown    ESCAPE 65       \033[6~

Function Keys

    ESCAPE KEY nn values    Generated by
    Normal                  Normal

F1  2                       \033OP
F2  3                       \033OQ
F3  4                       \033OR
F4  5                       \033OS

F5  6                       \033[15~
F6  7                       \033[17~
F7  8                       \033[18~
F8  9                       \033[19~

F9  34                      \033[20~
F10 35                      \033[21~
F11 36                      \033[23~
F12 37                      \033[24~
```

The above values are what the Windows XP telnet client provides. (added in 3.44)

E. **ATT**

AT&T type (ICTERM=att)

Assumes an AT&T 605 or upward compatible terminal.

Color is supported if the corresponding terminal supports the ANSI color sequences.

Compressed mode is supported for terminals that support compressed mode.

Additional Input Keys:

legend	action
END	Erase to end
Shift <-	ESCAPE 72
Shift ->	ESCAPE 71
Shift-Insert	Insert ON/Off
Insert	Insert ON/Off
Shift-Up-arrow	ESCAPE 70
Shift-Down-arrow	Down-arrow

Function Keys

	ESCAPE KEY nn values		Generated by \033 (plus the following)	
	Normal	Shift	Normal	Shift
F1	2	10	Oc	OC
F2	3	11	Od	OD
F3	4	12	Oe	OE
F4	5	13	Of	OF
F5	6	14	Of	OF
F6	7	15	Oh	OH
F7	8	16	Oi	OI
F8	9	17	Oj	OJ
F9	34	41	No	NO
F10	35	42	Np	NP
F11	36	43	Nq	NQ
F12	37	44	Nr	NR
F13	38	45	Ns	NS
F14	39	46	Nt	NT

F. **FILE**

File type (ICTERM=file)

Assumes a standard 8-bit ASCII format with no control codes generated for positioning except for carriage-return, line-feed, form-feed, and spaces. No function keys are supported in this entry.

This is most useful as a detached program.

This is required for cgiCOBOL programs.

NO compressed mode support.

Additional Input Keys:

None.

G. FREEDOM

Freedom type (ICTERM=freedom)
Assumes a Freedom ONE type terminal.

NO compressed mode support.

Additional Input Keys:

Function Keys

	ESCAPE KEY nn values		Generated by Ctrl-A (followed by below) then <cr>	
	Normal	Shift	Normal	Shift
F1	2	10	\040	\141
F2	3	11	A	a
F3	4	12	B	b
F4	5	13	C	c
F5	6	14	D	d
F6	7	15	E	e
F7	8	16	F	f
F8	9	17	G	g
F9	34	41	H	h
F10	35	42	I	i
F11	36	43	J	j
F12	37	44	K	k
F13	38	45	L	l
F14	39	46	M	m
F15	40	47	N	n

H. **IBM**

IBM type (ICTERM=ibm)

Assumes an IBM 3101 or upward compatible terminal with the turnaround character set to carriage-return (\015).

Compressed mode is supported for terminals that support compressed mode.

Additional Input Keys:

legend	action
ER INP	Erase field
CLEAR	Erase Field
ERASE EOF	Erase to end
PRINT	ESCAPE 73

Function Keys

	ESCAPE KEY nn values			Generated by \033 (plus the following)		
	Normal	Shift	Ctrl-Shift	Normal	Shift	Ctrl-Shift
F1	2	10	26	a\015	!a\015	"a\015
F2	3	11	27	b\015	!b\015	"b\015
F3	4	12	28	c\015	!c\015	"c\015
F4	5	13	29	d\015	!d\015	"d\015
F5	6	14	30	e\015	!e\015	"e\015
F6	7	15	31	f\015	!f\015	"f\015
F7	8	16	32	g\015	!g\015	"g\015
F8	9	17	33	h\015	!h\015	"h\015
F9	34	41	55	i\015	!i\015	"i\015
F10	35	42	56	j\015	!j\015	"j\015
F11	36	43	57	k\015	!k\015	"k\015
F12	37	44	58	l\015	!l\015	"l\015

PA1, PA2, PA3, Del LN, Ins LN, Jump, Print Line, Pr Msg, Send.. are all Ignored.

To support character attributes, more than a 3101 terminal must be available.

I. LINUX

LINUX type (ICTERM=linux)

Assumes the master console under Linux.

Color is supported if the corresponding monitor/emulator/terminal supports the ANSI color sequences.

A file is included in the examples directory, called linuxadd.map, that increases the function key support on the Linux master console to Shift-F9 - Shift-F12, Ctrl-F1 - Ctrl-F12, and Ctrl-Shift F1 - Ctrl-Shift F12. This file is used as "loadkeys linuxadd.map".

Please see the Linux documentation on loadkeys/dumpkeys for more information.

Additional Input Keys:

Function Keys

	ESCAPE KEY nn values				Generated by \033[(plus the following)			
	Normal	Shift	Ctrl	Ctrl-Shift	Normal	Shift	Ctrl	Ctrl-Shift
F1	2	10	18	26	[A	25~	39~	51~
F2	3	11	19	27	[B	26~	40~	52~
F3	4	12	20	28	[C	28~	41~	53~
F4	5	13	21	29	[D	29~	42~	54~
F5	6	14	22	30	[E	31~	43~	55~
F6	7	15	23	31	17~	32~	44~	56~
F7	8	16	24	32	18~	33~	45~	57~
F8	9	17	25	33	19~	34~	46~	58~
F9	34	41	48	55	20~	35~	47~	59~
F10	35	42	49	56	21~	36~	48~	60~
F11	36	43	50	57	23~	37~	49~	61~
F12	37	44	51	58	24~	38~	50~	62~

J. PCWINDOW

PCWINDOW type: (ICTERM=pcwindow)

This entry is only available when running **ICOBOL** on Windows.

This entry only makes sense when running on the Master Console.

This entry uses different colors for attributes. In general, the pcwindowmono selection should be used to more properly represent selected screen attributes.

Color is supported if the corresponding monitor supports color and/or shades.

At startup, the ICLINES and/or ICCOLUMNS settings will be used to set the display to have the given number of rows and columns. If running in full-screen mode make sure that the values work with the native graphics card. At termination, the screen is returned to its original setting.

Compressed mode is supported just by setting ICCOLUMNS to min:max. The default is 80:132.

Additional Input Keys:

legend	action
PageUp	ESCAPE 63
PageDown	ESCAPE 65
Ctrl-PageUp	ESCAPE 67
Ctrl-PageDown	ESCAPE 69

Function Keys

ESCAPE KEY nn values

				Alt or
	Normal	Shift	Ctrl	Ctrl-Shift
F1	2	10	18	26
F2	3	11	19	27
F3	4	12	20	28
F4	5	13	21	29
F5	6	14	22	30
F6	7	15	23	31
F7	8	16	24	32
F8	9	17	25	33
F9	34	41	48	55
F10	35	42	49	56
F11	36	43	50	57
F12	37	44	51	58

↑ (Uparrow)	70
↓ (Downarrow)	77
→ (Rightarrow)	71
← (Leftarrow)	72

Use ICCONFIG's or ICEDCFW's Terminal Description menu to print a listing of all the supported keys.

K. PCWINDOWMONO

PCWINDOW type: (ICTERM=pcwindowmono)

This entry is only available when running **ICOBOL** on Windows and is only useful when running on the Master Console.

This entry is just like pcwindow but it is monochrome only. No attributes are mapped to colors. Colors are supported only with actual Color combinations. The default is White on black.

Two(2) ways to have a different color other than White-on-Black

A different color combination can easily be chosen by simply replacing the particular color that you wish to change in the tdi file. For example, if you want green on black just go into the Configure Color/Attribute Map and change the "White" selections to "Green" leaving everything else the same. Save to the same or a separate name like "pcwindowgreen".

or

Use the environment variables ICCOLOR=process, ICFGCOLOR=green. This will set green on black. Or also set ICBGCOLOR=white and you will have Green-on-White.

At startup, the ICLINES and/or ICCOLUMNS settings will be used to set the display to have the given number of rows and columns. If running in full-screen mode make sure that the values work with the native graphics card. At termination, the screen is returned to its original setting.

Compressed mode is supported just by setting ICCOLUMNS to min:max. The default is 80:132.

Additional Input Keys:

legend	action
PageUp	ESCAPE 63
PageDown	ESCAPE 65
Ctrl-PageUp	ESCAPE 67
Ctrl-PageDown	ESCAPE 69

Function Keys

ESCAPE KEY nn values

	Normal	Shift	Ctrl	Alt or Ctrl-Shift
F1	2	10	18	26
F2	3	11	19	27
F3	4	12	20	28
F4	5	13	21	29
F5	6	14	22	30
F6	7	15	23	31
F7	8	16	24	32
F8	9	17	25	33
F9	34	41	48	55
F10	35	42	49	56
F11	36	43	50	57
F12	37	44	51	58

↑ (Uparrow)	70
↓ (Downarrow)	77
→ (Rightarrow)	71
← (Leftarrow)	72

Use ICCONFIG's or ICEDCFW’s Terminal Description menu to print a listing of all the supported keys.

L. **VT100**

VT100 type (ICTERM=vt100)

Assumes a standard DEC VT100 or upward compatible terminal. Only four function keys are supported in this entry.

Color is supported if the corresponding terminal supports the ANSI color sequences.

Compressed mode is supported for terminals that support compressed mode.

Additional Input Keys:

Function Keys

	ESCAPE KEY nn values	Generated by
	Normal	Normal
PF1	2	\033OP
PF2	3	\033OQ
PF3	4	\033OR
PF4	5	\033OS

M. **VT220**

VT220 type (ICTERM=vt220)

Assumes a standard DEC VT220 or upward compatible terminal.

Color is supported if the corresponding terminal supports the ANSI color sequences.

Compressed mode is supported for terminals that support compressed mode.

Additional Input Keys:

Function Keys

	ESCAPE KEY nn values	Generated by
	Normal	Normal
PF1	2	\033OP
PF2	3	\033OQ
PF3	4	\033OR
PF4	5	\033OS

Function keys F6 - F20 are treated like F1 - F15

	ESCAPE KEY nn values	Generated by
		\033 (plus the following)
	Normal	Normal
F6	2	[17~
F7	3	[18~
F8	4	[19~
F9	5	[20~
F10	6	[21~
F11	7	[23~
F12	8	[24~
F13	9	[25~
F14	34	[26~
F15	35	[28~
F16	36	[29~
F17	37	[31~
F18	38	[32~
F19	39	[33~
F20	40	[34~

N. **VT220PC**

VT220 type (ICTERM=vt220pc)

Assumes a standard Windows PC using an VT220 or Ansi emulator. The Microsoft Telnet that ships with Windows XP matches this emulation. Also later Xterm emulation under Linux more closely match this emulation.

Color is supported if the corresponding terminal supports the ANSI color sequences.

Compressed mode is supported for terminals that support compressed mode.

Additional Input Keys:

Function Keys

	ESCAPE KEY nn values				Generated by \033 (plus the following)			
	Normal	Shift	Ctrl	Ctrl Shift	Normal	Shift	Ctrl	Ctrl-Shift
F1	2	10	18	26	[11~	[11;2~	[11:5~	[11:6~
F2	3	11	19	27	[12~	[12;2~	[12:5~	[12:6~
F3	4	12	20	28	[13~	[13;2~	[13:5~	[13:6~
F4	5	13	21	29	[14~	[13;2~	[14:5~	[14:6~
F5	6	14	22	30	[15~	[15;2~	[15:5~	[15:6~
F6	7	15	23	31	[17~	[17;2~	[17:5~	[17:6~
F7	8	16	24	32	[18~	[18;2~	[18:5~	[18:6~
F8	9	17	25	33	[19~	[19;2~	[19:5~	[19:6~
F9	34	41	48	55	[20~	[20;2~	[20:5~	[20:6~
F10	35	42	49	56	[21~	[21;2~	[21:5~	[21:6~
F11	36	43	50	57	[23~	[23;2~	[23:5~	[23:6~
F12	37	44	51	58	[24~	[24;2~	[24:5~	[24:6~

O. WYSE and WY50

WYSE type (ICTERM=wyse)

Assumes a WYSE 60 or upward compatible terminal with hidden attributes.

Compressed mode is supported for terminals that support compressed mode.

WYSE50 type (ICTERM=wy50)

Assumes a WYSE 50 terminal with non-hidden attributes. In this case, only a single attribute is supported. Any Blink, Bright, Reverse, or Underline attribute will use write-protect mode which can be configured on a Wyse-50 to either dim, normal, or reverse. Thus, you will be able to see two attributes (normal and protected) as long as the protect mode display attribute is NOT set to normal. We recommend that reverse be used, especially if the Screen Handler is also being used.

Write protect mode on and off are ESC ")" and ESC "(" respectively. This mode does not actual write protect the screen unless protect mode has been enabled with an ESC "&". Protect mode disable is ESC """ (single-close-quote) and is the default.

Additional Input Keys:

legend	action
CLR LINE, ERASE EOF	Erase to end
CLR SCRN, ERASE EOP	Erase field
PRINT, Prt SC	ESCAPE 73

Function Keys

	ESCAPE KEY nn values				Generated by			
					\001 (plus the following)			
				Ctrl				Ctrl
	Normal	Shift	Ctrl	Shift	Normal	Shift	Ctrl	Shift
F1	2	10	18	26	@\015	`\015	\200\015	\220\015
F2	3	11	19	27	A\015	a\015	\201\015	\221\015
F3	4	12	20	28	B\015	b\015	\202\015	\222\015
F4	5	13	21	29	C\015	c\015	\203\015	\223\015
F5	6	14	22	30	D\015	d\015	\204\015	\224\015
F6	7	15	23	31	E\015	e\015	\205\015	\225\015
F7	8	16	24	32	F\015	f\015	\206\015	\226\015
F8	9	17	25	33	G\015	g\015	\207\015	\227\015
F9	34	41	48	55	H\015	h\015	\210\015	\230\015
F10	35	42	49	56	I\015	i\015	\211\015	\231\015
F11	36	43	50	57	J\015	j\015	\212\015	\232\015
F12	37	44	51	58	K\015	k\015	\213\015	\233\015
F13	38	45	52	59	L\015	l\015	\214\015	\234\015
F14	39	46	53	60	M\015	m\015	\215\015	\235\015
F15	40	47	54	61	N\015	n\015	\216\015	\236\015

F16 ignored ..

Next, Prev, Send, Send Line, SN Msg, Del Line, Ins Line are all Ignored.

For both WYSE and WY50

The down-arrow key sends a Ctrl-J which is normally a new-line, instead with ICTERM=wyse, Interactive COBOL treats a Ctrl-J like a normal down-arrow.

Given the behavior of the down-arrow, the Enter key (which sends a Ctrl-M (carriage-return) is treated like a normal new-line and acts as an Terminator with no Erase. (Thus, there is no key on a Wyse like the DG carriage-return, i.e., that functions as a Terminator with Erase.)

The backspace key and the left-arrow key send a Ctrl-H. By default, Interactive COBOL treats a Ctrl-H like a left-arrow

For the Ctrl and Ctrl-Shift function keys, the line that the terminal is on must be running in 8-bit mode since the high order bit is determining the function key.

P. **XENIX and SCO**

XENIX or SCO type (ICTERM=xenix or ICTERM=sco)

Assumes the master console under SCO XENIX or SCO UNIX.

Many terminal emulators for PC's support an SCO master console mode.

Some terminals have an SCO or XENIX or UNIX mode that matches this description.

Color is supported if the corresponding monitor/emulator/terminal supports the ANSI color sequences.

Additional Input Keys:

Function Keys

	ESCAPE KEY nn values				Generated by \033[(plus the following)			
				Ctrl-				Ctrl-
	Normal	Shift	Ctrl	Shift	Normal	Shift	Ctrl	Shift
F1	2	10	18	26	M	Y	k	w
F2	3	11	19	27	N	Z	l	x
F3	4	12	20	28	O	a	m	y
F4	5	13	21	29	P	b	n	z
F5	6	14	22	30	Q	c	o	@
F6	7	15	23	31	R	d	p	[
F7	8	16	24	32	S	e	q	\
F8	9	17	25	33	T	f	r	]
F9	34	41	48	55	U	g	s	^
F10	35	42	49	56	V	h	t	_
F11	36	43	50	57	W	i	u	`(open)
F12	37	44	51	58	X	j	v	{

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
128	200	80		Ç
129	201	81		ü
130	202	82		é
131	203	83		â
132	204	84		ä
133	205	85		à
134	206	86		å
135	207	87		ç
136	210	88		ê
137	211	89		ë
138	212	8A		è
139	213	8B		ï
140	214	8C		î
141	215	8D		ì
142	216	8E		Ä
143	217	8F		Å
144	220	90		É
145	221	91		æ
146	222	92		Æ
147	223	93		ô
148	224	94		ö
149	225	95		ò
150	226	96		û
151	227	97		ù
152	230	98		ÿ
153	231	99		Ö
154	232	9A		Ü
155	233	9B		¢
156	234	9C		£
157	235	9D		¥
158	236	9E		₧
159	237	9F		ƒ
160	240	A0	space	á
161	241	A1	¬	í
162	242	A2	½	ó
163	243	A3	µ	ú
164	244	A4	²	ñ
165	245	A5	³	Ñ
166	246	A6	¤	ª
167	247	A7	¢	º
168	250	A8	£	¿
169	251	A9	ª	⌐
170	252	AA	º	¬
171	253	AB	¡	½
172	254	AC	¿	¼
173	255	AD	©	¡
174	256	AE	®	«
175	257	AF	‡	»
176	260	B0	»	░
177	261	B1	«	▒
178	262	B2	¶	▓
179	263	B3	™	│
180	264	B4	ƒ	┤
181	265	B5	¥	╡
182	266	B6	±	╢
183	267	B7	≤	╖
184	270	B8	≥	╕
185	271	B9	·	╣
186	272	BA	` (grave)	║
187	273	BB	§	╗
188	274	BC	° (degree)	╝
189	275	BD	¨ (umlaut)	╜
190	276	BE	´ (acute)	╛
191	277	BF	↑	┐

Dec	Oct	Hex	DGI	PC
192	300	C0	Á	└
193	301	C1	À	┴
194	302	C2	Â	┬
195	303	C3	Ä	├
196	304	C4	Ã	─
197	305	C5	Å	┼
198	306	C6	Æ	╞
199	307	C7	Ç	╟
200	310	C8	É	╚
201	311	C9	È	╔
202	312	CA	Ê	╩
203	313	CB	Ë	╦
204	314	CC	Í	╠
205	315	CD	Ì	═
206	316	CE	Î	╬
207	317	CF	Ï	╧
208	320	D0	Ñ	╨
209	321	D1	Ó	╤
210	322	D2	Ò	╥
211	323	D3	Ô	╙
212	324	D4	Ö	╘
213	325	D5	Õ	╒
214	326	D6	Ø	╓
215	327	D7	Œ	╫
216	330	D8	Ú	╪
217	331	D9	Ù	┘
218	332	DA	Û	┌
219	333	DB	Ü	█
220	334	DC	space	▄
221	335	DD	Ÿ	▌
222	336	DE	space	▐
223	337	DF	space	▀
224	340	E0	á	α
225	341	E1	à	β
226	342	E2	â	Γ
227	343	E3	ä	π
228	344	E4	ã	Σ
229	345	E5	å	σ
230	346	E6	æ	µ
231	347	E7	ç	τ
232	350	E8	é	Φ
233	351	E9	è	Θ
234	352	EA	ê	Ω
235	353	EB	ë	δ
236	354	EC	í	∞
237	355	ED	ì	φ
238	356	EE	î	ε
239	357	EF	ï	∩
240	360	F0	ñ	≡
241	361	F1	ó	±
242	362	F2	ò	≥
243	363	F3	ô	≤
244	364	F4	ö	⌠
245	365	F5	õ	⌡
246	366	F6	ø	÷
247	367	F7	œ	≈
248	370	F8	ú	°
249	371	F9	ù	∙
250	372	FA	û	·
251	373	FB	ü	√
252	374	FC	ß	ⁿ
253	375	FD	ÿ	²
254	376	FE	space	■
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. RS-232C

Two basic types of serial interfaces have been defined by the Electronic Industries Association (EIA). These are DTE (Data Terminal Equipment) and DCE (Data Communications Equipment). These conventions specify the direction of information flow for data and control signals. The names of the signals are always based on their DTE function. Each position on a DTE connector performs a function which is complementary to the corresponding position on the DCE connector. This means that connections between DTE and DCE connectors can be made pin-for-pin. When making connections of the same type, the functions of the pins rather than the number of the pins, must be matched.

The connectors on terminals are almost all DTE, while the connectors on modems are almost all DCE. The connectors from computers can be either. The serial port(s) on the back of PC and AT class machines are almost always DTE. In the PC/AT world, almost all add-on serial boards (both dumb and smart) provide a DTE connection. On many UNIX machines the connection is usually DCE. You should always check your equipment's documentation to determine if it is DTE or DCE.

To connect a DTE connection to a DCE connection using the same connector, only a straight-thru cable is needed. For example, to connect a terminal or a PC to a modem, a 25-pin straight-thru cable can be used.

The following table lists the signals which are generally available on an RS-232C DTE DB25-pin or an AT DB9-pin connector. All the examples assume a DB25-pin connector.

Function	Signal	I/O	DB25	AT-DB9
Frame Ground	GND	-	Pin 1	plug
Transmit Data	TxD	O	Pin 2	Pin 3
Receive Data	RxD	I	Pin 3	Pin 2
Request To Send	RTS	O	Pin 4	Pin 7
Clear To Send	CTS	I	Pin 5	Pin 8
Data Set Ready	DSR	I	Pin 6	Pin 6
Signal Ground	GND	-	Pin 7	Pin 5
Data Carrier Detect	DCD	I	Pin 8	Pin 1
Data Terminal Ready	DTR	O	Pin 20	Pin 4
Ring Indicator	RI	I	Pin 22	Pin 9

A typical RS-232C wiring diagram for standard hardwired terminals or serial printers not needing hardware flow control to a DTE connection can be made with the following 3-wire connection:

```
         "Simplest Serial Cable"
Computer(DTE)                Terminal(DTE)

  TxD  2 ────────────────────────→ 3
  RxD  3 ←──────────────────────── 2
   SG  7 ───────────────────────── 7
```

Some devices, especially older Data General terminals, require the 4-5 and 6-8-20 loop-back on the terminal side to go ON LINE, but most newer terminals have an option to disable this requirement. Read the documentation for the device to be connected to see if any of these loop-back schemes are required.

```
           "Null Modem Cable"
Computer(DTE)                Terminal(DTE)

  TxD  2 ────────────────────────→ 3
  RxD  3 ←──────────────────────── 2
   SG  7 ───────────────────────── 7
 RTS  4 ──┐                     ┌── 4
 CTS  5 ←─┘                     └─→ 5
 DSR  6 ←──┐                   ┌──→ 6
 DCD  8 ←──┤                   ├──→ 8
 DTR 20 ───┘                   └─── 20
```

A typical wiring diagram for devices needing hardware flow control can be made with the following 4-wire connection:

```
           "Hardware Flow Control Cable"

     Computer(DTE)                Terminal(DTE)

       TxD  2 ──────────────────────→ 3
       RxD  3 ←────────────────────── 2
        SG  7 ─────────────────────── 7
       CTS  5 ←────────────────────── 6 or 20
     DSR  6 ←───┐                Depending on how
     DCD  8 ←───┤                the printer sends
     DTR 20 ────┘                hold.
```

Hardware flow control is used when a device (usually a printer) cannot send software flow control codes. The correct pin on the printer, usually 6 (DSR) or 20 (DTR), must be wired to pin 5 (CTS) such that the computer will know to stop sending data until pin 5 (CTS) is re-asserted.

A typical wiring diagram for devices needing modem control can be made with the following 9-wire connector:

```
           "Full Modem Control"

   Computer(DTE)                Modem(DCE)

      TxD  2 ──────────────────────→ 2
      RxD  3 ←────────────────────── 3
      RTS  4 ──────────────────────→ 4
      CTS  5 ←────────────────────── 5
      DSR  6 ←────────────────────── 6
       SG  7 ─────────────────────── 7
      DCD  8 ←────────────────────── 8
      DTR 20 ──────────────────────→ 20
      RI  22 ←────────────────────── 22
```

In many cases ring indicator (RI) is not needed or supported by the host device and in some cases the Data Set Ready (DSR) signal is not needed thus allowing an 8 or even 7-wire cable.

Hardware flow control is generally required for modems with data compression (MNP 5 or V.42bis) to keep the computer from overrunning its input buffer since hardware flow control is generally much faster than software flow control.

When using high speed modems with data compression support, the baud rate must be set to at least 56000 (for V.32bis modems) or 38400 (for V.32 modems) to realize the maximum throughput.

APPENDIX C. HELPFUL HINTS

1. If serial lines are being used to run COBOL programs you cannot push to CMD.EXE. An error 128 is returned. You can push to CMD.EXE and execute a command or .bat file that does not require input and sends all its output to a listing file. For example, the command "cmd /c dir >lll" will execute the "dir" command sending its output to the file "llll" and returning to the COBOL program.

In addition, you must be careful when pushing off from serial lines in that you do not start a GUI program that requires input since there is no desktop that is available to allow for input. If this happens you will need to terminate that process from the Task Manager.

2. On Windows, to open a network printer you can use redirected LPT ports. i.e., a "NET USE LPT9 \\machine\printer" can be done and then LPT9 can be entered as the device name for an @PRN or @SER logical device.

When using shared printers in this fashion remember that they are no longer "DIRECTLY CONNECTED". There will be some buffering and/or queueing that is done by the sharing entity that will cause WRITEs to the device to act differently that a real direct connect (like a serial or parallel port that physically resides in the current machine.) In most cases a CLOSE must be done to the device to ensure that all data has been written. Also in some cases an implicit CLOSE will be done automatically (not by ICOBOL) but by the sharing entity if there is no activity for some time period (like a minute) on that file. This is done to ensure that remote processes will not tie up a network resource.

3. After installation, to change the username/password value for any configured services you must use the ICSVCMGR utility. You CANNOT do an upgrade as it will NOT prompt for a new username/password.

APPENDIX D. NOTES AND WARNINGS

1. Serial device default baud rate settings:

The runtime uses the last setting for serial devices to set up the default parameters (baud, parity, data, etc.) on an open. The MODE command can be used to perform these settings if needed. This MODE will be remembered until another setting is stored. In addition, extended open options can be used to set the needed values.

2. Interactive COBOL on Windows uses .DLLs to load portions of its needed code. All the .DLL's shipped with Interactive COBOL are dynamically loaded at load time. The following rules apply as to how .DLL are found.

A. First the system searches the set of pre-installed DLLs.
B. Next, the directory where the executable module for the current process is located is searched.
C. The current directory is searched.
D. The windows system directory is searched.
E. The windows directory is searched.
F. The directories in the PATH environment variable are searched.

If a specific .DLL cannot be located, the system terminates the process and displays a dialog box that reports the error.

The **ICOBOL** runtime requires the following .DLL's. (ICRUN.DLL). The ICBLTN.DLL is loaded if found. The SP2.DLL is loaded when required for sp2. The QPR.DLL is loaded when required for FormPrint.

ICPERMIT.EXE requires the appropriate Rainbow Sentinel .dll's if a parallel or USB protection device is being used.

ICSP2.EXE and ICQPRW.EXE require the UIB.DLL.

3. After installing Interactive COBOL, it's folder (directory) should not be renamed. To place the installation in a new directory, you should uninstall and re-install while saving any modified files.

If the installation directory is renamed, Uninstalling from the Control Panel will not work, certain shortcuts set up by the installation script will not work, and any service entries may not work.

4. When trying to get printing to work on Windows use Notepad to see what printers are available on this machine and if you can print a sample file to that printer.

This can be done by starting Notepad and then select File > Page Setup. In Page Setup select printer and then pull-down the name box to see all the valid printer names. These are the names that can be entered into the PCQ device selection in ICCONFIG. Now select the printer to send some test output to and go back to the Notepad screen. Enter some data and then select File > Print. If the data does not print, work with Windows to get it to print before preceeding to ICRUN.

ICINFO can also be used to see the default and current printer selections.

If after setup, the runtime prints to a printer with no error but no paper comes out of the printer, do the following:

A) pause the Windows print spooler,
B) re-print the file from the runtime,
C) check the Windows print spooler and make sure the file is there,
D) if so, that says that **ICOBOL** did get the data to the Windows printer, now un-pause the printer,
E) if the file "prints" and disappears from the screen but no paper comes out of the printer, then this probably means that the printer in its current state does not handle ASCII data. Its running in "GDI" mode. You can confirm this if the printer is connected to a parallel port by going to the local machine with the printer and from the command prompt do a "dir >lpt1". If nothing comes out on the printer then its a GDI printer (or

Windows printer).

When buying printers, look for MS-DOS support and/or some UNIX support. These indicate that the printer will accept standard ASCII data and not need a special print driver that only works in a Windows environment.

5. When using Printer Control Queues from the runtime, the user must have write access to the Windows spool directory. On Windows it usually defaults to "\WINDOWS\system32\spool\printers" or otherwise an access denied (exception 5) is returned when you try to print from the runtime system.

6. After installation, the Interactive COBOL screen will be shown with all the installed shortcuts. This is the best time to change any needed startup parameter by selecting the shortcut and right-clicking to properties.

7. When sharing a .pq file across the network, use UNC names to name printer files that are stored in the .pq file. Using a local file name like "drive:\directory...\filename" will only allow that file to be used by processes that have that local drive.

If a UNC name is used, all processes across the network will be able to use that file.

A workaround for simple filenames is to set the ICPCQDIR environment variable to the UNC name of the spool directory where these print files will be stored. In this way, the full UNC name will be stored in the .pq file. I.E., set ICPCQDIR=\\mainmachine\c\prints.

8. Currently, Interactive COBOL handles at most fifteen (15) character usernames. Longer usernames will generate a warning and be truncated.

9. If you are using **ICOBOL** for Windows machine networked to a Windows machine (or any LAN Manager server) then you could be experiencing some database corruption.

This can be solved by using the Registry Editor to add/change a certain parameter. Follow the steps below to make the change.

> WARNING: USING REGISTRY EDITOR INCORRECTLY CAN CAUSE SERIOUS, SYSTEM-WIDE PROBLEMS THAT MAY REQUIRE YOU TO REINSTALL WINDOWS TO CORRECT THEM. USE THIS TOOL AT YOUR OWN RISK.

A. Start the Registry Editor and go to the following subkey:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\
   Services\LanmanWorkstation\Parameters
```

B. Add or change the following:

Value name:	UseLockReadUnlock
Data Type:	REG_DWORD
Data:	0 (Default: 1)

C. Exit the Registry Editor

**** NOTE: The 2.34 and up Installation will automatically perform this function.

10. If you are using **ICOBOL** on Windows machines with network interface cards (NICs) that used a shared memory address to communicate with the computer (SMC cards for example) some customers have reported network problems when running these cards in Full Duplex mode.

These problems seem to have been cleared up by forcing the cards to Half Duplex mode. This can be done either in the Hub or Switch or by selecting the network card and disabling Auto-Neg. and setting Half Duplex.

11. If specifying COM ports on Windows, COM ports above 9 must be given as \\.\COMx. You can also specify COM1-9 with this format also. Also note that most command line programs cannot handle COM10 or greater. (See Microsoft article Q115831 - HOWTO: Specify Serial Ports Larger than COM9)

12. On Windows, if you get one of the following errors:

A. "Initialization of the dynamic library ...\system32\user32.dll failed",

B. In the ICEXEC log, you get an exit code 128 when starting a runtime,

C. In the ICNETD log, you get an exit code 128 when starting an icnets surrogate, or

D. you just cannot seem to get more than 4-10 runtimes running on serial lines or telnet sessions or icnet surrogates logged on,

Then look at the Microsoft Article ID Q184802 which gives information on updating the registry for a particular value for heap memory.

WARNING: USING REGISTRY EDITOR INCORRECTLY CAN CAUSE SERIOUS,
SYSTEM-WIDE PROBLEMS THAT MAY REQUIRE YOU TO REINSTALL
WINDOWS TO CORRECT THEM. USE THIS TOOL AT YOUR OWN RISK.

This article describes updating the following subkey under the HKEY_LOCAL_MACHINE subtree:

\System\CurrentControlSet\Control\Session Manager\SubSystems\Windows

Where there is a "SharedSection=1024,3072" in the data for this value you must replace it with "SharedSection=1024,3072,512". You must reboot for the new value(s) to take effect. The "512" causes each heap allocation to be smaller to keep from overflowing the NT Heap space which is fixed allocated. This value should increase the number of allowed processes. To further increase process count continue to decrease this number to 256 and then 128, if needed.

Please read the Microsoft Article Q184802 for more information on this setting.

NOTE: ICINFO prints out the value for this registry entry as"NT Heap:" so you can see the setting without starting regedit or regedt32.

**** The above registry entry is set automatically when ****
**** installing **ICOBOL** 2.62 or above on a Windows machine. ****

13. On Windows, automatic services must be careful in their use of network files at startup. The following rules must be followed:

A. Mapped drive names cannot be used, use a UNC name. (No drive mapping is available to the service manager.)

B. A specific username/password must be specified for the service that allows running as a service and has network access to the remote machine(s). (The default username for a service does not allow remote access of network files.) When doing a client install on a Windows machine, this "Service username/password is prompted for at installation time."

C. The service must have a dependency set that prevents it from starting until after the network is available. (In the default case, services are started in basically alphabetical order.) "When doing a client install on a Windows machine, these dependencies are set by the Install program."

ICSVCMGR can be used to set or change the username/password and the dependency settings if needed.

This applies to the .lic file for ICPERMIT and the .cfi and .pq files for ICEXEC and to any needed executables.

14. On Windows, when using standard Microsoft networking (i.e., you open files using drive letters that have been mapped to shared drives/directories on another machine) if you experience file corruption then make the following change to the Windows registry.

Follow the steps below to make the change.

WARNING: USING REGISTRY EDITOR INCORRECTLY CAN CAUSE SERIOUS, SYSTEM-WIDE PROBLEMS THAT MAY REQUIRE YOU TO REINSTALL WINDOWS TO CORRECT THEM. USE THIS TOOL AT YOUR OWN RISK.

A. Start the Registry Editor and go to the following subkey:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters
```

B. Add or change the following:

Value name:	EnableOpLocks
Data Type:	REG_DWORD
Data:	0 (Default: 1)

(This entry defaults to 1 (True) if no value is specified.)

C. Exit the Registry Editor

EnableOpLocks specifies whether the server allows clients to use oplocks (opportunistic locking) on files. Oplocks are a performance enhancement, but have the potential to cause lost cached data on some networks, particularly wide-area networks.

**** The above registry entry is set automatically when ****
**** installing **ICOBOL** 2.44 or above on a Windows machine. ****

On a Windows machine only being used as a client, opportunistic locking can be disabled just on that machine as follows. (If you have made the above change to the server machine this change is unnecessary.)

Follow the steps below to make the change.

WARNING: USING REGISTRY EDITOR INCORRECTLY CAN CAUSE SERIOUS, SYSTEM-WIDE PROBLEMS THAT MAY REQUIRE YOU TO REINSTALL WINDOWS TO CORRECT THEM. USE THIS TOOL AT YOUR OWN RISK.

A. Start the Registry Editor and go to the following subkey:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanWorkstation\Parameters
```

B. Add or change the following:

Value name:	UseOpportunisticLocking
Data Type:	REG_DWORD
Data:	0 (Default: 1)

(This entry defaults to 1 (True) if no value is specified.)

C. Exit the Registry Editor

UseOpportunisticLocking specifies whether the workstation redirector should use opportunistic-locking (oplock) performance enhancement.

ICINFO has been updated to show these values on Windows.

More information on this can be found by reading Microsoft Articles Q129202 or Q102967.

15. On Windows, if an environment entry (like ICROOT) is placed in the system environment then if you change its value you must remember to re-boot to allow all the processes (especially the service manager) to see the change. This is especially true if ICROOT was placed in the system environment and then a new revision is loaded into a different directory.

NOTE: Setting ICROOT in the system environment is generally not recommended as it is always available from the registry entry set by the **ICOBOL** Installation for all **ICOBOL** executables.

16. On Terminal Server edition, we recommend doing a full install (either server or local) and to place the configuration and printer control queue files locally.

If you do not follow the above then the following may be required:

A. You may need to run the REGISTER command with the /SYSTEM option on the icexec.exe executable if you get an error when starting the runtime about no shared area when logged in using Terminal Server Clients but the runtime works when run from the master console. For example:

```
REGISTER ICEXEC.EXE /SYSTEM
```

B. Services may not start and give a error code 4 in the Event Log. If this is true and the username/password option was prompted for in the **ICOBOL** install then the Audit directory for the services is set to a Read-Only directory for this username. Either use ICSVCMGR or REGEDIT to change the command line to the particular service to may the Audit directory a different directory. Generally we recommend setting it to "C:\WINDOWS".

Windows Terminal Server is a good platform for **ICOBOL** since the applications actually run on the server and only the screen/keyboard is run on the client.

17. Performance Hints on a network.

A. Set ICTMPDIR to a local directory.
B. If using a link file, make the link file local. (VERY IMPORTANT)
C. If using a library file, make the library file local.
D. Use ICNET to access remote files.
E. If possible, open a file EXCLUSIVELY, it will process much, much faster.

18. On Windows, when doing a client install, if the error"Unable to start runtime executive service is given" then ICEXEC was installed as a service but it could not start. See the Event log for more specifics but in most cases it will be that the username/password given on the client machine does not match that on the server.

19If the **ICOBOL** User Library is being used, the icuser.dll or icusers.dll that the application program uses must be kept in sync with the released version of **ICOBOL** installed on the machine. This can be done by manually copying the appropriate icuser.dll or icusers.dll from the icobol\user_lib release directory into the application's release directory replacing the one already there.

This is especially needed if the shared area changes from one revision to the next.

20. To have hot keys work in sp2, you just need to make the hot key be a control key for each panel by selecting Panel-> Control Keys and adding the appropriate values. For example, if F8 is the hot key and you want it to call hotkey09, you would define 322 to be a control key (value is from the sp2 manual, appendix A)

Then in the cobol program something like this:

```
call "sp2" using sp2-converse-data mainmenu-converse-data
```

```
if mainmenu-key = 322
    call "hotkey09"
end-if.
```

APPENDIX E. COMMON PROBLEMS

Review the moments just before the problem occurred to see if you changed anything.

Following is a list of general problems that some users have encountered along with our initial suggestion on what to look at in your system.

1. If you try to start an Interactive COBOL command and you get the message "Cannot run in DOS mode:" then for some reason the environment is set up to Not Detect windows. Go into Properties, Program, Advanced, and uncheck that option.

If your problems still persist fill out a Support Information Request (SIR) form using the one from the end of this manual as a guide.

!!Send a copy of the ICINFO report for all reported problems.!!

For the fastest response to an Interactive COBOL problem please follow these guidelines and if possible either e-mail or fax the needed information. The E-mail is support@icobol.com and the FAX number is (919) 851-4609.

INDEX

Support Information Request (SIR)

Support E-MAIL: support@icobol.com Date: __/__/__

Support FAX: (919) 851-4609

FROM Originator ID No:__________

Contact: __________________________ Phone: _______________

Company: __________________________ Fax: _______________

Address: __________________________ E-mail: _______________

__________________________ 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)

Host Machine Configuration

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, diskette, listings, etc.)

(Please label attachments with company, contents, format, and "# of #" (e.g. 1 of 2).

From: ______________________________ Stamp

____

____

ICOBOL Support
Envyr Corporation
#160
4904 Waters Edge Drive
Raleigh, N.C. 27606
U.S.A.

---
(Fold Here)

CUSTOMER DOCUMENTATION COMMENT FORM

Support FAX: (919) 851-4609 Date: ___/___/___

Name: ____

Title: ____

Company: ____

Street: ____

City: ____

State: ______ Zip: ____________________

Country: ____

Manual: Installing and Configuring Interactive COBOL on Windows
No: 011-00403-08

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
#160
4904 Waters Edge Drive
Raleigh, N.C. 27606
U.S.A.

(Fold Here)