

Softway Systems INTERIX[™] Release 2.2
POSIX.2 Conformance Document

© 1998 Softway Systems, Inc. All rights reserved.

INTERIX is a trademark of Softway Systems, Inc.

Microsoft and Windows are registered trademarks and Windows NT is a trademark of Microsoft Corporation.

June 1998

Table of Contents

TABLE OF CONTENTS.....	I
PREFACE.....	III
PURPOSE AND AUDIENCE.....	III
MANUAL ORGANIZATION.....	III
NOTATIONS AND CONVENTIONS	III
SECTION 1: GENERAL.....	1
1.1 GENERAL.....	1
SECTION 2: TERMINOLOGY AND GENERAL REQUIREMENTS.....	1
2.2 DEFINITIONS	1
2.4 CHARACTER SET.....	1
2.5 LOCALE	2
2.6 ENVIRONMENT VARIABLES.....	2
2.9 DEPENDENCIES ON OTHER STANDARDS.....	2
2.11 UTILITY DESCRIPTION DEFAULTS	2
2.14 TERMINAL CHARACTERISTICS	3
SECTION 3: SHELL COMMAND LANGUAGE	1
3.7 REDIRECTION.....	1
SECTION 4: EXECUTION ENVIRONMENT UTILITIES.....	2
4.1 AWK — PATTERN SCANNING AND PROCESSING LANGUAGE	2
4.2 BASENAME — RETURN NON-DIRECTORY PORTION OF PATHNAME	2
4.5 CD — CHANGE WORKING DIRECTORY.....	2
4.7 CHMOD — CHANGE FILE MODES.....	2
4.13 CP — COPY FILES.....	2
4.18 DIRNAME — RETURN DIRECTORY PORTION OF FILENAME.....	2
4.19 ECHO — WRITE ARGUMENTS TO STANDARD OUTPUT	3
4.20 ED — EDIT TEXT.....	3
4.24 FIND — FIND FILES	3
4.33 LN — LINK FILES	3
4.34 LOCALE — GET LOCALE-SPECIFIC INFORMATION	3
4.35 LOCALEDEF — DEFINE LOCALE ENVIRONMENT	3
4.36 LOGGER — LOG MESSAGES	3
4.39 LS — LIST DIRECTORY CONTENTS.....	4
4.40 MAILX — PROCESS MESSAGES.....	4
4.43 MV — MOVE FILES	5
4.45 OD — DUMP FILES IN VARIOUS FORMATS.....	5
4.48 PAX — PORTABLE ARCHIVE INTERCHANGE	5
4.55 SED — STREAM EDITOR.....	5
4.56 SH — SHELL, THE STANDARD COMMAND LANGUAGE INTERPRETER	6
4.59 STTY — SET THE OPTIONS OF A TERMINAL.....	6
4.62 TEST — EVALUATE EXPRESSION.....	6
4.63 TOUCH — CHANGE FILE ACCESS AND MODIFICATION TIMES	6
4.64 TR — TRANSLATE CHARACTERS.....	6
4.68 UNAME — RETURN SYSTEM NAME	6
SECTION 5: USER PORTABILITY UTILITIES OPTION	7

SECTION 6: SOFTWARE DEVELOPMENT UTILITIES OPTION.....	8
6.1 AR—CREATE AND MAINTAIN LIBRARY ARCHIVES	8
6.2 MAKE—MAINTAIN, UPDATE, AND REGENERATE GROUPS OF PROGRAMS.....	8
SECTION 7: LANGUAGE-INDEPENDENT SYSTEM SERVICES	10
ANNEX A: C LANGUAGE DEVELOPMENT UTILITIES	11
INDEX	12

Preface

Purpose and Audience

The purpose of this document is to meet the requirements outlined in §1.3.1.2 of the POSIX.2 Standard.

This manual is intended for programmers who are writing POSIX-conforming programs and need to know the behavior of the implementation-defined features discussed in the POSIX.2 Standard.

The conformance document shall describe the behavior of the implementation for all implementation-defined features defined in this part of ISO/IEC 9945. This requirement shall be met by listing these features and providing either a specific reference to the system documentation or providing full syntax and semantics of these features. When the value or behavior in the implementation is designed to be variable or customized on each instantiation of the system, the implementation provider shall document the nature and permissible ranges of this variation. When information required by this part of ISO/IEC 9945 is related to the underlying operating system and is already available in the POSIX.1 conformance document, the implementation need not duplicate this information in the POSIX.2 conformance document, but may provide a cross-reference for this purpose

The conformance document may specify the behavior of the implementation for those parts where this part of ISO/IEC 9945 states that implementations may vary or where features are identified as undefined or unspecified.

Manual Organization

This manual follows the POSIX.2 standard in format and structure.

Notations and Conventions

Throughout this manual, the following typographic notations are used:

- **Fixed-width font** (Courier) is used to reference operands other than options, and to provide examples of system input/output.
- *Italic* typeface is used to reference utility names and options, C language function names, and environment variables. Examples: *sh -l*, *printf()*, and *TERM*.
- **Bold** typeface is used to reference filenames, pathnames, C language data types, and environment variables.
- Braces are used to enclose symbolic constants and limits defined by various header files. Examples: {CLK_TCK}.

Section 1: General

1.1 General

INTERIX Release 2.2 conforms to ISO/IEC 9945-2:1993, which is equivalent to IEEE Std 1003.2-1992, as amended by IEEE Std 1003.2a-1993, hereafter referred to as POSIX.2. This implementation of POSIX.2 is built upon a fully conforming implementation of POSIX.1 (ISO/IEC 9945-1:1990 equivalent to IEEE 1003.1-1990). The text of this conformance document occasionally references the INTERIX POSIX.1 Conformance Document, the conformance document shipped with the INTERIX subsystem.

Section 2: Terminology and General Requirements

2.2 Definitions

2.2.2 General Terms

2.2.2.8 appropriate privileges

A process cannot acquire appropriate privileges.

2.2.2.27 byte

A byte is eight bits.

2.2.2.61 extended security controls

Microsoft Windows NT does not implement any extended security controls.

2.2.2.65 file

No other file types are supported by NTFS through the INTERIX POSIX subsystem.

2.2.2.66 file access permissions

Windows NT further provides access control lists (ACL) in NTFS as an additional mechanism for controlling file access. ACLs cannot be manipulated via the INTERIX subsystem, but an ACL can further restrict access to a file beyond its permission bits.

2.2.2.68 file group class

There are no further implementation defined members of a file's group class. It is interesting to note, however, that UIDs and GIDs share the same name space on Windows NT, which means a file can be "owned" by a group in certain situations.

2.2.2.80 file times update

No other implementation-defined timestamps are supported through the INTERIX subsystem.

2.2.2.120 parent process ID

When the lifetime of the parent process of a child process ends, the parent process ID of the child process is assigned to a special system process ID of "1". [From the INTERIX POSIX.1 Conformance Document.]

2.2.2.121 pathname

Pathnames beginning with two successive slashes indicate the file system drive on which the file resides. For example, **//A/pathname** indicates that the pathname is found on drive A:. Only upper case drive letters are supported.

2.2.2.141 read-only file system

Read-only file systems are not supported. Windows NT does not provide a mechanism to deny all write access to a file system. [From the INTERIX Release 2.2 POSIX.1 Conformance Document.]

2.4 Character Set

INTERIX Release 2.2 does not support character sets with a locking-shift encoding.

2.4.1 Character Set Description File

INTERIX Release 2.2 does not support the creation of additional character maps.

2.5 Locale

INTERIX Release 2.2 does not support locales beyond the C (POSIX) locale. There are no implementation-defined categories in the locale grammar.

2.5.2 Locale Definition

2.5.2.1 LC_CTYPE

INTERIX Release 2.2 does not support locales beyond the C (POSIX) locale.

2.5.2.5 LC_CTIME

Additional LC_CTIME (optional) keywords are not supported.

2.5.3 Locale Definition Grammar

No additional locale grammar constructs are supported.

2.6 Environment Variables

No additional semantics are defined for LANG, LC_COLLATE, LC_CTYPE, LC_MESSAGES, LC_MONETARY, LC_NUMERIC, LC_CTIME, LC_TIME.

If *PATH* is unset, or set to null, then the *exec()* call searches only the current working directory..

The implementation defined default locale is “C” if LANG is not set or set to null.

2.9 Dependencies on Other Standards

2.9.1 Features Inherited From POSIX.1

2.9.1.4 File Read, Write, and Creation

There are no other implementation defined file types that apply in this clause.

2.9.1.5 File Removal

If a current working directory of any process is removed, the process whose working directory was removed is no longer able to perform file operations in that (removed) directory.

2.11 Utility Description Defaults

2.11.5 External Influences

2.11.5.2 Input Files

There are no limits descending the directory tree in NTFS.

2.13.2 Symbolic Constants for Portability Specifications

Name	Defined
POSIX2_C_BIND	no
POSIX2_C_DEV	no
POSIX2_CHAR_TERM	yes

Name	Defined
POSIX2_FORT_DEV	no
POSIX2_FORT_RUN	no
POSIX2_LOCALEDEF	no
POSIX2_SW_DEV	yes
POSIX2_UPE	no

2.14 Terminal Characteristics

This clause only applies to implementations claiming support for the User Portability Utilities Option. The current release of INTERIX does not claim support for the User Portability Utilities Option.

Section 3: Shell Command Language

3.5.3 Variables

The shell does not set the default value of *PS1* to be anything different based upon a user's privileges.

3.7 Redirection

The file descriptors 0 through 9 are supported by the INTERIX shell.

Section 4: Execution Environment Utilities

4.1 **awk** — Pattern scanning and processing language

Manipulating ENVIRON does not effect the environment of children (*system()*). The default value of SUBSEP is \000.

4.2 **basename** — Return non-directory portion of pathname

4.2.2 Description

If the pathname begins with a double slash, the rest of the name up to the next slash is a disk designator, e.g. *//P/pathname* indicates that pathname resides on P: drive. Steps 2 through 5 of the *basename* algorithm are used.

4.5 **cd** — Change working directory

4.5.2 Description

If the *HOME* environment variable is empty or undefined, *cd* prints an error and the current working directory remains the same. If *directory* is “-” (a dash), the previous working directory (if any) becomes the current working directory, and the name of the directory is written to standard output.

4.7 **chmod** — Change file modes

4.7.2 Description

The *chmod* utility does not affect additional access control mechanisms.

4.7.7 Extended Description

The *chmod* utility cannot effect the set-user-ID-on-execution or set-group-ID-on-execution bits.

4.13 **cp** — Copy files

4.13.2 Description

NTFS does not support other file types through the POSIX subsystem. The *cp* utility does not copy special files if a *-r* option is present.

4.13.3 Options

There are no other implementation defined bits in the file mode.

4.18 **dirname** — Return directory portion of filename

4.18.2 Description

If the remaining string is *//* at step (6), then steps (7) and (8) are applied.

4.19 **echo** — Write arguments to standard output

4.19.4 Operands

If the first operand is `-n` then the *echo* utility will not write a newline at the end of output. If the any operand contains a `\` then the *echo* utility will output it as it would any other character.

4.20 **ed** — Edit text

4.20.7.3.14 List Command

The size of a byte on Windows NT is eight bits.

4.24 **find** — Find files

4.24.4 Operands

The *find* utility replaces `{}` with the current pathname for the `-exec` primary.

4.33 **ln** — Link files

4.33.2 Description

NTFS does not support other file types through the INTERIX subsystem.

4.33.4 Operands

Directories can not be linked.

4.34 **locale** — Get locale-specific Information

4.34.3 Options

Currently only the C (POSIX) locale is supported.

4.34.3 Operands

No keyword values are written for the categories `LC_CTYPE` and `LC_COLLATE`.

4.35 **localedef** — Define locale environment

4.35.2 Description

Currently, the *localedef* utility does not support the capability to define new locales.

4.36 **logger** — Log messages

4.36.2 Description

Currently only the C (POSIX) locale is supported by INTERIX.

4.39 ls — List directory contents

4.39.3 Options

No implementation-defined condition exists that will otherwise cause directory entries with names beginning with a period to be printed.

4.39.5.3 Environment Variables

If *COLUMNS* is not set or is invalid, then the *ls* utility shall use a column width that will accommodate the longest directory entry name found.

4.39.6.1 Standard Output

Output to console windows accommodates as many entries as possible per line to allow all directory entry names to appear in left aligned columns.

Character special and block special files show a file size of 0 in a *-l* long listing.

4.40 mailx — Process messages

4.40.5.1 Standard Input

Standard input lines beginning with a tilde (~) character invoke *mailx* commands as described in the system documentation (see the reference page for *mailx*, *INTERIX Commands and Utilities Reference*).

4.40.5.3 Environment Variables

If the *DEAD* environment variable is not defined as */dev/null*, the variable is interpreted as the pathname of the file in which to save partial messages in case of interrupts or delivery errors.

If *EDITOR* is undefined, the default editor for the *edit* and *~e* commands is *ex*.

If the *PAGER* variable is unset or null, *mailx* uses the default paginator, *more*.

4.40.6.3 Output Files

Messages that are not text messages are subject to the following interpretation. The first cntrl-Z character is treated as the end of file. Any <carriage-return><line-feed> combinations are converted to a <line-feed>.

4.40.7 Extended Description

This subclause (4.40.7 to 4.40.7.3) does not apply, because INTERIX Release 2.2 does not claim conformance to the User Portability Utilities option. However, INTERIX Release 2.2 supports all of the functionality described in these subclauses.

4.40.7.1 mailx Internal Variables

If *crt* is set to null, rather than being unset, *mailx* always invokes the command specified by *PAGER* to display the message.

4.40.7.2.8 Edit messages

If *EDITOR* is undefined, the default editor is *ex*.

4.40.7.3 Command Escapes

The *~h* command does not prompt for any values other than To, Subject, Cc, and Bcc.

4.43 mv — Move files

4.43.2 Description

NTFS does not support other file types through the INTERIX subsystem.

4.45 od — Dump files in various formats

4.45.7 Extended Description

The default number of bytes transformed by output type specifiers correspond to the various C-language types as described in the Microsoft Visual C/C++ environment for Windows NT.

The byte order used when interpreting numeric values is little-endian. The size of a byte on Windows NT is eight bits. There is no support for multi-byte characters at this time.

4.48 pax — Portable archive interchange

4.48.2 Description

NTFS does not support other file types through the INTERIX subsystem.

The default output archive format is *ustar*.

For archives that span files, the *pax* utility determines the next file to read or write by prompting to */dev/tty* for filenames, and reading from */dev/tty*. If there is no controlling terminal, *pax* exits.

4.48.3 Options

No other implementation defined option characters are supported for the *-p* option. No other implementation defined file characteristics are supported.

INTERIX *pax* also supports *bcpio*, *sv4cpio*, and *sv4crc*.

4.48.5.2 Input Files

INTERIX *pax* also supports *bcpio*, *sv4cpio*, and *sv4crc*.

4.48.6.1 Standard Output

INTERIX *pax* also supports *bcpio*, *sv4cpio*, and *sv4crc*.

4.48.6.3 Output Files

INTERIX *pax* also supports *bcpio*, *sv4cpio*, and *sv4crc*.

4.55 sed — Stream editor

4.55.7.3 Editing Commands

A byte is eight bits on Windows NT.

4.56 sh — Shell, the standard command language interpreter

4.56.5.3 Environment Variables

The default value for *PS1* does not change depending upon user privileges.

4.59 **stty** — Set the options of a terminal

4.59.2 Description

The *stty* utility reports the characteristics and settings obtained by *tcgetattr()*.

4.59.4.4 Local Modes

No implementation defined special control characters are supported for the `exten` setting.

4.62 **test** — Evaluate expression

4.62.4 Operands

The *test* utility also supports the `-a` (logical AND) and `-o` (logical OR) primary operators, as well as “(” and “)”.

4.63 **touch** — Change file access and modification times

4.63.3 Options

The “end of time” based on `time_t` is Mon Jan 18 22:14:07 2038.

4.64 **tr** — Translate characters

4.64.7 Extended Description

The length of a byte is eight bits on Windows NT.

4.68 **uname** — Return system name

4.68.2 Description

The INTERIX Release 2.2 *uname* utility writes the information returned by the INTERIX POSIX.1 *uname()*.

4.68.3 Options

The node name within the current Windows NT networking environment is displayed for the `-n` option. Windows NT supports many networking protocols and devices. The machine’s assigned name, regardless of the network architecture, is returned.

INTERIX Release 2.2 returns the complete Windows NT release number as the release level and the service pack level for the current version level.

Section 5: User Portability Utilities Option

The User Portability Utilities Option is not supported in its entirety, and conformance is not claimed to this option.

Complete support is provided for the following utilities described in this section of POSIX.2:

<i>crontab</i>	<i>ex</i>	<i>nm</i>	<i>tput</i>
<i>csplit</i>	<i>expand</i>	<i>patch</i>	<i>unexpand</i>
<i>ctags</i>	<i>file</i>	<i>ps</i>	<i>uudecode</i>
<i>df</i>	<i>man</i>	<i>split</i>	<i>uuencode</i>
<i>du</i>	<i>more</i>	<i>strings</i>	<i>vi</i>

Support for the following utilities described in this section is not provided:

<i>alias</i>	<i>fc</i>	<i>newgrp</i>	<i>time</i>
<i>at</i>	<i>fg</i>	<i>nice</i>	<i>unalias</i>
<i>batch</i>	<i>jobs</i>	<i>renice</i>	<i>who</i>
<i>bg</i>	<i>mesg</i>	<i>talk</i>	<i>write</i>

The commands *alias*, *bg*, *fc*, *fg*, *jobs*, and *unalias* are provided as built-ins to *sh*, but not as separate utilities.

Section 6: Software Development Utilities Option

The Software Development Option is supported in this release, and conformance is not claimed to this option. However, the utilities *ar*, *make*, and *strings* are provided in INTERIX Release 2.2. The utilities *ar* and *strings* provide the functionality required by the standard. The *make* utility does not provide the .POSIX special target.

6.1 Ar—Create and maintain library archives

6.1.3 Options

If the **-r** option is given, no *file(s)* are specified and the *archive* exists, the *ar* utility exits without error.

6.1.4 Operands

If two or more *file* operands have the same basename, both are added to the archive.

6.1.6.3 Output Files

The *ar* produces archives in the COFF file format.

6.2 Make—Maintain, update, and regenerate groups of programs

6.2.7 Extended Description

6.2.7.1 Makefile Syntax

After searching for **./makefile** and **./Makefile** are found, *make* searches for the file **./depend**.

6.2.7.2 Makefile Execution

The *make* utility adds two other variables to the environment of *make*: **.CURDIR**, stores the path to where *make* was executed, and **.OBJDIR** stores the path to the directory where the targets are built.

If the **MAKEFLAGS** environment variable is not set in the environment in which *make* was invoked or in the makefile or on the command line, *make* creates it as described in this subclause.

The *make* utility does not add any options.

6.2.7.3 Target Rules

The *make* utility performs no special processing on targets containing the characters “%” and “”.”.

Defining **SHELL** in the makefile or on the command line has not effects other than those listed in this subclause.

The *make* utility processes makefiles as specified by the clause even if the .POSIX target does not appear.

6.2.7.4 Macros

Internal macros (as defined in 6.2.7.7) are not accepted from the command line.

Section 7: Language-Independent System Services

The Language-Independent System Services are not supported in this release, and conformance is not claimed to this option.

Annex A: C Language Development Utilities

Conformance to Annex A is not claimed. The utilities mentioned in Annex A (*c89*, *lex*, and *yacc*) are provided in INTERIX Release 2.2 and perform as described. The *lex* utility has minor differences that prevent it from conforming. (For a list, refer to the *lexdoc* man page, available on-line and in the *INTERIX Commands and Utilities Reference*.)

Index

/dev/tty, 5
access control lists (ACLs), 1
appropriate privileges, 1
bcpio, 5
block special files, 4
byte, size of, 1
cd, 2
character special files, 4
chmod, 2
COLUMNS environment variable, 4
cp, 2
DEAD, 4
dirname, 2
EDITOR, 4
exec(), 2
find, 3
HOME, 2
ln, 3
locale, 3
localedef, 3
logger, 3
ls, 4
mailx, 4
more, 4
PAGER, 4
PATH, 2
pax, 5
POSIX2_C_BIND, 2
POSIX2_C_DEV, 2
POSIX2_CHAR_TERM, 2
POSIX2_FORT_DEV, 3
POSIX2_FORT_RUN, 3
POSIX2_LOCALEDEF, 3
POSIX2_SW_DEV, 3
POSIX2_UPE, 3
PS1, 1, 5
stty, 6
sv4cpio, 5
sv4crc, 5
system(), 2
tcgetattr(), 6
test, 6
 logical AND and logical OR, 6
timestamps, 1
tr, 6
uname, 6
uname(), 6