AIX Version 7.1

Keyboard Technical Reference

SC23-6725-00

AIX Version 7.1

Keyboard Technical Reference

SC23-6725-00

Note

Before using this information and the product it supports, read the information in “Notices,” on page 9.

First Edition (September 2010)

This edition applies to AIX Version 7.1 and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

About This Book

This book provides programmers with complete, detailed information about keyboard layouts and translation tables for national languages supported by the AIX® operating system. Translation tables are listed alphabetically, and complete descriptions are given for the supported languages. Also included is a key to abbreviations used in the tables and an explaination of keyboard states with valid and invalid key sequences.

Highlighting

The following highlighting conventions are used in this book:

Bold
Identifies commands, subroutines, keywords, files, structures, directories, and other items whose names are predefined by the system. Also identifies graphical objects such as buttons, labels, and icons that the user selects.

Italics
Identifies parameters whose actual names or values are to be supplied by the user.

`Monospace`
Identifies examples of specific data values, examples of text similar to what you might see displayed, examples of portions of program code similar to what you might write as a programmer, messages from the system, or information you should actually type.

Case-Sensitivity in AIX®

Everything in the AIX® operating system is case-sensitive, which means that it distinguishes between uppercase and lowercase letters. For example, you can use the **ls** command to list files. If you type `LS`, the system responds that the command is "not found." Likewise, **FILEA**, **FiLea**, and **filea** are three distinct file names, even if they reside in the same directory. To avoid causing undesirable actions to be performed, always ensure that you use the correct case.

ISO 9000

ISO 9000 registered quality systems were used in the development and manufacturing of this product.

Related Publications

The following books contain information about or related to keyboards:

- *AIX Version 7.1 Commands Reference, Volume 1*
- *AIX Version 7.1 Commands Reference, Volume 2*
- *AIX Version 7.1 Commands Reference, Volume 3*
- *AIX Version 7.1 Commands Reference, Volume 4*
- *AIX Version 7.1 Commands Reference, Volume 5*
- *AIX Version 7.1 Commands Reference, Volume 6*
- *AIX Version 7.1 General Programming Concepts: Writing and Debugging Programs*
- *AIX Version 7.1 Kernel Extensions and Device Support Programming Concepts*

Keyboard Overview

The operating system supports two different types of keyboards: X server and low function terminal (LFT) keyboards. Although these two keyboard maps appear to be the same, they are separate and distinct.

An X server has an attached keyboard. The server uses mapping tables to manage the mapping of keyboard events. The mapping of an X server keyboard can be changed by using the **xmodmap** command. This command converts the keyboard so that it returns the key symbol supported by this system. The **xmodmap** command can also be used to switch keys to more convenient locations.

You can use several commands and operations to control the mapping of LFT keyboards. You can specify which keyboard map is to be used by the LFT subsystem. You can also add new keyboard maps or change existing maps.

The operating system supports three types of keyboards: the 101-key keyboard, 102-key keyboard, and 106-key keyboard. Each keyboard differs slightly in its layout and function.

Understanding Keyboard States

The keyboard mapping table maps a key position to an ASCII character, extended character, function, or string of characters. Each key on the keyboard has a numeric position code that is combined with the keyboard state when the key position is reported.

Available keyboard states are:

- Base
- Shift
- Control
- Alternate
- Alternate Graphics
- Kana Base
- Kana Shift

The operating system supports three types of keyboards:

101-key The 101-key keyboards have all available states. However, the Alt-Graphics state and the Alt state are identical and the Kana Base state and the Kana Shift state are identical.

102-key The 102-key keyboards have all of the available states except the Kana Base state and the Kana Shift state.

106-key The 106-key keyboards have all of the available states except the Alt-Graphics state. Only the right Alt key is available.

Depending on the keyboard, some of these keys are governed by the Caps Lock key or the Shift Lock key.

On keyboards that support the Caps Lock key, Caps Lock affects only those keys whose Shift state yields the uppercase character (A, B, C) of the Base state lowercase character (a, b, c) of the key. On keyboards that support the Shift Lock key, Shift Lock has the same effect as pressing a key while the Shift key is pressed.

Each of the hardware keyboards can produce some, but not all, of these states.

A software keyboard is selected at installation. A customized keyboard can be used as the system default after keyboard reconfiguration. The workstation must be restarted before the customized keyboard can be used.

The following keys are not redefinable by the keyboard device driver:

Note: The Caps Lock key can be redefined for the 106-key keyboard using the **chhwkbd** command.

Table 1. Keys with Predefined Functions

Key Position	Function	States That Cannot Be Remapped
30	Caps Lock key or Shift Lock key	All states
44	Left Shift key	All states
57	Right Shift key	All states
58	Control key	All states
60	Left Alt key	All states
62	Right Alt key	All states
64	Action key	Shift, Control, Alternate, and Alternate Graphics
90	Num Lock key	Base and Shift states
133	Hiragana	All states

Understanding Key Sequences

Most keying is done with either one-key or two-key sequences. For example, the `a` character is most often produced by one key (the A key) and the `A` character by two keys (Shift-A key sequence). If more than one state key is pressed when a character is keyed, (for example, Ctrl-Shift-A) only one state key affects the conversion of the character. With the Ctrl-Shift-A key sequence, the control state takes precedence over the shift state.

Some three-key sequences have special meanings for this operating system. The following keystroke combinations start the indicated system function. The notation Pad*n*, where *n* is a digit, indicates the *n* key on the numeric keypad to the right of the main keyboard area.

Note: Functions started with the Alt-*key* (or Shift-*key*) sequence can be selected with either the left or right Alt key (or Shift key).

There are two types of key sequences that have special meaning for this operating system:

- Kernel debugger key sequences
- System dump key sequences

Note: See "System Dump Facility" in *AIX Version 7.1 Kernel Extensions and Device Support Programming Concepts* before attempting to use any of the key sequences that perform system dumps.

The following key sequences issue special instructions to the operating system:

Ctrl-Alt-Pad4
Invokes the kernel debugger.

Ctrl-Alt-Pad1
Performs a system dump to the primary device. This key sequence works only from the native keyboard.

Ctrl-Alt-Pad2
Performs a system dump to the secondary device. This key sequence works only from the native keyboard. Supports dumping to a logical volume or tape. Requires user intervention in releases prior to AIX® 4.2.1.

Understanding Nonspacing Characters

A *nonspacing character sequence* is a two-key sequence consisting of a diacritic mark followed by an alphabetic character.

Valid Sequences

Valid nonspacing character sequences are restricted to combinations of diacritical marks and alphabetic characters. Nonspacing character sequences are folded into a single character before passing the keyboard input to the application.

A special case exists when the nonspacing character sequence consists of a diacritic mark followed by a space. In this case, the diacritic mark is displayed and sent to the application.

A valid nonspacing character sequence causes a single accented character to be returned.

Examples of a valid nonspacing character are:

Valid Nonspacing Character Examples		
1st Key Pressed	**2nd Key Pressed**	**Returned**
Grave	e	e Grave - 1 character
Grave	Space	Grave accent - 1 character

Invalid Sequences

If the nonspacing character sequence is not valid, the LFT subsystem passes the nonspacing character to the application followed by the second character of the sequence. Nonspacing character sequences that are not valid include sequences that start with one of the following three parameters:

- A nonspacing character followed by an alphabetic character. (The resulting diacritical mark does not exist in the system.)
- A nonspacing character followed by a nonalphabetic character (numeric, control, and function key).
- A nonspacing character followed by another nonspacing character.

A nonspacing character sequence that is not valid returns the accent character, followed by the code for the key pressed after the nonspacing key.

Examples of an not valid nonspacing character are:

Not valid Nonspacing Character Examples		
1st Key Pressed	**2nd Key Pressed**	**Returned**
Grave	z	Grave accent - 2 z characters
Acute	PF1	Acute accent (0xef) - 1 character PF1 (0x1b5b313731xx)

An not valid nonspacing character sequence (*nonspacing character - nonspacing character*) causes the first nonspacing character of the sequence to be passed to the application. The next nonspacing character starts a new nonspacing character sequence.

Understanding Keyboard Table Information

The keyboard table:

- Specifies the key assignments to be used on keyboards.
- Provides a link to the symbol engraved on the key.
- Lists the character sequences that are to be produced.

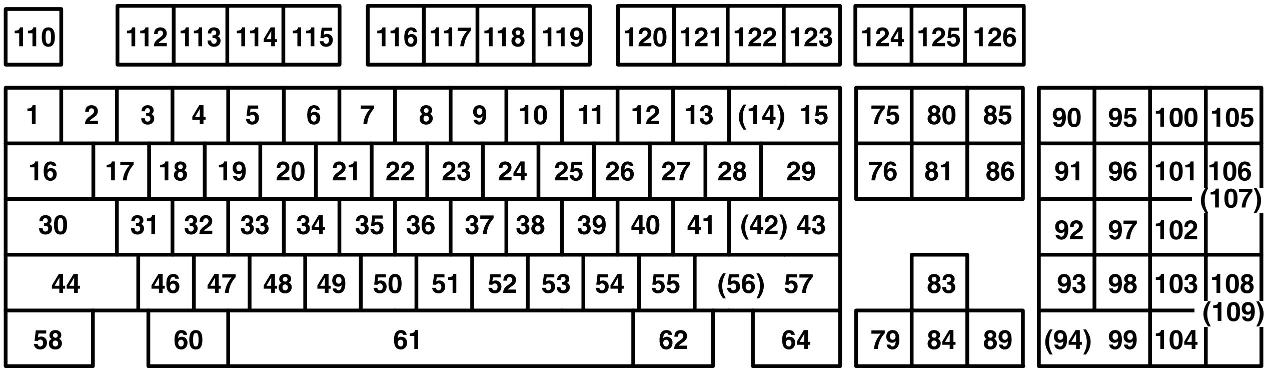
Refer to the following listing for the meanings of the column headings:

Heading	**Meaning**
Key Posn	Keyboard key position.
Shift State	The shift state of the Base, Shift, Ctrl, Alt, or AltGr positions.
Assignment Returned	The character or control assigned to that key.
Returned String	Specifies the data that is returned to the program reading the keyboard.

The Alt key, followed by one or more numbered keys on the numeric pad, returns a single character when the value is entered on the numeric pad. The value accumulates while the Alt key is held down and returns when the key is released.

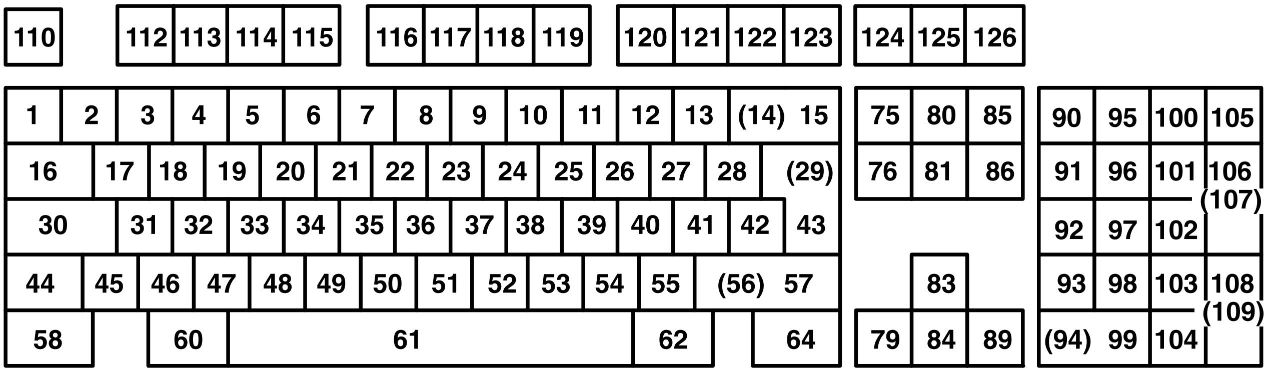
Key Position Codes and Scan Codes for Keyboards

The diagrams depict the key position codes for the 101-key keyboard, the 102-key keyboard, and the 106-key keyboard, respectively.



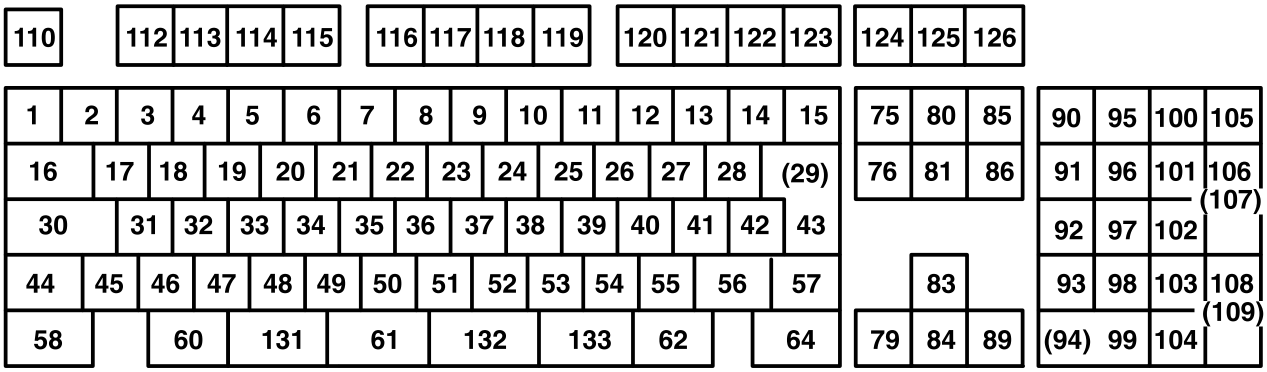
101-Key Keyboard Position Codes

. *This illustration shows the key position codes for the 101-key keyboard.*



102-Key Keyboard Position Codes

. *This illustration shows the key position codes for the 102-key keyboard.*



106-Key Keyboard Position Codes

. *This illustration shows the key position codes for the 106-key keyboard.*

The following table contains the keyboard scan codes for all keyboards. Each key on the keyboard is assigned a unique 8-bit scan code that is sent when the key is pressed.

Table 2. Key Positions and Their Scan Codes

Key Posn	**Scan Code**									
1	0x0e	23	0x3c	45	0x13	67	N/A	89	0x6a	111	N/A
2	0x16	24	0x43	46	0x1a	68	N/A	90	0x76	112	0x07
3	0x1e	25	0x44	47	0x22	69	N/A	91	0x6c	113	0x0f
4	0x26	26	0x4d	48	0x21	70	N/A	92	0x6b	114	0x17
5	0x25	27	0x54	49	0x2a	71	N/A	93	0x69	115	0x1f
6	0x2e	28	0x5b	50	0x32	72	N/A	94	0x68	116	0x27
7	0x36	29	0x5c	51	0x31	73	N/A	95	0x77	117	0x2f
8	0x3d	30	0x14	52	0x3a	74	N/A	96	0x75	118	0x37
9	0x3e	31	0x1c	53	0x41	75	0x67	97	0x73	119	0x3f
10	0x46	32	0x1b	54	0x49	76	0x64	98	0x72	120	0x47
11	0x45	33	0x23	55	0x4a	77	N/A	99	0x70	121	0x4f
12	0x4e	34	0x2b	56	0x51	78	N/A	100	0x7e	122	0x56

Table 2. Key Positions and Their Scan Codes (continued)

Key Posn	Scan Code										
13	0x55	35	0x34	57	0x59	79	0x61	101	0x7d	123	0x5e
14	0x5d	36	0x33	58	0x11	80	0x6e	102	0x74	124	0x57
15	0x66	37	0x3b	59	N/A	81	0x65	103	0x7a	125	0x5f
16	0x0d	38	0x42	60	0x19	82	N/A	104	0x71	126	0x62
17	0x15	39	0x4b	61	0x29	83	0x63	105	0x84	127	N/A
18	0x1d	40	0x4c	62	0x39	84	0x60	106	0x7c	128	N/A
19	0x24	41	0x52	63	N/A	85	0x6f	107	0x7b	129	N/A
20	0x2d	42	0x53	64	0x58	86	0x6d	108	0x79	130	N/A
21	0x20	43	0x5a	65	N/A	87	N/A	109	0x78	131	0x20
22	0x35	44	0x12	66	N/A	88	N/A	110	0x08	132	0x28
										133	0x30

Text Fonts for the Operating System

A set of precompiled text fonts are supplied with AIX® Version 4. The set is for all display devices supported by the operating system. Font definitions can be supplied to the LFT by configuring new font files into the system. This is done by issuing the **mkfont** command. The fonts supplied are compatible with Enhanced X-Windows and AIX® Computer Graphics Interface Toolkit/6000.

Text Font Format

The text font definition file has four major parts in the following sequence:

1. A font header that describes the font. The header is the same for all text fonts.
2. A set of character descriptions and lookup information to find the glyph data in the font.
3. Glyph data.
4. Property strings.

Font Header

The font header is a structure common to all fonts for all display screens. It is a fixed length. This structure is called *aixFontInfo* and is defined in the **aixfont.h** file. It contains the following members:

version1 This field contains the version stamp.

allExist For the number of glyph sets indicated below, all glyphs exist.

drawDirection Used for stroke fonts only.

noOverlap The glyph has a border zone that contains no glyph data. Indicated as true if max(rightSidebearing-characterWidth) <= minbounds->metrics.leftSideBearing.

constantMetrics If true, indicates that all glyphs are a constant size.

terminalFont True if constant metrics && leftSideBearing = = 0 && rightSideBearing = = characterWidth & & ascent = = fontAscent & & descent = = fontDescent.

linear True if firstRow = = lastRow.

constantWidth Indicates that all glyphs are the same width.

inkInside Indicates whether all character information is contained within a distinct box.

inkMetrics Ink metricw ! = bitmap metrics used with terminalFont.

strokes Glyphs are strokes, not rasters.

firstCol Defines the first Column of ink data within the glyph.

lastCol Defines the last column of ink data within the glyph.

firstRow Defines the first row of ink data within the glyph.

lastRow Defines the last row of ink data within the glyph.
nProps Indicates the number of aixFontProp structures to follow.
lenStrings Total length in bytes of all the property strings.
chDefault Default character.
fontDescent Extent below baseline for spacing; minimum for quality typography.
fontAscent The extent above baseline for spacing; minimum for quality typography.
minbounds MIN of glyph metrics over all characters in the font.
maxbounds MAX of glyph metrics over all characters in the font.
pixDepth Intensity bits per pixel.
glyphSets Number of glyph sets.
version2 Version stamp double-check.

Character Descriptions

Character descriptions contain offsets into the character data structure, also known as character glyphs. Because the data bytes within the data stream are used to access the character descriptions, there must be 256 entries. Entries for which a character is not defined should be set to offset values of a valid default character. One such valid offset is zero. Because each font has at least one character defined, there is always a first character pointed to by the offset zero, the first character in the glyph data.

The character data, or glyph, can be drawn relative to any point in a given x, y coordinate system. The `pel box` is the area where the glyph is positioned on the display screen when the font is used. The character description information is contained in the **aixCharInfo** structure, as defined in the **aixfont.h** file. This structure contains the following elements:

leftSideBearing Character origin to the left edge of the raster. Assuming that this character's pel box is referenced from position (x, y), the left vertical edge of the pel box is located at x+ leftSideBearing.
rightSideBearing Character origin to the right edge of the raster. Assuming that this character's pel box is referenced from position x, y, the right vertical edge of the pel box is located at x+ rightSideBearing.
characterWidth Advance to next character origin. Assuming that this character's pel box is referenced from position (x, y), the origin for the next character is at point (x + characterWidth,y).
ascent Baseline to the top edge of the raster. Assuming that this character's pel box is reference from position (x, y), the upper horizontal edge of the pel box is located at y - ascent.
descent Baseline to the bottom edge of the raster. Assuming that this character's pel box is reference from position (x, y), the lower horizontal edge of the pel box is located at y + descent.
attributes Must be zero.
byteOffset Byte offset of raster from the beginning of the glyph data.
exists True if glyph exists for this character.

Glyph Data

The glyph data includes information pertinent to each character in the font. The information for each character is a set of bits representing the character image.

Property Strings

name Offset of a string
value A number or a string offset
indirect Value is a string offset.

Display Symbols

Display symbols are the set of character symbols that can be displayed on an LFT.

Each character code passed in KSR data is translated into one of 256 10-bit display symbol codes. Display symbols 0 through 31 (0x1f) represent control functions and have no graphic representations. The display symbols 32 (0x20) through 126 (0x7e) represent the 7-bit ASCII range of characters. These characters, with minor exceptions, are invariant across all of the display symbols. The display symbols 128 (0x80) through 255 (0xff) vary based on the selected font. These codes are predefined to be common across all shared terminals. See *AIX® Version 7.1 General Programming Concepts* for an illustration of each of the supported code sets.

Related Information

Understanding Keyboard Mapping.

The **chhwkbd** command, the **mkfont** command and the **xmodmap** command.

Appendix. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Dept. LRAS/Bldg. 903

11501 Burnet Road
Austin, TX 78758-3400
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

(c) (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs.
(c) Copyright IBM Corp. _enter the year or years_.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at Copyright and trademark information at www.ibm.com/legal/copytrade.shtml.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Index

Printed in U.S.A.

SC23-6725-00