

IBM(R) Host Access Toolkit Version 6



Getting Started

IBM(R) Host Access Toolkit Version 6



Getting Started

Note

Before using this information and the product it supports, read the information in "Chapter 6. Notices" on page 13.

Third Edition (September 2001)

This edition applies to the Host Access Toolkit component of IBM® Host On-Demand Version 6.0, and to all subsequent releases and modifications until otherwise indicated in new editions.

© Copyright International Business Machines Corporation 2000, 2001. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About this book	v	Chapter 4. Migrating from a previous version of Host On-Demand	7
Chapter 1. Introducing Host Access Toolkit	1	Chapter 5. Introducing the Host Access Toolkit Application Programming Interfaces	9
What's new in Version 6	2	Introducing Host Access Class Library	9
Chapter 2. Planning for Host Access Toolkit	3	Introducing Host Access Beans	10
Supported operating systems	3	Descriptions	10
Disk space requirements	3	Events	11
Supported browsers	3	Introducing Host On-Demand J2EE Connector	11
Java 2 Support.	4	Chapter 6. Notices.	13
Chapter 3. Installing Host Access Toolkit 5		Trademarks	15

About this book



This icon is used to denote facilities unique to Host On-Demand.



This icon is used to denote facilities unique to the Screen Customizer.

Note: You can directly link to some of the other toolkit documents (for example, the Host Access Class Library Reference) referenced below; however, if you are viewing the PDF version of this book, you must do so using the browser plugin capability of the Adobe Acrobat Reader.

Chapter 1. Introducing Host Access Toolkit

The Host Access Toolkit consists of the Host Access Class Library, the Host Access Beans for Java™, and the Host Access ActiveX controls. These are APIs that help programmers write custom applications to either run with the IBM Host On-Demand (hereafter referred to as Host On-Demand) product or run standalone as a custom host access application.

Additional components of the Host Access Toolkit are available as part of the IBM Screen Customizer product. If both the Host Access Toolkit and IBM Screen Customizer product are installed on the same system, the Screen Customizer Component Interface and Custom Terminal JavaBean are available as part of the Host Access Toolkit.

The Host Access Class Library provides a core set of classes and methods that allow the development of platform-independent applications that can access host information without the need for a graphical display. The library represents an object-oriented abstraction of a host connection that includes reading and writing the host presentation space, enumerating the fields in the presentation space, reading the operator information area (OIA) for status information, transferring files, and performing asynchronous notification of significant events. See *Introducing Host Access Class Library* for a brief description. For complete information about the Host Access Class Library, refer to the *Host Access Class Library* Reference publication.

The Host Access Beans provide emulator functions as a set of JavaBeans™ that developers can use to rapidly develop custom applications that deliver the specific functions they want. See *Introducing Host Access Beans* for a brief description. For complete information about the Host Access Beans, refer to the *Host Access Beans for Java* Reference publication.

The Host On-Demand J2EE Connector provides access to 3270, 5250, Customer Information and Control System (CICS), and Virtual Terminal (VT) hosts from the Internet. The Host On-Demand J2EE Connector is a Java programming interface which conforms to the J2EE Connector Specification, Version 1.0 - Proposed Final Draft 2 from Sun Microsystems, and the Common Connector Framework (CCF) infrastructure required by VisualAge® for Java - Enterprise Access Builder and Component Broker. Refer to the *Host On-Demand J2EE Connector* Reference publication for more information.



IBM Screen Customizer for Host Integration (Screen Customizer) provides a graphical user interface for IBM Host On-Demand or IBM Personal Communications 3270 and 5250 sessions. Extensive customization capability is provided that does not require programming or modification of host applications. When Screen Customizer is installed, the Screen Customizer Component Interface and Custom Terminal JavaBean are available in the Host Access Toolkit. For more information about Screen Customizer, refer to the Screen Customizer website; for details of using it in conjunction with the Host Access Toolkit, see *Introduction to the Host Access Beans*.

The Host Access ActiveX Controls provide emulator functions as a set of ActiveX Controls that developers can use to rapidly develop custom host access

applications. For more information about the Host Access ActiveX Controls, refer to the *Host Access Beans for Java* Reference publication.

What's new in Version 6

Beginning with Version 5, the Host Access Toolkit is available on a separate CD-ROM, as a component of Host On-Demand; it is shipped with the Host On-Demand product. The Host Access Toolkit CD-ROM contains all of the Host Access programming API's and documentation necessary for writing Host Access applets and applications.

Additional and improved support items of note for the toolkit are:

- Run Applet Improvements
 - Current user applets must implement `ECLAppletInterface`, which provides access to an active `ECLSession` at run-time. This new facility adds `CustomInterface`, which provides user applets with access to HOD additional components:
 - `HostTerminal`
 - Applet instance
 - Frame
- Host On-Demand J2EE Connector

For more information about what's new in each of the APIs, see each reference publication:

- *Host Access Class Library* Reference
- *Host Access Beans for Java* Reference
- *Host On-Demand J2EE Connector* Reference
- *Screen Customizer Component Interface* Reference

Chapter 2. Planning for Host Access Toolkit

- Supported operating systems
- Disk space requirements
- Supported browsers
- Java 2 Platform Support

Supported operating systems

For updates to this information, refer to the README file.

The Host Access Toolkit CD-ROM provides an installation utility for the following operating systems:

- Windows[®] 95 or Windows 98
- Windows Me
- Windows NT[®] 4.0 with service pack 3 or higher
- Windows 2000

Applications or applets developed using the Host Access Toolkit can be used on other operating systems that support Java; however, installation of the toolkit is not supported on those other systems. The necessary toolkit jar files needed to run your application or applet can be packaged with your application or applet and copied to those other systems within the bounds of your licensing agreement.

Disk space requirements

These requirements are based on a typical installation and are only estimates. Sizes can vary by operating system.

- Windows 95 or 98 or Me- 45MB
- Windows NT - 45MB
- Windows 2000 - 45MB

Supported browsers

For the most up to date information, refer to the README file and to the Host On-Demand Web site.

The following browsers can be used to run Host Access Beans for Java or Host Access Class Library applets:

- Netscape Navigator 4.08 or 4.5 or higher (Windows 95, 98, Me, NT, Windows 2000) or higher
- Microsoft[®] Internet Explorer 4.01 with service pack 1, or 5.0 or higher (Windows 95, 98, Me, NT, Windows 2000) (**Note:** JVM level must be 3165 or higher.)

Java 2 Support



The Host Access Beans for Java are supported in a Java 2 environment using JDK™ 1.3.0. However, the visual components of the Host Access Beans for Java are built using AWT classes. There are known issues when building applications or applets that use both AWT components and Swing components. For more information on compatibility between AWT and Swing components, see the Sun Java Web site.

Note: You may have to download the latest Java Virtual Machine runtime environment for your server.

Chapter 3. Installing Host Access Toolkit

The basic steps for installing Host Access Toolkit on a Windows system are:

1. Insert the CD-ROM into your CD-ROM drive. If AutoPlay is on, inserting the CD-ROM will automatically invoke the CD-ROM interface program.

Note: If AutoPlay is off, do the following:

- a. Select **Run...** from either the **File** pull-down or the task bar.
- b. Enter

E:\setup

in the Command Line box (where *E:* is your CD-ROM drive).

2. If you wish, click on **View Documentation**, **Visit Our Web site**, or **Explore This CD** to learn more.
3. When you are ready, click **Install Product**.
4. Follow the directions in the installation windows.

Chapter 4. Migrating from a previous version of Host On-Demand

Previous installations of Host On-Demand do not affect installation of the Host Access Toolkit. Host Access Beans for Java and Host Access Class Library applications and applets written using a previous version of Host On-Demand will work with this version of the Host Access Toolkit.

Chapter 5. Introducing the Host Access Toolkit Application Programming Interfaces

- Introducing Host Access Class Library
- Introducing Host Access Beans

Introducing Host Access Class Library

The Host Access Class Library (HACL) for Java provides a core set of classes and methods that allow the development of platform independent applications that can access host information at the data stream level. HACL implements the core host access function in a complete class model that is independent of any graphical display and only requires a Java-enabled browser or comparable Java environment to operate. The class library represents an object-oriented abstraction of a host connection that includes reading and writing the host presentation space, enumerating the fields in the presentation space, reading the operator information area (OIA) for status information, transferring files, and performing asynchronous notification of significant events.

With HACL, application developers can write Java applets that manipulate data from the data stream presentation space (such as 3270, 5250, and Virtual Terminal) without requiring the users to have the applets residing on their machines. The presentation space represents a virtual terminal screen that contains both data and associated attributes presented by host applications. HACL Java applets can open a session to the host, wait for incoming host data, get specific strings from the imaginary screen, get associated attributes of the strings, set new string values, send data stream function keys back to the host, and wait for the next host response. After an interaction is complete, the applet can switch to other tasks or simply close the session. The entire operation can be done without ever showing host screens.

HACL is a significant improvement over traditional emulator programming interfaces, like EHLLAPI, in several ways. These improvements are:

- EHLLAPI requires significant preparation before and interpretation after function calls.
- The one EHLLAPI entry point function is confusing and without context.
- EHLLAPI has unintuitive mnemonic key values. For example, in EHLLAPI the mnemonic for Enter is '@E' and in HACL it is '[enter]'.
- HACL is an object oriented API, with all the well known benefits of the object oriented programming paradigm.



Host On-Demand's HACL implementation provides the following additional benefits:

- It is 100% Java.
- It is platform independent.
- It is downloadable and executable on client workstations using standard Web and Java technology. This provides significant maintenance and resource savings.

Finally, applications or applets written for HACL are portable between Host On-Demand and Personal Communications.

Note

The API books describe functionality for both the Host On-Demand and Personal Communications environments. Functions specific to Personal Communications are flagged. However, these books are not meant to serve as the documentation for the Personal Communications support. Please refer to the version of the books supplied with your version of Personal Communications to determine the specific functions supported in that environment.

Introducing Host Access Beans

The Host Access Beans for Java provide emulator functions as a set of JavaBeans. JavaBeans are components that have configurable properties, use events to communicate, and can be manipulated in visual development environments. (See the Sun JavaBeans API Specification for more information). The Host Access Beans can be used by developers to rapidly develop custom applications that deliver the specific functions they want to include in their host access applications.

Descriptions

Each bean description below is preceded by the icon used to represent the bean in most visual development environments.



Session - This non-visual bean provides methods and properties for setting up and establishing communications with the host system. The Session bean fires events that allow listeners to be notified of presentation space, operator information area (OIA), and communication changes.



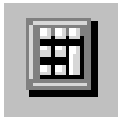
Screen - This visual bean provides the graphical interface for displaying the host data from a Session bean. The Screen bean listens to presentation space, OIA, and GUI events fired by the Session bean and interprets the events to display the main text area and the operator information area. It fires keystroke events to registered listeners, and also provides the clipboard cut, copy, and paste functions.



Terminal - This visual bean combines the Session and Screen beans to provide a composite bean that encompasses both the communication with the host and the graphical interface for displaying the host data.



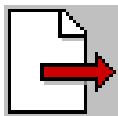
CustomTerminal - This visual bean is installed with the Screen Customizer product and provides a customizable graphical interface for displaying host data. Only KeyPad, KeyRemap and FileTransfer beans can be "wired" to this bean. The CustomTerminal bean is not supported as an ActiveX control.



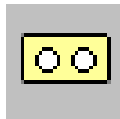
KeyPad - This visual bean provides a simple grid of buttons which invoke various host functions.



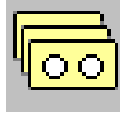
KeyRemap - This visual bean provides keyboard remap capability. Using KeyRemap, keystrokes can be mapped to alternate characters or directly to host functions.



FileTransfer - This visual bean provides a toolbar interface for transferring files to and from a host.



Macro - This non-visual bean records and plays a single macro. Macro employs advanced screen recognition technology to reliably navigate host applications in any environment. Macro also provides the ability to prompt for user input and extract text from the screen during playback.



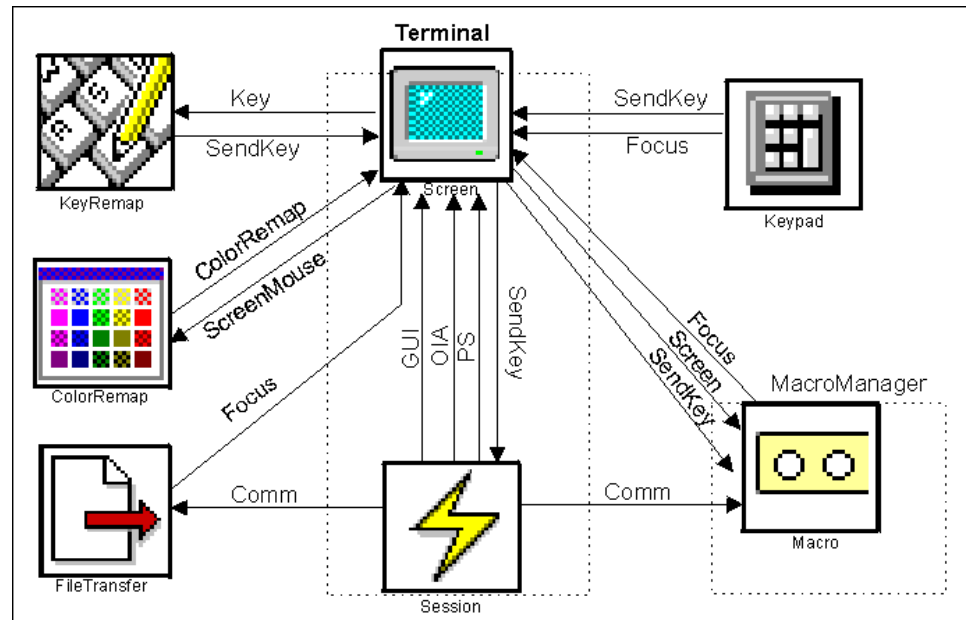
MacroManager - This visual bean provides a toolbar interface for managing multiple macros. The MacroManager bean allows you to record, play, load, delete and edit multiple macros.



Color Remap - This visual bean provides a simple interface for modifying the colors displayed by the Screen or Terminal beans.

Events

The following diagram shows the major events that are passed between each of the Host Access Beans and the composition of the Terminal bean.



The CustomTerminal bean is wired similarly to the Terminal bean; however, only Keypad, KeyRemap, and FileTransfer beans can be wired to the CustomTerminal bean.

Introducing Host On-Demand J2EE Connector

Before the J2EE Connector architecture was introduced, there was no standard architecture for integrating heterogeneous Enterprise Information Systems (EIS). Host On-Demand users had to use Host Access Class Library (HACL) to access hosts; other vendors provide specific architectures for this purpose. J2EE Connector architecture provides a standard set of services allowing developers to quickly connect and integrate their applications with virtually any backend EIS, and to any application servers conforming to the J2EE Connector architecture.

The Host On-Demand J2EE Connector provides access to 3270, 5250, Customer Information and Control System (CICS**), and Virtual Terminal (VT) hosts from the Internet. The Host On-Demand J2EE Connector is a Java programming interface which conforms to the J2EE Connector Specification version 1.0 - Proposed Final Draft 2 from Sun Microsystems, and the Common Connector Framework (CCF) infrastructure required by VisualAge for Java - Enterprise Access Builder and Component Broker. This interface is intended for use with these products to develop applications in a consistent manner with other connectors. This is one of the programming environments Host On-Demand customers may use. In other environments, where the customer needs the complete functionality of Host On-Demand, the HACL and JavaBeans interface should be used.

Chapter 6. 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 (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation

Licensing

2-31 Roppongi 3-chome, Minato-ku

Tokyo 106, 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.

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.

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

Department T01

Building B062

P.O. Box 12195

Research Triangle Park, NC 27709-2195

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 information 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.

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.

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

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

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

- IBM
- VisualAge

Lotus[®] and Domino[™] are trademarks or registered trademarks of Lotus Development Corporation in the United States, or other countries, or both.

Java, JavaBeans JDK and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

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



Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.