Platform SDK 7.6

Deployment Guide

About Genesys

Genesys Telecommunications Laboratories, Inc., a subsidiary of Alcatel-Lucent, is 100% focused on software for call centers. Genesys recognizes that better interactions drive better business and build company reputations. Customer service solutions from Genesys deliver on this promise for Global 2000 enterprises, government organizations, and telecommunications service providers across 80 countries, directing more than 100 million customer interactions every day. Sophisticated routing and reporting across voice, e-mail, and Web channels ensure that customers are quickly connected to the best available resource—the first time. Genesys offers solutions for customer service, help desks, order desks, collections, outbound telesales and service, and workforce management. Visit www.genesyslab.com for more information.

Each product has its own documentation for online viewing at the Genesys Technical Support website or on the Documentation Library DVD, which is available from Genesys upon request. For more information, contact your sales representative.

Notice

Although reasonable effort is made to ensure that the information in this document is complete and accurate at the time of release, Genesys Telecommunications Laboratories, Inc., cannot assume responsibility for any existing errors. Changes and/or corrections to the information contained in this document may be incorporated in future versions.

Your Responsibility for Your System's Security

You are responsible for the security of your system. Product administration to prevent unauthorized use is your responsibility. Your system administrator should read all documents provided with this product to fully understand the features available that reduce your risk of incurring charges for unlicensed use of Genesys products.

Trademarks

Genesys, the Genesys logo, and T-Server are registered trademarks of Genesys Telecommunications Laboratories, Inc. All other trademarks and trade names referred to in this document are the property of other companies. The Crystal monospace font is used by permission of Software Renovation Corporation, www.SoftwareRenovation.com.

Technical Support from VARs

If you have purchased support from a value-added reseller (VAR), please contact the VAR for technical support.

Technical Support from Genesys

If you have purchased support directly from Genesys, please contact Genesys Technical Support at the following regional numbers:

Region	Telephone	E-Mail
North and Latin America	+888-369-5555 or +506-674-6767	support@genesyslab.com
Europe, Middle East, and Africa	+44-(0)-1276-45-7002	support@genesyslab.co.uk
Asia Pacific	+61-7-3368-6868	support@genesyslab.com.au
Japan	+81-3-6361-8950	support@genesyslab.co.jp

Prior to contacting technical support, please refer to the *Genesys Technical Support Guide* for complete contact information and procedures.

Ordering and Licensing Information

Complete information on ordering and licensing Genesys products can be found in the ***Genesys 7 Licensing Guide*.**

Released by

Genesys Telecommunications Laboratories, Inc. www.genesyslab.com

Document Version: 76sdk_dep_platform_12-2008_v7.6.201.00

Table of Contents

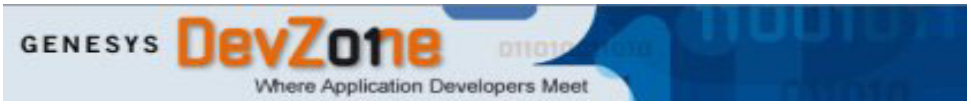
List of Procedures

Preface

Welcome to the *Platform SDK 7.6 Deployment Guide*. This guide introduces you to the architecture, required components, and procedures relevant to the deployment of the various Platform SDKs in your contact center.

This document is valid only for the 7.6 release of this product.

Note: For versions of this document created for other releases of this product, please visit the Genesys Developer website, the DevZone (http://www.genesyslab.com/developer), the Genesys Technical Support website, or request the Developer Documentation Library DVD, which you can order by e-mail from Genesys Order Management at orderman@genesyslab.com.



This preface provides an overview of this document, identifies the primary audience, introduces document conventions, and lists related reference information:

In brief, you will find the following information in this guide:

- Prerequisites for deploying Platform SDKs.
- How to install the Platform SDKs.
- How to configure related portions of your Genesys environment to accommodate the Platform SDKs.

Intended Audience

This guide is primarily intended for application developers who are familiar with Java or .NET technologies and who are planning to develop custom applications for the Genesys Framework environment.

The purpose of this document is to identify how to deploy your Platform SDKs and to highlight some important conceptual issues.

It assumes that you have a basic understanding of:

- The underlying concepts and terminology for the type of application you plan to develop. For instance, an understanding of CTI technology is important for developing an application with Voice Platform SDK.
- Network design and operation.
- Your own network configurations.

You should also be familiar with messaging-compliant programming and Java- and .NET-related development tools, as well as how client and server applications work.

Usage Guidelines

The Genesys developer materials outlined in this document are intended to be used for the following purposes:

- Creation of contact-center agent desktop applications associated with Genesys software implementations.
- Server-side integration between Genesys software and third-party software.
- Creation of a specialized client application specific to customer needs.

The Genesys software functions available for development are clearly documented. No undocumented functionality is to be utilized without Genesys's express written consent.

The following Use Conditions apply in all cases for developers employing the Genesys developer materials outlined in this document:

1. Possession of interface documentation does not imply a right to use by a third party. Genesys conditions for use, as outlined below or in the *Genesys Developer Program Guide,* must be met.
2. This interface shall not be used unless the developer is a member in good standing of the Genesys Interacts program or has a valid Master Software License and Services Agreement with Genesys.
3. A developer shall not be entitled to use any licenses granted hereunder unless the developer's organization has met or obtained all prerequisite licensing and software as set out by Genesys.

4. A developer shall not be entitled to use any licenses granted hereunder if the developer's organization is delinquent in any payments or amounts owed to Genesys.
5. A developer shall not use the Genesys developer materials outlined in this document for any general application development purposes that are not associated with the above-mentioned intended purposes for the use of the Genesys developer materials outlined in this document.
6. A developer shall disclose the developer materials outlined in this document only to those employees who have a direct need to create, debug, and/or test one or more participant-specific objects and/or software files that access, communicate, or interoperate with the Genesys API.
7. The developed works and Genesys software running in conjunction with one another (hereinafter referred to together as the "integrated solutions") should not compromise data integrity. For example, if both the Genesys software and the integrated solutions can modify the same data, then modifications by either product must not circumvent the other product's data integrity rules. In addition, the integration should not cause duplicate copies of data to exist in both participant and Genesys databases, unless it can be assured that data modifications propagate all copies within the time required by typical users.
8. The integrated solutions shall not compromise data or application security, access, or visibility restrictions that are enforced by either the Genesys software or the developed works.
9. The integrated solutions shall conform to design and implementation guidelines and restrictions described in the *Genesys Developer Program Guide* and Genesys software documentation. For example:
    a. The integration must use only published interfaces to access Genesys data.
    b. The integration shall not modify data in Genesys database tables directly using SQL.
    c. The integration shall not introduce database triggers or stored procedures that operate on Genesys database tables.

Any schema extension to Genesys database tables must be carried out using Genesys Developer software through documented methods and features.

The Genesys developer materials outlined in this document are not intended to be used for the creation of any product with functionality comparable to any Genesys products, including products similar or substantially similar to Genesys's current general-availability, beta, and announced products.

Any attempt to use the Genesys developer materials outlined in this document or any Genesys Developer software contrary to this clause shall be deemed a material breach with immediate termination of this addendum, and Genesys shall be entitled to seek to protect its interests, including but not limited to, preliminary and permanent injunctive relief, as well as money damages.

Chapter Summaries

In addition to this opening chapter, this guide contains these chapters:

- Chapter 1, "Overview," on page 15, summarizes the architecture and general concepts related to the Platform SDKs.
- Chapter 2, "Installation and Configuration," on page 27, offers details about the installation procedures for the various Platform SDKs, and about the changes you need to make in the Configuration Layer to accommodate your application.
- Chapter 3, "Starting and Testing Your Applications," on page 37, briefly outlines some issues you may need to consider to get your applications up and running.

Document Conventions

This document uses certain stylistic and typographical conventions—introduced here—that serve as shorthands for particular kinds of information.

Document Version Number

A version number appears at the bottom of the inside front cover of this document. Version numbers change as new information is added to this document. Here is a sample version number:

76sdk_dep_platform_12-2008_v7.6.201.00

You will need this number when you are talking with Genesys Technical Support about this product.

Type Styles

Italic

In this document, italic is used for emphasis, for documents' titles, for definitions of (or first references to) unfamiliar terms, and for mathematical variables.

Examples:

- Please consult the *Genesys 7 Migration Guide* for more information.
- *A customary and usual practice* is one that is widely accepted and used within a particular industry or profession.
- Do *not* use this value for this option.
- The formula, $x + 1 = 7$ where x stands for . . .

Monospace Font

A monospace font, which looks like `teletype or typewriter text`, is used for all programming identifiers and GUI elements.

This convention includes the *names* of directories, files, folders, configuration objects, paths, scripts, dialog boxes, options, fields, text and list boxes, operational modes, all buttons (including radio buttons), check boxes, commands, tabs, CTI events, and error messages; the values of options; logical arguments and command syntax; and code samples.

Examples:

- Select the `Show variables on screen` check box.
- Click the `Summation` button.
- In the `Properties` dialog box, enter the value for the host server in your environment.
- In the `Operand` text box, enter your formula.
- Click `OK` to exit the `Properties` dialog box.
- The following table presents the complete set of error messages T-Server® distributes in `EventError` events.
- If you select `true` for the `inbound-bsns-calls` option, all established inbound calls on a local agent are considered business calls.

Monospace is also used for any text that users must manually enter during a configuration or installation procedure, or on a command line:

Example:

- Enter `exit` on the command line.

Screen Captures Used in This Document

Screen captures from the product GUI (graphical user interface), as used in this document, may sometimes contain a minor spelling, capitalization, or grammatical error. The text accompanying and explaining the screen captures corrects such errors *except* when such a correction would prevent you from installing, configuring, or successfully using the product. For example, if the name of an option contains a usage error, the name would be presented exactly as it appears in the product GUI; the error would not be corrected in any accompanying text.

Square Brackets

Square brackets indicate that a particular parameter or value is optional within a logical argument, a command, or some programming syntax. That is, the parameter's or value's presence is not required to resolve the argument, command, or block of code. The user decides whether to include this optional information. Here is a sample:

```
smcp_server -host [/flags]
```

Angle Brackets

Angle brackets indicate a placeholder for a value that the user must specify. This might be a DN or port number specific to your enterprise. Here is a sample:

```
smcp_server -host <confighost>
```

Related Resources

Consult these additional resources as necessary:

Platform SDK Documentation

- *Platform SDK 7.6 Developer's Guide*, which provides detailed information on how to develop applications of all types using your Platform SDK.
- *Platform SDK 7.6 API Reference* for the particular SDK you are using, which provides the authoritative information on methods, functions, and events for your SDK.
- *Platform SDK 7.6 Application Block Guides* for the particular application block you are using. Each *Guide* explains how to use the application block and documents all code used in the application block itself. (Application blocks are production-quality available code.)
- *Platform SDK 7.6 Code Examples* for the particular SDK you are using, which offer illustrative ways to begin using your SDK. These code examples are fully functioning software applications, but are for educational purposes only and are not supported.
- The Release Notes and Product Advisories for this product, which are available on the Genesys Technical Support website at http://genesyslab.com/support.

Supporting Documentation

- The *Deployment Guides* for the underlying Genesys servers with which you intend to have your Platform SDK applications integrate. For instance, check the *Framework 7.6 SIP Server Deployment Guide* if you plan on using the Voice Platform SDK and the SIP Endpoint Application Block.

  Note: The *SIP Server Deployment Guide* also has details on the T-Server and DN options you need to set to allow the SIP Endpoint Application Block to work with Genesys SIP Server.
- Other Genesys SDK documentation for extended information on ways to integrate custom applications with Genesys Servers. This includes documents such as the *Genesys 7 Events and Models Reference Manual,* which contains detailed information on the TLIB protocol and on message exchanges with T-Servers.

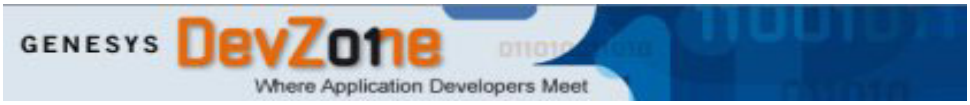
  Note: The *Voice Platform SDK API Reference* documents also contain a reference of all TLIB events, data types, and functions calls.

- The *Genesys 7 Migration Guide*, also on the Genesys Documentation Library DVD, which provides a documented migration strategy from Genesys product releases 5.1 and later to all Genesys 7.x releases. Contact Genesys Technical Support for additional information.
- The *Genesys Technical Publications Glossary,* which ships on the Genesys Documentation Library DVD and which provides a comprehensive list of the Genesys and CTI terminology and acronyms used in this document.
- Information on supported hardware and third-party software is available in the following documents:
  - *Genesys Supported Operating Systems and Databases*
  - *Genesys Supported Media Interfaces*

Locating Genesys Documentation

Genesys product documentation is available from the:

- Genesys Technical Support website at http://genesyslab.com/support.
- Genesys Documentation Library DVD, which you can order by e-mail from Genesys Order Management at orderman@genesyslab.com.
- DevZone website at http://www.genesyslab.com/developer.



Making Comments on This Document

If you especially like or dislike anything about this document, please feel free to e-mail your comments to Techpubs.webadmin@genesyslab.com.

You can comment on what you regard as specific errors or omissions, and on the accuracy, organization, subject matter, or completeness of this document. Please limit your comments to the information in this document only and to the way in which the information is presented. Speak to Genesys Technical Support if you have suggestions about the product itself.

When you send us comments, you grant Genesys a nonexclusive right to use or distribute your comments in any way it believes appropriate, without incurring any obligation to you.

Chapter

1 Overview

This chapter summarizes the capabilities of the Platform SDKs and includes the following sections:

Introduction

The Platform SDKs allow you to write .NET and Java applications that communicate with Genesys servers in their native protocols. You can think of the Platform SDKs as "Server SDKs," since each one unlocks the capabilities of the server it connects to. In contrast to the abstraction of the Interaction SDKs, the Platform SDKs offer low-level components and fine-grained, message-driven interfaces which are also XML friendly.

The Platform SDKs connect to the following Genesys servers:

- Configuration Platform SDK—Configuration Server
- Contacts Platform SDK—Universal Contact Server
- Management Platform SDK—Solution Control Server (SCS), Message Server, and Local Control Agent (LCA)
- Open Media Platform SDK—Interaction Server
- Outbound Contact Platform SDK—Outbound Contact Server
- Statistics Platform SDK—Stat Server

- Voice Platform SDK—T-Server
- Web Media Platform SDK—E-mail Server Java, Chat Server, and Callback Server

Development Aids

To assist with development, each Platform SDK comes packaged with its own API reference. There are also a number of production-quality *application blocks* available for integration into your code, as well as sample applications that illustrate the basic usage of the SDKs. Genesys recommends that you use the application blocks to handle basic functions such as connecting to a server and handling events. See "Application Blocks" on page 19 and "Sample Applications" on page 22 for more details.

Product Release Notes

For every Genesys product there is a Release Notes file that provides any late-breaking product information that could not be included in the manual. This information can often be important. Open the `Read_Me.html` file in the application home directory for a link to the latest Release Notes.

Platform SDK Technologies

Each Platform SDK allows you to develop applications using more than one technology. These technologies are as follows for the 7.6 release:

- .NET
- Java
- XML

The Platform SDKs

This section briefly outlines the capabilities of the Platform SDKs. Starting with release 7.6.2, each of the SDKs listed below is available for .NET and Java development platforms. For detailed information on all of the Platform SDKs, see the *Platform SDK 7.6 Developer's Guide.*

Configuration Platform SDK

The Configuration Platform SDK allows you to build applications that add, modify, and delete information in the Configuration Layer of your Genesys environment. This SDK accesses information directly from Configuration

Server, allowing you to design applications that view Configuration Layer data and then modify it on the fly.

See the *Configuration Platform SDK 7.6 API Reference* for .NET or Java to find detailed interface descriptions. The Configuration Platform SDK also includes the Configuration Object Model Application Block. See "Application Blocks" on page 19 for details.

Note: Genesys recommends that you use the Configuration Object Model Application Block when you are working with Configuration Layer objects.

Contacts Platform SDK

The Contacts Platform SDK allows you to build applications that view, or interact with, the contact information for your contact center. This SDK accesses information directly from Universal Contact Server, allowing you to design applications that access contact information when dealing with multimedia interactions such as chat or e-mail, for example.

See the *Contacts Platform SDK 7.6 API Reference* for .NET or Java to find detailed interface descriptions.

Management Platform SDK

The Management Platform SDK allows you to design applications that monitor and start/stop applications in your Genesys environment. Because it integrates with the core Management Layer Servers (Message Server and Solution Control Server) and LCA, it gives you direct access to the status and logs of any component that your Genesys software is already tracking in your system.

See the *Management Platform SDK 7.6 API Reference* for .NET or Java to find detailed interface descriptions.

Open Media Platform SDK

The Open Media Platform SDK allows you to design applications that might do any number of things:

- At an agent desktop, handle interactions of open media types.
- From a server-side perspective, introduce open media interactions into your Genesys environment from external sources.
- Perform external service processing (ESP) on interactions that are already in the Genesys environment.

- The interaction submission and ESP capabilities of the Open Media Platform SDK allow you to build custom media servers that become part of your Genesys Multimedia environment. This enables you to use Interaction Routing Designer's specialized objects to route interactions to and from your custom media server.

See the *Open Media Platform SDK 7.6 API Reference* for .NET or Java to find detailed interface descriptions.

Outbound Contact Platform SDK

The Outbound Contact Platform SDK allows you to design applications that manage outbound campaigns through direct integration with the Outbound Contact Server. You can start and stop outbound campaigns that are already defined in the Configuration Layer. For instance, you can set a given campaign to begin based on some programmed time or date or on some observed threshold reached in your contact center.

See the *Outbound Contact Platform SDK 7.6 API Reference* for .NET or Java to find detailed interface descriptions.

Statistics Platform SDK

The Statistics Platform SDK allows you to design applications that solicit and monitor statistics from a Genesys environment. This SDK accesses information provided by the Stat Server. Your design can tap into statistics in an ad hoc manner, or subscribe to particular statistics and have them updated automatically.

See the *Statistics Platform SDK 7.6 API Reference* for .NET or Java to find detailed interface descriptions.

Voice Platform SDK

The Voice Platform SDK allows you to design applications that monitor and control voice interactions in a Genesys environment. This SDK is designed to work solely with T-Servers, including those in premise or network roles, and using traditional voice or IP voice technologies. Use this SDK to build either server-side or agent-desktop applications.

See the *Voice Platform SDK 7.6 API Reference* for .NET or Java to find detailed interface descriptions. Users of the Voice Platform SDK may find the related SIP Endpoint Application Block of interest. See "Application Blocks" on page 19.

Web Media Platform SDK

The Web Media Platform SDK Java and .NET allows you to design applications that integrate with Genesys non-voice media servers. This allows you to introduce into a Genesys environment e-mail, chat and callback request interactions that begin in a web environment. The Web Media Platform SDK thereby gives you an opportunity to fully customize the experiences your customers have when trying to contact your company over the Internet.

See the *Web Media Platform SDK 7.6 API Reference* for .NET or Java to find detailed interface descriptions.

Application Blocks

This section gives a list of the Platform SDK application blocks that currently ship with the product, and describes some of the basic requirements for using these components.

What are Application Blocks?

When you are working with a message-based API, you need to handle events. When you are using an application that needs to communicate with more than one server, you have to manage the connections to each server and keep track of the interactions with each one.

These basic functions are common to almost all client applications, so why should every development team have to write new code to address functionality that others have already had to deal with?

Genesys provides reusable production-quality components that carry out these functions and other common development tasks facing Platform SDK developers. We call these components *application blocks*. They have been designed using industry best practices so you can use them without modification. We have also included the source code so you can tailor them if you need to.

In the examples mentioned above, you should simply use the Message Broker Application Block to easily handle message-based events that are returned from Genesys servers, or the Protocol Manager Application Block to manage connections to one or more servers.

How Can I Use Application Blocks?

Genesys intends for all developers to use these components when building custom applications, so that the development process becomes faster and more efficient. So the question becomes: what steps are required to include the application blocks in your code?

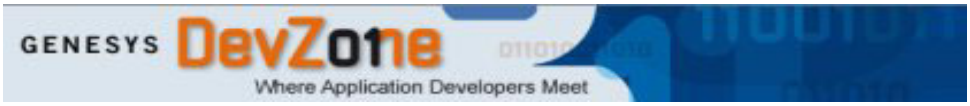
To encourage you to use the application blocks, the process has been streamlined as much as possible. Simply install Platform SDK on your system, locate the application blocks inside the installation folder, and compile the components you will use into a JAR (for Java development) or DLL (for .NET development) that will be referenced in your code. Most application blocks

even include a quick start application to compile the component for you, making the process even easier.

Note: You must have .NET Framework 2.0 installed to use Genesys application blocks with Platform SDK for .NET 7.6.

For examples of exactly how these application blocks are used, check the code examples included with your Platform SDKs before you begin building your own application. See "Sample Applications" on page 22 for additional details.

To download the most recent versions of Genesys application blocks, or to find more developer information, visit the Genesys Developer Zone forums at `http://www.genesyslab.com/developer`.



For more information about using the quick start applications, or about the application blocks in general, refer to your *Platform SDK 7.6 Developer's Guide*.

Application Blocks Shipping With Platform SDK 7.6

Note: Unless otherwise noted, all application blocks released with Platform SDK 7.6 are available for Java and .NET development.

Unless specified otherwise, all application blocks for the 7.6 release are available for both the Java and .NET platform, and provide the following functionality:

Configuration Object Model

Use the Configuration Object Model (COM) Application Block when you need to:

- Create, modify, or delete Configuration Layer objects.
- Update the permissions for Configuration Layer objects.

The COM Application Block allows you to work with an efficient object-based model for configuration objects, providing a layer of abstraction over the XML documents returned by the Configuration SDK.

For example, many contact center applications need to retrieve information about the properties of `Application` objects in the Genesys Configuration Layer. Using the COM Application Block allows you to work with a convenient instance of the `CfgApplication` class instead of having to write code to parse and interpret XML responses that describe the `Application` objects.

For more information on the COM Application Block, see the *Application Block Guide* included with this application block.

Message Broker

Use the Message Broker Application Block to handle events in an efficient way. Almost all applications will need to deal with events.

This application block allows you to set up individual classes to handle specific events coming from Genesys servers. It receives all of the events from the servers you specify, and sends each one to the appropriate handler class. Message Broker Application Block is a high-performance way to hide the complexity of event-driven programming—so you can focus on other areas of your application.

For more information on the Message Broker Application Block, see the *Application Block Guide* included with this application block. To see this application block in use, refer to the code examples included with your Platform SDKs.

Protocol Manager

Use the Protocol Manager Application Block whenever you plan to use more than one Platform SDKs in a custom application. This application block uses a service-based API to manage Platform SDK connections to Genesys servers, and also includes built-in warm standby capabilities.

For instance, you might want a voice-based agent desktop that also presents some Configuration Layer access, such as the ability for an agent to change his or her password. The Protocol Manager Application Block allows you to easily open, access, and close these connections in a centralized way within your application.

For more information on the Protocol Manager Application Block, see the *Application Block Guide* included with this application block. To see this application block in use, refer to the code examples included with your Platform SDKs.

SIP Endpoint (for .NET only)

Use the SIP Endpoint Application Block to build a SIP endpoint that can seamlessly connect agent desktop applications with the Genesys SIP Server in order to handle audio or video calls.

The SIP Endpoint Application Block is designed to be integrated into an agent desktop so the agent can use a single user interface to control calls. Genesys recommends that this be done in a way that leaves actual control in the hands of a T-Lib–based agent desktop application, which has a fuller feature set and is also fully supported by Genesys.

The SIP Endpoint Application Block is also designed to integrate with the Genesys SIP Server. It supports the SIP, SDP, and RTP/RTCP protocols.

This application block should be deployed with applications you develop using the Voice Platform SDK. It is an out-of-the-box SIP voice application for use with Genesys SIP Server. For your convenience, this application block also

includes a quick-start application so that you can run it in stand-alone mode if desired.

For more information on the SIP Endpoint Application Block, see the *Application Block Guide* included with this application block.

Warm Standby

The Warm Standby Application Block provides code that enables warm-standby high availability for applications you develop with any of the Platform SDKs.

Many contact center environments require redundant backup servers that are able to take over quickly if a primary server fails. In this situation, the primary server operates in active mode, accepting connections and exchanging messages with clients. The backup server, on the other hand, is in standby mode. If the primary server fails, the backup server switches to active mode, assuming the role and behavior of the primary server.

For more information on the Warm Standby Application Block, see the *Application Block Guide* included with this application block.

Note: If you have comments or suggestions about the application blocks, then contact us through the Genesys Developer Zone forums located at http://www.genesyslab.com/developer.

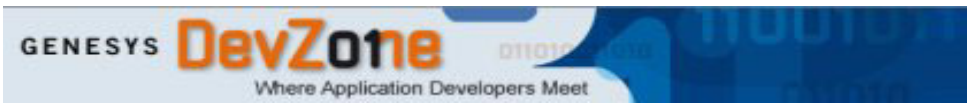
Sample Applications

Genesys understands how helpful working code can be in learning to use a new API. Because of that, the Platform SDKs include several sources of code snippets, including the Platform SDK Developer's Guide, the API references, and the application blocks.

The Platform SDKs also include sample applications—compiled, ready-to-run code examples that typically illustrate common use cases to help you get up and running during your application development. All code examples are for educational purposes only.

Note: Although you can use the sample applications provided with the Platform SDKs in any way that you like, they are neither tested nor supported by Genesys in any way.

To download the most recent versions of Genesys sample applications, or to find more developer information, visit the Genesys Developer Zone forums at http://www.genesyslab.com/developer.



A general overview of the code examples follows. For a detailed analysis of each code example, and for instructions on setting up your development environment to use these samples, refer to your *Platform SDK 7.6 Developer's Guide*.

Code Examples Shipping With Platform SDK 7.6

Unless specified otherwise, all of the following samples are provided in both C# .NET and Java:

Configuration Example

This sample application allows you to retrieve data from Configuration Server about objects of a specific type. This example uses the Configuration Platform SDK together with the Message Broker and Protocol Manager Application Blocks, demonstrating how easily application blocks can be integrated into your custom code.

Complex Example (for .NET only)

This example shows how to work with multiple Genesys servers in a single application, using the services of the Protocol Manager and Message Broker Application Blocks. It makes use of concepts you have learned from other code examples to connect with both Configuration Server and Stat Server, and teaches how to handle incoming events from multiple servers correctly.

Open Media Examples

Two separate applications that demonstrate two main features of this SDK. The first part is a server-side application that submits open media type interactions (workitems) to Interaction Server for processing. The second part simulates an agent desktop application, allowing you to view and retrieve open media interactions, and then mark them `Done` after they are processed.

Statistics Example

This simple example subscribes to a statistic, and then delivers the resulting information and updates to the desktop. Both the Protocol Manager and Message Broker Application Blocks are used in this example, demonstrating how easily application blocks can be integrated into your custom code.

Voice Example (for .NET only)

This example creates a basic agent desktop application that allows you to log an agent in and out, and to handle incoming voice interactions (answer, hold, retrieve, transfer) or initiate outgoing voice interactions.

Note: The Open Media and Voice code example do not use the Protocol Manager or Message Broker Application Blocks. However, Genesys recommends that you use these application blocks when writing production code.

Environment Prerequisites

Note the following prerequisites for your Platform SDK environment:

- For all Platform SDKs, Genesys only supports the use of release 7.2 or higher of the underlying servers to which your custom applications connect.
- For .NET implementations, ensure that .NET Framework 2.0 is installed on the computer where you plan to run your Platform SDK application. The .NET Framework can be downloaded free of charge from Microsoft. Check the *Genesys Supported Operating Systems and Databases* for details.
- If you plan on using the SIP Endpoint Application Block, be sure that RTC release 1.3, is installed on the computer where your applications run. You can download the RTC API SDK from Microsoft:

  `http://www.microsoft.com/downloads/details.aspx?FamilyID=c3a7bd15-fd1c-4bf7-a505-3f8faf1e120a&DisplayLang=en`

  Once you install the RTC API SDK, run `RtcApiSetup.msi` (available in the subdirectory: `\RTC Client API v1.3 SDK\Installation\`). Running this file installs the actual RTC Client API, version 1.3, and allows the side-by-side RTC Client API version 1.3 DLLs to be used on that host. Refer to Microsoft's RTC Client API SDK documentation for more information about this installation, repackaging, and redistribution of the RTC Client API:

  `http://msdn.microsoft.com/library/default.asp?url=/library/en-us/rtcclnt/rtc/real_time_communications_rtc_client_start_page.asp?frame=true`

  and

  `http://msdn.microsoft.com/library/default.asp?url=/library/en-us/rtcclnt/rtc/how_to_install_an_rtc_client_application.asp?frame=true.`

  Note: The SIP Endpoint Application Block is an architectural guide for developers who want to add IP telephony capabilities to their softphone applications. It is provided as source code that can either be used as is, or can be modified and extended by developers.

  The Microsoft RTC stack is one of many codecs that can be used with this application block, and is discussed here for guidance only.

- To ensure that conferences work, check that your multipoint control unit (MCU)—for instance, Genesys Stream Manager—is installed and configured correctly. See your *Framework 7.6 SIP Server Deployment Guide* for details.
- For Java implementations, refer to the *Genesys Supported Operating Systems and Databases* for the version of JDK you need. Note that you may need a different version of the JDK if you plan to use the application blocks.
- Configuration Platform SDK for Java, and all application blocks or code samples that make use of this SDK, require `jwsdp` 1.6 (Java Web Services Developer's Pack). The `jwsdp` 1.6 is available with your installation of the Platform SDK in the `thirdparty` folder.

Use of Configuration Platform SDK & AES Cryptography

Standard JRE and JDK installations do not accommodate AES cryptography, with a 128-bit encryption key (considered too strong for some countries and thus subject to export restrictions). However, Configuration Platform SDK uses this level of encryption.

To upgrade your Java environment accordingly, download the required software from, for example, one of the following sites:

- `http://javashoplm.sun.com/ECom/docs/Welcome.jsp?StoreId=22&PartDetailId=7503-jce-1.4.2-oth-JPR&SiteId=JSC&TransactionId=noreg`
- `https://www6.software.ibm.com/dl/jcesdk/jcesdk-p`

Deployment Task Summary

Table 1 summarizes the work flow used to deploy the Platform SDKs on your system. For the detailed procedures that make up this work flow, see Chapter 2 on page 27 and Chapter 3 on page 37.

Table 1: Platform SDK Deployment Work Flow

Objective	Related Procedures and Actions
Install and verify the Platform SDKs on your system.	1. Installing the Platform SDKs, page 28. 2. Verifying Installed Components, page 30.

Table 1: Platform SDK Deployment Work Flow (Continued)

Objective	Related Procedures and Actions
Configure SIP Server (and related) Application objects in Configuration Manager to use the SIP Endpoint Application Block.	Overview: Configuring SIP Endpoint Application Block Overview, page 33 Supporting procedures: 1. Configuring SIP Server Options, page 34 2. Configuring SIP DN Options, page 34
Build the `.dll` or `.jar` reference files for your application block (if making any customization to the included source code).	1. Building the .NET Application Blocks, page 38 2. Building the Java Application Blocks, page 39
Use the QuickStart applications to test your application blocks.	1. Starting the .NET QuickStart Applications, page 41 2. Starting the Java QuickStart Application, page 41

Chapter

2 Installation and Configuration

This chapter describes the process of installing the Platform SDKs on Windows and UNIX operating systems. Although the Platform SDKs come in two “flavors” (.NET and Java), the installation procedures are essentially the same.

This chapter contains the following sections:

Prepare for the Platform SDK Installation

The Platform SDK Product CD

Your Platform SDK product CD contains both Java and .NET installation packages and a full collection of related documentation to help with your application engineering.

Java Installation Considerations for UNIX

Take the following into consideration for a Java deployment on a UNIX operating system:

- If you choose an installation directory that already exists, and which has files in it, you cannot opt for an alternative directory without terminating the installation process.
- To terminate your installation process, avoid the use of `Ctr+C`. Instead, use the character defined by your site administrator to send an interrupt signal.

- If you decide to use a shell script to perform your installation, you may want your script to include the following logic:

  If the selected directory already has files in it, suspend the installation and then launch a new shell job to examine the unexpected directory.
  - If that directory cannot be reused, terminate the installation.
  - If that directory can be reused, continue with the original installation.

About Licensing

Before working with any Genesys components, note that you may require licenses. For information about products that require licenses and the License Manager installation procedure, refer to the *Genesys 7 Licensing Guide*.

Install the Platform SDKs

This section provides detailed procedures that describe how to install and configure the Platform SDKs on your system.

TIP If you attempt this procedure when an instance of Platform SDK is already installed on your system, you will be prompted to remove Platform SDK in Step 2 on page 29 instead of seeing a `Welcome` dialog.

Procedure: Installing the Platform SDKs

Purpose: To install the Platform SDK and all related files on your computer.

Prerequisites

- Check the list of "Environment Prerequisites" on page 24, and confirm that these requirements are met prior to installing the Platform SDKs.

Start of procedure

1. Run the correct installation program on your product CD, according to your development need:
   - For .NET, the installation file is named `setup.exe` and located in the `\PlatformSDK\DotNet\windows\` directory.
   - For Java, the installation file is named either `setup.exe` for Windows or `install.sh` for Unix, and is located in the `\PlatformSDK\Java\<Platform OS Name>\` directory.

   The Genesys Installation Wizard appears to guide you through the installation and setup process.

2. Click `Next` at the `Welcome` dialog.

   The `Genesys License Agreement` dialog appears, allowing you to read and confirm the license agreement before continuing.

3. Click the checkbox to accept the terms and conditions described, and then click `Next` to continue with the installation.

   The `Choose Destination Location` dialog appears, showing the default destination directory. For Window, the default directory is: `C:\Program Files\GCTI\Platform SDK for <.NET/Java> 7.6`.

4. Click `Next` to accept the default destination folder.

   If you prefer to specify a different directory, complete the following steps:

   a. Click `Browse` to open the `Choose Folder` dialog.

   b. Navigate to and select a directory path.

   c. Click `OK` to return to the `Choose Destination Location` dialog.

   d. Click `Next` to accept the destination folder that you have selected.

5. At the `Ready to Install` dialog, click `Install`.

   The Wizard installs the Platform SDK, and all associated files, in the directory you selected. When the installation is finished, the `Installation Complete` dialog appears.

6. Click `Finish`.

End of procedure

Next Steps

- To review the installation and confirm the location of your Platform SDK files, continue with Verifying Installed Components.
- To configure additional options required by Platform SDK implementations involving SIP, go to Configuring SIP Endpoint Application Block Overview, page 33.

Procedure:
Verifying Installed Components

Purpose: To confirm that the Platform SDK installation was successful, and become familiar with the folder structure of Platform SDK components.

Start of procedure

1. Use your file manager to locate the destination directory for the Platform SDK installation.

   For this procedure, we will assume a Windows installation that uses the default location:
   `C:\Program Files\GCTI\Platform SDK for <.NET/Java> 7.6.`

2. Confirm that your Platform SDK components look similar to the .NET and Java examples in Figure 1.

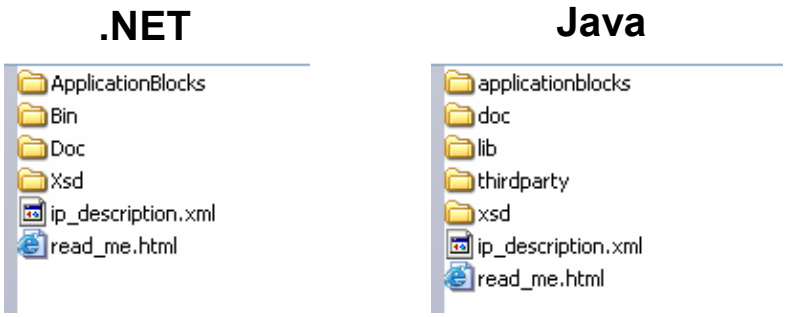


Figure 1: Sample .NET and Java Installations

3. Examine the contents of each folder (including the root installation folder) to confirm their contents.

   Refer to Table 2 on page 31 for a description of the expected contents for a .NET installation, or Table 3 on page 32 for a Java installation.

End of procedure

Next Steps

- To configure additional options required by Platform SDK implementations involving SIP, continue with Configuring SIP Endpoint Application Block Overview.

Table 2: .NET Folder Contents

Folder	Contents
\	The root directory contains the following two files: • `ip_description.xml`—This contains the data for the `read_me.html` file. • `read_me.html`—The `ReadMe` identifies the build number, platform compatibility, and a link to the latest release notes.
\ApplicationBlocks	The `ApplicationBlocks` directory contains one subdirectory for each application block included with the Platform SDK. Each application block subdirectory is a self-contained unit that has: • A `Doc` subdirectory that contains an *Application Block Guide* in `.chm` format. • A `QuickStart` subdirectory (where applicable). This location includes a small code sample that allows you to see the application block in action. • An `Src` subdirectory with all source code for that application block. This code is available for you to use as-is, or to customize as needed. • Related technology-based files. For example, this includes the .NET Microsoft Visual Studio `.sln` files.
\Bin	This directory has the .NET libraries (`.dll` files) for all the .NET Platform SDKs, including the `Core` and `Commons` `.dll` files.
\Doc	This directory stores API references (in `.chm` format) for all of the SDKs you installed.
\Xsd	Here you will find the XML messages used by the SDKs you installed.

Table 3: Java Folder Contents

Folder	Contents
\	The root directory contains the following two files: • `ip_description.xml`—This contains the data for the `read_me.html` file. • `read_me.html`—The `ReadMe` identifies the build number, platform compatibility, and a link to the latest release notes.
\applicationblocks	The `applicationblocks` directory contains one subdirectory for each application block included with the Platform SDK. Each application block subdirectory is a self-contained unit that has: • A `doc` subdirectory that contains an *Application Block Guide* in JavaDoc format (archived in `api.jar` for UNIX). • A `quickstart` directory (where applicable). This location includes a small code sample that allows you to see the application block in action. • An `src` directory with all the application block's source code. This code is available for you to use as-is, or to customize as needed. (For UNIX installations, the source code is archived as `src.jar` files.) • Related technology-based files.
\doc	This directory stores API references (as `.html` for Windows, or as an `api.jar` archive for UNIX) for all of the SDKs you installed.
\lib	This directory contains the Java archive (`.jar`) files for all the Java SDKs and application blocks, including: `commons.jar`, `connection.jar`, `concurrent.jar`, `kvlistbinding.jar`, `kvlists.jar`, `protocol.jar`, and `system.jar`.
\thirdparty	This directory contains required third-party components necessary for implementing your Platform SDK.
\xsd	Here you will find the XML messages used by the SDKs you installed.

Configure Platform SDK Options

Other than making provisions for your environment to use the SIP Endpoint Application Block, there are no Configuration Layer planning issues related to the Platform SDKs.

If you intend to integrate your custom application into a Genesys environment, you may need to configure certain `Application` objects and related resources. However, these are not prerequisites for using the Platform SDKs. If you need to implement Configuration Layer objects, see the *Framework 7.6 Deployment Guide* for details on how to perform this and related tasks.

Procedure: Configuring SIP Endpoint Application Block Overview

Purpose: On your Genesys Environment side, to use the SIP Endpoint application block with your Configuration Layer, you need to make sure you configure your telephony resources properly. Each SIP endpoint instance must point to a DN resource on your SIP switch, a `switch` object in the Configuration Layer. In addition, you may need to alter some options in your SIP Server `Application` object.

Prerequisites

- Prior to using the SIP Endpoint application block, ensure you have the proper environment prerequisites (including .NET Framework 2.0 and the current RTC release). See “Environment Prerequisites” on page 24 for details.

Start of procedure

1. Use Configuration Manager to configure options for your SIP Server `Application` object. Detailed steps for this process are available in Configuring SIP Server Options, page 34.
2. Use Configuration Manager to configure options for each DN that will be used with your SIP Endpoint instance. Detailed steps for this process are available in Configuring SIP DN Options, page 34.

End of procedure

Next Steps

- Instructions for using the SIP Endpoint Application Block’s QuickStart application are available in the *SIP Endpoint .NET Application Block 7.6 Guide.* That document also includes instructions on what values to change in the configuration files associated with the application block.

Procedure: Configuring SIP Server Options

Purpose: To configure the `TServer` section of your SIP Server `Application` object to address environment issues when using the SIP Endpoint application block.

TIP See your *Framework 7.6 SIP Server Deployment Guide* for more information on these and other options related to Genesys SIP Server.

Start of procedure

1. Open Configuration Manager.
2. Select your SIP Server `Application` object, and view the `TServer` section of the `Options` tab.
3. If you are using a mixed RTC phone and Polycom 500/600 environment, set the `sip-enforce-sdp-origin-rules` option to `true`.

Persistent Storage of Registrar Information

4. To store registrar information persistently, set the following options:
   - `internal-registrar-enabled` — true
   - `internal-registrar-persistent` — true

   If SIP Server is configured for persistent storage of registrar information, then SIP Endpoint Application Block retains its connection to SIP Server after SIP Server restarts.
5. Click `Apply` to save changes in Configuration Manager.

End of procedure

Next Steps

- Be sure to also apply the appropriate security settings for your SIP `Switch` object. For more information, refer to the *SIP Server Deployment Guide.*
- Configuring SIP DN Options, page 34

Procedure: Configuring SIP DN Options

Purpose: To configure each `DN` object that you plan to use with a SIP endpoint instance. This includes creating a section named `TServer` on the `Annex` tab, and adding options and values as described in this procedure.

TIP See your *Framework 7.6 SIP Server Deployment Guide* for more information on these and other options related to Genesys SIP Server.

After you have finished this procedure, the `TServer` section for your `DN` should look similar to Figure 2.

Start of procedure

1. Open Configuration Manager.
2. Select your DN that will be used with a SIP endpoint instance, and view the `Annex` tab.

> **Note:** If the `Annex` tab is not visible, then use the `View > Options` menu in Configuration Manager to select the `Show Annex tab in object properties` option.

3. Create a new section named `TServer`.
4. On the `Annex` tab, create the following new options, with the values specified:
   - `reinvite-requires-hold` — `true`
   - `sip-cti-control` — `talk,hold`
   - `transfer-complete-by-refer` — `false`
5. Confirm that the `refer-enabled` option is either absent, or (if present) set to `true`.

> **Note:** Do not use the `make-call-rfc3755-flow` option, which has been deprecated.

6. Click `Apply` to save these changes in Configuration Manager.

End of procedure

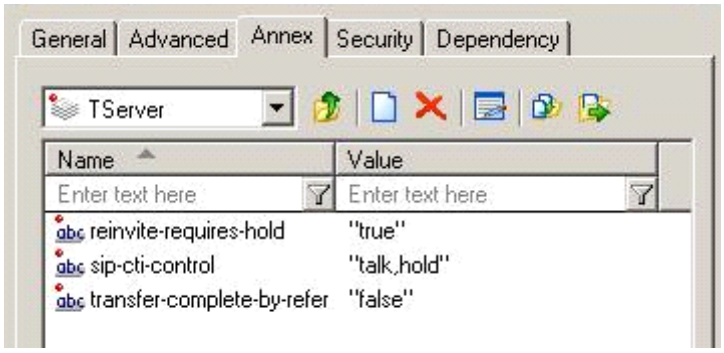


Figure 2: Mandatory SIP Endpoint Options for Each SIP DN

Chapter

3 Starting and Testing Your Applications

This chapter briefly describes how to get started using your Platform SDKs and any associated application blocks you may have installed. Details on the particulars of all these SDKs are available in the *Platform SDK 7.6 Developer's Guide* and in the *API Reference* for each product.

This chapter has the following sections:

Concepts

Using the Platform SDKs requires not only a thorough knowledge of the technology you plan to use (.NET, Java, or XML), but also a solid understanding of the workings of the underlying Genesys server with which you plan to integrate. For instance, for use of the Voice Platform SDK, you should be familiar with the concepts of CTI (computer-telephony integration) and of T-Server–client interaction. Additionally, for use of the SIP Endpoint Application Block, you should be familiar with IP Voice technology in general, and SIP technology and the Genesys SIP Server in particular.

Using Application Blocks and Code Examples

Once the Platform SDKs are installed and configured on your system, Genesys recommends that you run the existing QuickStart and code examples before

beginning your own development. This ensures that your system is working correctly, and that you are ready to begin creating custom applications.

This section includes the following topics:

Building Application Blocks

Starting with release 7.6.2, you are no longer required to build application blocks before using them in your development. You can simply reference the included `.dll` or `.jar` files (depending on whether you are using .NET or Java, respectively) as resources for your project and begin programming.

However, if you customize or adapt the application blocks by changing the source code provided (for example, by adding UCS support to the Protocol Manager Application Block as described in the Application Block Guide) then you will need to manually build the application blocks before those changes are reflected.

The build process for these files is different depending on whether you are using .NET or Java.

- Each .NET application block comes with a Visual Studio Solution file that you use to build the related `.dll`. See Building the .NET Application Blocks for details.
- Each Java application block comes with a build file that is designed to automatically create the related `.jar` file for you:
  - For Windows systems, this file is named `build.bat`.
  - For Unix systems, this file is named `build.sh`.

  See Building the Java Application Blocks, page 39 for details.

Procedure: Building the .NET Application Blocks

Purpose: To create the `.dll` files required to use .NET application blocks in your Platform SDK development.

Prerequisites

- Install Platform SDK for .NET on your system, as described in “Install the Platform SDKs” on page 28.

Start of procedure

1. Locate and open the Visual Studio Solution file provided for the application block you plan to use:
   `<installation path>\ApplicationBlocks\<AB_Name>\<AB_Solution>.sln`

   Note: Some application blocks also contain "QuickStart" solutions. For more information about the purpose and use of these QuickStart applications, see "Starting QuickStart Applications" on page 40.

2. Build the Solution.

   Visual Studio creates the relevant `.dll` files in the following location:
   `<installation path>\ApplicationBlocks\<AB_Name>\Src\bin\Debug\`
3. Repeat this procedure for each application block that you plan to use.

End of procedure

Next Steps

- Now that the .NET application blocks are ready to use, try starting the QuickStart and sample applications before beginning development. See "Starting the Code Examples" on page 40 for details.

Procedure: Building the Java Application Blocks

Purpose: To create the .jar files required to use Java application blocks during your Platform SDK development.

Prerequisites

- Install Platform SDK for Java on your system, as described in "Install the Platform SDKs" on page 28.
- Ensure that ANT is installed and configured correctly on your system. (This includes setting the `ANT_HOME` and `JAVA_HOME` environment variables.)
- Before building the Protocol Manager Application Block for Java, you must build the Warm Standby Application Block.

TIP Protocol Manager includes warm standby functionality, and requires the `warmstandbyappblock.jar` file to build correctly.

Start of procedure

1. Locate the build file for the application block you plan to use:
   `<installation path>\applicationblocks\<AB_Name>\`

Note: This build file assumes that your `ANT_HOME` environment does not contain any spaces. If this is not the case, then you should manually edit the build file to account for those spaces. For example, an edited `build.bat` file might read:

```
CALL "%ANT_HOME%\bin\ant" -logfile ANT.log
```

2. Double-click on the build file.

   ANT creates the new `.jar` file, and places it at the following location:

   `<installation path>\applicationblocks\<AB_Name>\dist\lib\`

3. Repeat this procedure for each application block that you plan to use.

End of procedure

Next Steps

- Now that the Java application blocks are ready to use, try starting the QuickStart and sample applications before beginning development. See "Starting the Code Examples" for details.

Starting the Code Examples

The Platform SDKs are installed and ready to use. The application blocks have been built into useful libraries that your projects can reference. So what comes next?

There is one final step before you begin developing your own applications: starting the QuickStart and code examples to ensure that your system is set up correctly.

Starting QuickStart Applications

QuickStart applications make it easy for you to see how the application blocks are used. They are packed-in with the application blocks, and can be used to check if your development system is ready for use.

The following application blocks include a QuickStart application:

- Configuration Object Model (.NET and Java)
- SIP Endpoint (.NET only)
- Warm Standby (.NET only)

In each case, the QuickStart application is included in the same location as the application block code for easy access.

Note: The QuickStart applications are for educational purposes only, and are neither tested nor supported by Genesys in any way.

Procedure:
Starting the .NET QuickStart Applications

Purpose: To test your installation by configuring and running the .NET QuickStart applications.

Prerequisites

- Before trying to start a QuickStart application, check the folder where that application is located for any `.config` files. These files provide information about the Genesys environment, and must be edited to point to your environment before continuing.

Start of procedure

1. Open the `<installation path>\ApplicationBlocks\<AB_Name>` folder.
2. Double-click `<AB_Name>QuickStart.sln` to open the Visual Studio Solution.
3. Build the solution.

   Visual Studio will create an executable for the QuickStart application.
4. Locate and double-click the executable file:

   `<installation path>\ApplicationBlocks\<AB_Name>\QuickStart\bin\Debug\<AB_Name>QuickStart.exe`

End of procedure

Procedure:
Starting the Java QuickStart Application

Purpose: To test your installation by configuring and running the Java QuickStart application.

Prerequisites

- Ensure that your system PATH environment variable includes the location of your installed JDK.

  TIP Supported versions of JDK are listed in the *Genesys Supported Operating Systems and Databases*.

- Before trying to start the QuickStart application, check the folder where the application is located and configure the `quickstart.properties` file. This properties file provides information about the Genesys environment, and must be edited to point to your environment before continuing.

Start of procedure

1. Open the `<installation path>\applicationblocks\com\quickstart` folder.
2. Double-click the appropriate build file to compile the `MainClass` class.
   - For Windows, use `build.bat`.
   - For Unix, use `build.sh`.
3. Double-click the appropriate QuickStart file to run application.
   - For Windows, use `quickstart.bat`.
   - For Unix, use `quickstart.sh`.

End of procedure

Starting Code Examples

"Sample Applications" on page 22 describes the various code examples that are included with your Platform SDK installation.

Refer to the *Platform SDK Developer's Guide* for instructions about how to start these code examples, and for a detailed explanation of how the code works.

Development Issues

Development issues are covered in depth in the *Platform SDK 7.6 Developer's Guide.* Furthermore, the *API Reference* for each Platform SDK and the various *Application Block Guides* have code snippets and general information to help you get started.

Index

Symbols

A

C

D

E

I

M

P

R

S

T

V

W