Platform SDK 8.0

Deployment Guide

The information contained herein is proprietary and confidential and cannot be disclosed or duplicated without the prior written consent of Genesys Telecommunications Laboratories, Inc.

Copyright © 2009 Genesys Telecommunications Laboratories, Inc. All rights reserved.

About Genesys

Genesys Telecommunications Laboratories, Inc., a subsidiary of Alcatel-Lucent, is 100% focused on software for contact 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 regional numbers provided on page 10. For complete contact information and procedures, refer to the ***Genesys Technical Support Guide***.

Ordering and Licensing Information

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

Released by

Genesys Telecommunications Laboratories, Inc. www.genesyslab.com

Document Version: 80sdk_dep_platform_07-2009_v8.0.001.00

Table of Contents

List of Procedures

Preface

Welcome to the *Platform SDK 8.0 Deployment Guide*. This document introduces you to the architecture, required components, and procedures relevant to the deployment of Platform SDK in your contact center.

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

- Prerequisites for deploying Platform SDK.
- How to install Platform SDK on your system.
- How to configure related portions of your Genesys environment to accommodate Platform SDK.

This document is valid only for the 8.0 release of this product.

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

This preface contains the following sections:

For information about related resources and about the conventions that are used in this document, see the supplementary material starting on page 37.

About Platform SDK 8.0

Platform SDK 8.0 allows you to write .NET and Java applications that communicate with Genesys servers in their native protocols. You can think of the APIs in Platform SDK as “Server APIs,” since each one unlocks the capabilities of the server it connects to. In contrast to the abstraction found in other Genesys SDKs, Platform SDK was designed to offer low-level components and fine-grained, message-driven interfaces which are also XML friendly.

Intended Audience

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

It has been written with the assumption 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, Java- and .NET-related development tools, and 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 the express written consent of Genesys.

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 current Genesys 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.

Making Comments on This Document

If you especially like or dislike anything about this document, 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 scope of this document only and to the way in which the information is presented. Contact your Genesys Account Representative or 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.

Contacting Genesys Technical Support

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

Region	Telephone	E-Mail
North and Latin America	+888-369-5555 (toll-free) +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
Before contacting technical support, refer to the *Genesys Technical Support Guide* for complete contact information and procedures.		

Region	Telephone	E-Mail
India	1-800-407-436379 (toll-free) +91-(022)-3918-0537	support@genesyslab.com.au
Japan	+81-3-6361-8950	support@genesyslab.co.jp
Before contacting technical support, refer to the *Genesys Technical Support Guide* for complete contact information and procedures.		

Chapter

1 Overview

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

Introduction

The Platform SDK allows you to create unique, custom-built applications that connect to Genesys servers to provide the information and interactivity you need.

Table 1 provides a full list of Genesys servers that you can work with, along with the related Platform SDK API that must be used to communicate with that server. For more information, see "Working with Genesys Servers" on page 15.

Table 1: Genesys Servers and Related APIs

Genesys Server	Related API
Configuration Server	Configuration Platform SDK
Interaction Server	Open Media Platform SDK
Management Layer Servers	Management Platform SDK
Outbound Contact Server	Outbound Platform SDK

Table 1: Genesys Servers and Related APIs (Continued)

Genesys Server	Related API
Stat Server	Statistics Platform SDK Packaged Statistics Platform SDK[a]
T-Servers	Voice Platform SDK
Universal Contact Server	Contacts Platform SDK
Web Media Servers	Web Media Platform SDK

a. The Packaged Statistics Platform SDK API is deprecated, and only included with this release to provide backwards compatibility with earlier releases of Platform SDK. This API is not intended for ongoing development.

Development Aids

To assist with development, the SDK is packaged with a combined *Platform SDK API Reference* document that allows you to find reference information, coding recommendations, and code snippets in a single location for all of the APIs.

Note: Information about deprecated Packaged Statistics Platform SDK API is provided in a separate, stand-alone *API Reference*.

Also included are a number of production-quality *application blocks* available for integration into your code, and 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 18 and “Sample Applications” on page 20 for more details.

Finally, every Genesys product also includes a *Release Note* 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 Note* for this product.

These development aids can be located on the Developer Documentation Library CD, and also from the Genesys Developer website located at http://www.genesyslab.com/developer.

Platform SDK Technologies

For the 8.0 release, each Platform SDK allows you to develop applications using one of the following technologies:

- .NET and XML
- Java and XML

Working with Genesys Servers

This section briefly outlines the Genesys servers that you will be working with, and provides a short description of the capabilities that related APIs offer. Please note that every API is made available for both .NET and Java development platforms.

Refer to the *Platform SDK 8.0 API Reference* for either .NET or Java to find more information, including detailed interface descriptions.

Configuration Server

Platform SDK allows you to build applications that add, modify, and delete information in the Configuration Layer of your Genesys environment. By using the Configuration Platform SDK API to access information directly from Configuration Server, your applications can view Configuration Layer data and then modify it on the fly.

The Configuration Object Model (COM) Application Block included with Platform SDK makes it easier for you to work with configuration objects. See “Application Blocks” on page 18 for more details.

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

Interaction Server

Using the Open Media Platform SDK API allows you to design applications that interact with Interaction Server. Access to this Genesys server might be required for many tasks, including:

- Handling open media type interactions from an agent desktop.
- Introducing new open media interactions into your Genesys environment from external sources (from a server-side perspective).
- Performing external service processing (ESP) on interactions that are already stored in the Genesys environment.

- Submitting new interactions, combined with the ESP capabilities provided by the Open Media Platform SDK API, allows you to build custom media servers that can become part of your Genesys Multimedia environment.
- After building a custom media server, you can use specialized objects in Interaction Routing Designer to route interactions to and from your custom media server.

Management Layer Servers

The Management Layer Servers that Platform SDK can communicate with include Solution Control Server (SCS), LCA, and Message Server.

Using the Management Platform SDK API to communicate with these core Management Layer Servers give your application direct access to the status and logs of any component that your Genesys software is already tracking in your system. With this, you can design applications that monitor, start, or stop any other applications in your Genesys environment.

Outbound Contact Server

The Outbound Platform SDK API 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 schedule a given campaign to begin when a programmed time or date is reached, or after meeting some observed threshold within your contact center.

Stat Server

Stat Server provides contact center managers with a wide range of information about customer interaction networks, allowing organizations to maximize their efficiency and flexibility. It also converts the data accumulated for directory numbers (DNs), agents, agent groups, and non-telephony-specific object types, such as e-mail and chat sessions, into statistically useful information, and passes these calculations to other software applications that request data. For more information about Stat Server, consult the Reporting Technical Reference Guide and the Stat Server User's Guide.

You can use the Platform SDK to design Java or .NET applications that gather and manipulate statistical information from Stat Server. These applications may be fairly simple or quite advanced depending on their purpose.

For the 8.0 release of Platform SDK, there are two APIs that can be used to communicate with Stat Server:

- Statistics Platform SDK API has been completely restructured for release 8.0. This updated API provides improved scalability and compatibility with Stat Server, and should be used for all new development.

- Packaged Statistics Platform SDK provides backwards compatibility with code from previous releases. This is not compatible with the improved API described above, and any existing applications that were developed with the Packaged Statistics Platform SDK will not be able to take advantage of the improved capabilities found in the Statistics Platform SDK API.

  Packaged Statistics Platform SDK is provided as a separate library, and is not described or documented in the combined Platform SDK API Reference. For additional information, please refer to the separate *Packaged Statistics Platform SDK 8.0 API Reference* documentation.

Note: Genesys **does not recommend** using the Packaged Statistics API. That API has been deprecated, and is only provided to support existing applications. The Packaged Statistics API should not be used for any new development, and is not compatible with the new and improved Platform SDK Statistics API.

T-Servers

Using the Voice Platform SDK API allows you to design applications that communicate with T-Server in a Genesys environment—accomplishing tasks such as monitoring and controlling voice interactions. This API is designed to work solely with T-Servers, including those in premise or network roles, and using traditional voice or IP voice technologies. Server-side and agent-desktop applications can both make use of this API.

Universal Contact Server

Information stored in Universal Contact Server (UCS) can be accessed directly using the Contacts Platform SDK API. This allows you to build applications that view or interact with contact information for your contact center.

A practical example of an application that would require contact information might include simple agent desktops that are created to deal with multimedia interactions such as chat or e-mail, and provide standard agent responses for the new outgoing interactions that are created.

Web Media Servers

The Web Media Platform SDK API provides integration with Genesys non-voice media servers, and allows requests that are initiated in a web environment to be introduced into Genesys environment e-mail, chat, and callback servers. Using this API gives you an opportunity to fully customize your customer experience when they are trying to contact your company over the Internet.

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 working with a message-based API, you need to handle events. Many of your applications will also need to communicate with multiple servers, manage those connections, and keep track of the interactions that are returned from each server.

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 between your application and 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 reference the appropriate JAR (for Java development) or DLL (for .NET development) in your code. Source code is provided in case you want to see how application blocks are designed, or if you want to make modifications. Some application blocks even include a quick start application to get you started.

Please visit the Genesys Developer Zone website at http://www.genesyslab.com/developer to download the most recent versions of Genesys application blocks, documentation, and code examples. You will also find useful information and discussions in the forums.

Application Blocks Shipping With Platform SDK 8.0

There are four application blocks included with this release of Platform SDK, with a brief description of each provided in this section.

For examples of how these application blocks should be used in your applications, check the detailed descriptions in the *Platform SDK 8.0 API Reference* or the code examples described in "Sample Applications" on page 20.

Information about customizing and building the application blocks can be found in "Customizing Application Blocks" on page 33.

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.

Configuration Object Model

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

- Retrieve, 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.

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.

Protocol Manager

Use the Protocol Manager Application Block whenever you plan to use more than one Platform SDK 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.

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.

Sample Applications

Genesys understands how helpful working code can be in learning to use a new API. Because of that, the Platform SDK includes code snippets in the *Platform SDK API Reference* guides and application blocks.

However, Platform SDK also includes a set of sample applications for your convenience. These compiled, ready-to-run code examples typically illustrate common use cases and can help you get up and running during early 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.

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 3.5 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 for details.
- For Java implementations, refer to the *Genesys Supported Operating Environment Reference Manual* 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 `JAXB 2.1` (Java Architecture for XML Binding). The `JAXB 2.1` is available with your installation of the Platform SDK in the `thirdparty` folder.

Platform SDK and AES Cryptography

Platform SDK uses AES cryptography as part of the Configuration Platform SDK API. AES encryption uses a 128-bit encryption key, which is considered too strong for some countries and thus subject to export restrictions.

For information about how your Java environment supports this standard, please check the following website:

- `http://java.sun.com/developer/technicalArticles/Security/AES/AES_v1.html`

Deployment Task Summary

Task Summary: Platform SDK Deployment Work Flow 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 23 and Chapter 3 on page 29.

Task Summary: Platform SDK Deployment Work Flow

Objective	Related Procedures and Actions
Install and verify the Platform SDKs on your system.	1. Procedure: Installing Platform SDK 8.0, on page 24. 2. Procedure: Verifying Installed Components, on page 25.
Use the QuickStart applications to test your application blocks.	1. Procedure: Starting the .NET QuickStart Applications, on page 30 2. Procedure: Starting the Java QuickStart Applications, on page 31
Build the `.dll` or `.jar` reference files for your application block (if making any customization to the included source code).	1. Procedure: Building the .NET Application Blocks, on page 33 2. Procedure: Building the Java Application Blocks, on page 34

Chapter

2 Installation and Configuration

This chapter describes the process of installing Platform SDK on Windows and UNIX operating systems. Although Platform SDK is available 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 Genesys Platform SDK Product CD

Your Genesys 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 8 Licensing Guide*.

Install Platform SDK 8.0

This section provides detailed procedures that describe how to install and configure Platform SDK 8.0 on your system.

Tip: Starting with Platform SDK release 7.6.201, you are no longer required to remove other instances of Platform SDK from your system before installing.

Procedure: Installing Platform SDK 8.0

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

Prerequisites

- Check the list of "Environment Prerequisites" on page 21, and confirm that these requirements are met prior to installing Platform SDK.

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. Select 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. For Windows installations, the default directory is:
   C:\Program Files\GCTI\Platform SDK for <.NET/Java> 8.0.

4. Click Next if you want to accept the default destination folder that is specified.

   If you prefer to install Platform SDK in a different location than the default 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 Procedure: Verifying Installed Components.

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> 8.0

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

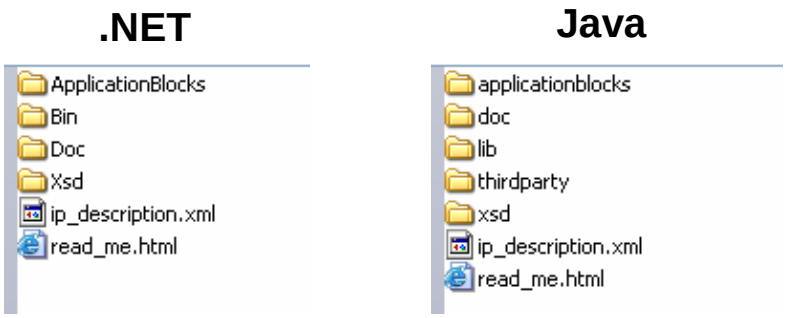

Figure 1: Sample .NET and Java Installations

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

   Refer to Table 2 for a description of the expected contents for a .NET installation, or Table 3 on page 27 for a Java installation.

End of procedure

Table 2: .NET Folder Contents

Folder	Contents
\	The root directory contains the following two files: • `ip_description.xml`—This file contains 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 has one subdirectory for each application block that is included with this release of Platform SDK. Each application block subdirectory is a self-contained unit that has: • A `QuickStart` subdirectory (where applicable). This 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 could include the .NET Microsoft Visual Studio `.sln` files.

Table 2: .NET Folder Contents (Continued)

Folder	Contents
\Bin	This directory holds the .NET libraries (as both `.dll` and `.xml` files) for all APIs and application blocks contained in the .NET version of Platform SDK, including the `Core` and `Commons` namespaces.
\Doc	This directory stores the combined *Platform SDK API Reference* (in `.chm` format). This document contains detailed information about how to develop content using Platform SDK 8.0. A second `.chm` file in this folder contains stand-alone documentation for the deprecated Packaged Statistics SDK. For more information about this deprecated API, see "Working with Genesys Servers" on page 15.
\Xsd	The `Xsd` directory contains XML Schema Definition files that define the structure of the serialized messages used by Platform SDK.

Table 3: Java Folder Contents

Folder	Contents
\	The root directory contains the following two files: • `ip_description.xml`—This file contains 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 `quickstart` directory (where applicable). This 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.

Table 3: Java Folder Contents (Continued)

Folder	Contents
\doc	There are two subdirectories containing documentation: • The `api` subdirectory stores the combined *Platform SDK API Reference* (as expanded `JavaDoc` files for Windows, or in the `api.jar` archive for UNIX). This document contains detailed information about how to develop content using Platform SDK 8.0. • The `packagedstatisticsdeprecated` directory contains stand-alone documentation for the deprecated Packaged Statistics SDK. For more information about this deprecated API, see "Working with Genesys Servers" on page 15.
\lib	This directory contains the Java archive (`.jar`) files for all APIs and application blocks contained in the Java version of the Platform SDK, including: `commons.jar`, `connection.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	The `Xsd` directory contains XML Schema Definition files that define the structure of the serialized messages used by Platform SDK.

Chapter

3 Starting and Testing Your Applications

This chapter briefly describes how to start using Platform SDK (and the associated application blocks that you have installed) in your applications. Details for each API, along with code snippets and recommendation, are available from the *Platform SDK API Reference*.

This chapter has the following sections:

Concepts

Using Platform SDK 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 for each underlying Genesys server that you plan to work with. For instance, when connecting to T-Servers through the Voice Platform SDK, you should be familiar with the concepts of CTI (computer-telephony integration) and of T-Server–client interaction.

Using Application Blocks and Code Examples

Platform SDK is installed and ready to use. So what comes next? Before beginning development on your own applications, Genesys recommends that you run code examples (including existing QuickStart applications) to ensure that your system setup is configured correctly.

After running the code examples, you may choose to review or customize code for the application blocks. If any changes or update are made, you will need to rebuild the related application block libraries.

This section includes the following topics:

Working with Code Examples

This section will describe how to access the two types of code examples available for use with Platform SDK:

1. QuckStart Applications that are distributed with the source code for some application blocks.
2. Additional code examples that can be located and downloaded from the Genesys Developer website at http://www.genesyslab.com/developer.

Introducing the 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
- Warm Standby

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.

Starting the .NET QuickStart Applications

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

Prerequisites

- Install Platform SDK for .NET on your system, as described in “Install Platform SDK 8.0” on page 24.

- Before you build or debug the application block QuickStart applications, you need to update the project references to point to libraries stored in `<installation path>\Bin`.
- Before trying to start a QuickStart application, check the folder where that application is located for the `App.config` file. This file uses XML to provide configuration details that are specific to your Genesys environment. You must edit the `App.config` file to contain the correct information 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 Applications

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

Prerequisites

- The Java QuickStart applications reference the libraries created in default folders when you build the Java application blocks. You should successfully build the application blocks before starting either the COM or Warm Standby QuickStart applications.

  For details on how to build an application block, see Procedure: Building the Java Application Blocks, on page 34.
- 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 Environment Reference Manual*.

- Ensure that your system ANT_HOME environment variable is set, and Apache Ant is configured correctly on your system.

  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
  ```

- Before trying to start a QuickStart application, check the folder where that application is located for the `quickstart.properties` file. This file provides configuration details that are specific to your Genesys environment. You must edit the `quickstart.properties` file to contain the correct information 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

Genesys provides code examples that help to show simple applications that make use of the Platform SDKs. All code examples are intended 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 instructions on how start and work with these code examples, visit the Genesys Developer Zone forums at http://www.genesyslab.com/developer.

Customizing Application Blocks

Platform SDK does not require you to build application blocks before using them in your development—you can simply reference the appropriate `.dll` or `.jar` files (depending on whether you are using .NET or Java, respectively) that are included as resources for your project and begin programming right away!

However, if you customize or adapt the application blocks by changing the source code provided—for example, by customizing the Protocol Manager Application Block as described in the *Platform SDK API Reference*—then you need to manually build the application blocks before those changes are reflected.

The build process for application block source 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 Procedure: Building the .NET Application Blocks for details.
- Each Java application block comes with a build script 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 Procedure: Building the Java Application Blocks, on page 34 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 Platform SDK 8.0” on page 24.

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 "Introducing the QuickStart Applications" on page 30.

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 your customized .NET application blocks are ready to use, try starting the QuickStart and sample applications again using your new `.dll` files as a quick test before beginning development. See "Working with Code Examples" on page 30 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

- 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 references the `warmstandbyappblock.jar` file that is created after the Warm Standby Application Block is built.

- Install Platform SDK for Java on your system, as described in "Install Platform SDK 8.0" on page 24.
- Ensure that ANT is installed and configured correctly on your system. (This includes setting the `ANT_HOME` and `JAVA_HOME` environment variables.)

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 your customized Java application blocks are ready to use, try starting the QuickStart and sample applications again using your new `.jar` files as a quick test before beginning development. See “Working with Code Examples” on page 30 for details.

Development Issues

For more information about developing content using Platform SDK, please refer to the *Platform SDK API Reference*. This combined document provides introductions to each Platform SDK protocol, coding recommendations and suggestions, code snippets to help get you started, and other useful information.

If you experience development issues that are not covered in your version of the API Reference, please visit the Genesys Developer website (located at http://www.genesyslab.com/developer) to download the most recent version, which may contain useful corrections and additions.

While at the Genesys Developer website, browse and participate in our developer forums where users can support each other, provide feedback about this product, and get answers for difficult questions about Platform SDK usage.

Related Documentation Resources

The following resources provide additional information that is relevant to this software. Consult these additional resources as necessary.

Platform SDK 8.0

- *Platform SDK API Reference*, available for both Java and .NET, which provides introductions to each protocol contained in the Platform SDK, recommendations and suggestions for "best practice" development, code snippets that can help jump-start your projects, and other useful information.
- *Platform SDK Code Examples*, which provide working examples of how you can use Platform SDK in projects of your own. For more details, see "Sample Applications" on page 20.

Genesys

- *Genesys Technical Publications Glossary*, which ships on the Genesys Documentation Library DVD and which provides a comprehensive list of the Genesys and computer-telephony integration (CTI) terminology and acronyms used in this document.
- *Genesys Migration Guide*, which ships on the Genesys Documentation Library DVD, and which provides documented migration strategies for Genesys product releases. Contact Genesys Technical Support for more information.
- Release Notes and Product Advisories for this product, which are available on the Genesys Technical Support website at http://genesyslab.com/support.

Information about supported hardware and third-party software is available on the Genesys Technical Support website in the following documents:

- *Genesys Supported Operating Environment Reference Manual*

- *Genesys Supported Media Interfaces Reference Manual*

Consult the following additional resource as necessary:

- *Genesys Licensing Guide*, which introduces you to the concepts, terminology, and procedures relevant to the Genesys licensing system.

For additional system-wide planning tools and information, see the release-specific listings of System Level Documents on the Genesys Technical Support website, accessible from the system level documents by release tab in the Knowledge Base Browse Documents Section.

Genesys product documentation is available on 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.
- Genesys Developer website at http://www.genesyslab.com/developer.

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:

80fr_ref_06-2008_v8.0.001.00

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

Screen Captures Used in This Document

Screen captures from the product graphical user interface (GUI), as used in this document, may sometimes contain minor spelling, capitalization, or grammatical errors. 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.

Type Styles

Table 4 describes and illustrates the type conventions that are used in this document.

Table 4: Type Styles

Type Style	Used For	Examples
Italic	• Document titles • Emphasis • Definitions of (or first references to) unfamiliar terms • Mathematical variables Also used to indicate placeholder text within code samples or commands, in the special case where angle brackets are a required part of the syntax (see the note about angle brackets on page 40).	Please consult the *Genesys Migration Guide* for more information. Do *not* use this value for this option. A *customary and usual* practice is one that is widely accepted and used within a particular industry or profession. The formula, $x + 1 = 7$ where *x* stands for . . .

Table 4: Type Styles (Continued)

Type Style	Used For	Examples
Monospace font (Looks like `teletype` or `typewriter text`)	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. • Code samples. Also used for any text that users must manually enter during a configuration or installation procedure, or on a command line.	Select the `Show variables on screen` check box. In the `Operand` text box, enter your formula. Click `OK` to exit the `Properties` dialog box. T-Server distributes the error messages 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. Enter `exit` on the command line.
Square brackets ([ ])	A particular parameter or value that is optional within a logical argument, a command, or some programming syntax. That is, the presence of the parameter or value is not required to resolve the argument, command, or block of code. The user decides whether to include this optional information.	`smcp_server -host [/flags]`
Angle brackets (< >)	A placeholder for a value that the user must specify. This might be a DN or a port number specific to your enterprise. **Note:** In some cases, angle brackets are required characters in code syntax (for example, in XML schemas). In these cases, italic text is used for placeholder values.	`smcp_server -host <confighost>`

Index

T

V

W